



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO

INSTITUTO DE CIENCIAS BÁSICAS E INGENIERÍA

**DOCTORADO EN CIENCIAS EN AUTOMATIZACIÓN Y
CONTROL**

TESIS

**DESARROLLO Y CONTROL DE UNA ARQUITECTURA ROBÓTICA MÓVIL
HETEROGÉNEA Y SU APLICACIÓN AL MONITOREO AMBIENTAL**

Para obtener el grado de
Doctor en Ciencias en Automatización y Control

PRESENTA

M. en C. Gildardo Godinez Garrido

Director:

Dr. Hugo Romero Trejo

Codirector:

Dr. Omar Jacobo Santos Sánchez

Comité tutorial:

Dr. Omar Arturo Domínguez Ramírez

Dr. Jesús Patricio Ordaz Olivier

Dr. Sergio Salazar Cruz

Dr. Eduardo Cornejo Velázquez

Pachuca de Soto, Hgo., México 11 de marzo de 2024



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO

Instituto de Ciencias Básicas e Ingeniería

School of Engineering and Basic Sciences

Área Académica de Computación y Electrónica

Department of Electronics and Computer Science

Mineral de la Reforma Hidalgo, a 22 de enero de 2024

Número de control: ICBI-AACyE/0067/2024

Asunto: Autorización de impresión de tema de tesis

**MTRA. OJUKY DEL ROCÍO ISLAS MALDONADO
DIRECTORA DE ADMINISTRACIÓN ESCOLAR DE LA UAEH**

El Comité Tutorial del nombre del producto que indique el documento curricular del programa educativo de posgrado titulado “**DESARROLLO Y CONTROL DE UNA ARQUITECTURA ROBÓTICA MÓVIL HETEROGÉNEA Y SU APLICACIÓN AL MONITOREO AMBIENTAL**”, realizado por el sustentante **Gildardo Godínez Garrido** con número de cuenta 300945, perteneciente al programa del Doctorado en Ciencias en Automatización y Control, una vez que se ha revisado, analizado y evaluado el documento recepcional de acuerdo a lo estipulado en el artículo 110 del Reglamento de Estudios de Posgrado, tiene a bien extender la presente.

AUTORIZACIÓN DE IMPRESIÓN

Por lo que el sustentante deberá cumplir con los requisitos del Reglamento de Estudios de Posgrado y con lo establecido en el proceso de grado vigente.

Atentamente

“Amor, Orden y Progreso”

**Dr. Hugo Romero Trejo
Director de Tesis**

**Dr. Omar Jacobo Santos Sánchez
Codirector de Tesis**

Comité Tutorial

- Dr. Omar Arturo Domínguez Ramírez Presidente
- Dr. Jesús Patricio Ordaz Oliver Secretario
- Dr. Hugo Romero Trejo Vocal 1
- Dr. Omar Jacobo Santos Sánchez Vocal 2
- Dr. Sergio Salazar Cruz Vocal 3
- Dr. Eduardo Cornejo Velázquez Suplente

UAEH _____
 UAEH _____
 UAEH _____
 UAEH _____
 IPN-UMI **Sergio Salazar Cruz** _____
 UAEH _____

Ciudad del Conocimiento
Carretera Pachuca-Tulancingo km 4.5 Colonia Carboneras, Mineral de la Reforma, Hidalgo, México. C.P. 42184
Teléfono: +52 (771) 71 720 00 ext. 2250, 2251
Fax 2109
aacye_icbi@uaeh.edu.mx





M. en C. Gildardo Godinez Garrido

**Desarrollo y Control de una
Arquitectura Robótica Móvil
Heterogénea y su Aplicación al
Monitoreo Ambiental**

Dedicatoria

Con amor para mis hijos Gilberto y Gael Godinez Monterrubio, ustedes son la fuente del sentimiento más poderoso que he experimentado en mi vida, los amo hijos.

Agradecimientos

Agradezco a Dios por haberme permitido concluir con este reto tan importante en mi vida profesional y académica, que sin duda contribuye a ejercer de mejor manera mi noble profesión de profesor.

A mis padres Ines Garrido Ibarra y Arnulfo Godinez Cabrera, por apoyarme incondicionalmente en cada uno de los retos que he tenido a lo largo de mi vida, por su enseñanza y amor brindados.

A mi esposa y compañera de vida Griselda Monterrubio Balderrama, por estar siempre a mi lado en momentos felices y complejos, siempre haciendo equipo, gracias por tu amor, tu paciencia y tu ayuda, te amo.

A mis hermanos Amalia y José Arnulfo, por todo su apoyo brindado y ayuda, gracias por ser mis mejores amigos y aliados en esta vida, los quiero mucho.

A mis directores de tesis Dr. Hugo Romero Trejo y Dr. Omar Jacobo Santos Sánchez, por su apreciable apoyo e incondicional ayuda a lo largo de mi formación académica, indudablemente dos personas con excelente conocimiento y habilidades, pero sobre todo con una gran calidad humana, ya que en momentos complejos personales, ambos me han apoyado incondicionalmente, gracias doctores, muchas gracias.

A la Universidad Tecnológica de Tulancingo por el apoyo económico y de tiempo brindado, así como del equipo e instalaciones que ocupé durante el desarrollo de este trabajo. Gracias, igualmente, a la Universidad Autónoma del Estado de Hidalgo por el apoyo académico y de infraestructura del Instituto de Ciencias Básicas e Ingeniería.

A mi comité sinodal, Dr. Omar Arturo Domínguez Ramírez, Dr. Jesús Patricio Ordaz Olivier, Dr. Sergio Salazar Cruz y Dr. Eduardo Cornejo Velázquez, gracias por sus apreciables observaciones y recomendaciones, para este trabajo.

A mis amigos y compañeros, Dr. Orlando García Pérez, por todas la enseñanza teórica y prácticas a lo largo de mi formación como maestro y ahora como doctor, muchas gracias, al Dr. Juan Carlos González Islas, al Dr. Juan Daniel Ramírez Zamora, la M. en C. Rodrigo Edgardo Velasco Rebollo, al M. en C. Omar Samperio Vásquez, gracias por todo su apoyo técnico, por compartir sus conocimientos conmigo y su valiosa amistad.

A CONACYT por el apoyo económico por un periodo de 17 meses al CVU: 552521.

Resumen

El proyecto de investigación abordado, presenta un sistema que integra una estación fija con un sistema SCADA creado en LABview® y 2 robots móviles para monitorear contaminantes nocivos en aire y tierra, empleando para ello un cuadricóptero y un carro todoterreno, ambos no tripulados y manipulados empleando leyes de control avanzadas para optimizar su ahorro energético o el desempeño en el seguimiento de trayectorias, esto con la finalidad de diseñar, crear y aplicar una plataforma heterogénea basada en robótica móvil que sirva para caracterizar la concentración de contaminación en zonas de interés.

Para el control de los robots se propone el uso del controlador PIX-HAWK®, el cual cuenta con prestaciones que facilitan la navegación empleando posicionamiento global, además, de permitir programar leyes de control a bajo nivel, a través de herramientas y librerías de código abierto. Es contemplada la aplicación de leyes de control basadas en el método del gradiente descendente en conjunto con el filtro de kalman y un controlador PID, para proponer un controlador subóptimo discreto autosintonizable en línea, dando las ventajas de contar con un lazo cerrado que permita garantizar, robustez, estabilidad y autosintonización, esto último es aplicado en el robot terrestre, debido a las demandas a las que es sometido acorde a las tareas a realizar, mientras que, para el robot aéreo se plantea un controlador Integral subóptimo, como estrategia de control. El sistema integra equipo basado en robots open hardware, para tomar muestras sólidas de suelo, mediante el carro, todoterreno, respecto a las lecturas de agentes nocivos en aire, se emplea un sistema de sensores embarcado en un cuadricóptero, con dichas muestras y lecturas es posible tener material e información, que pueden ser estudiadas e interpretadas por expertos en el campo de la contaminación ambiental. En este trabajo no se aborda esta etapa, el alcance está acotado a al desarrollo de ambos robots y la toma de muestras y lecturas que permitan deducir la presencia de contaminación. Cada robot cuenta con un sistema de navegación autónomo, y además, pueden ser manipulados remotamente

mediante radiofrecuencia, lo que permite realizar un trabajo colaborativo entre ellos y hacer un mejor aprovechamiento de los recursos energéticos y optimización de trayectorias, e interactuar con el usuario mediante un sistema SCADA y un ambiente virtual creado en Unity® donde pude apreciarse el comportamiento en tiempo real de cada robot.

Abstract

The research project addressed presents a system that integrates a fixed station with a SCADA system developed in LABview® and two mobile robots designed for the monitoring of deleterious contaminants in both air and soil mediums. This is executed through the utilization of an unmanned quadcopter and an off-road unmanned vehicle. These robotic entities are operated through the implementation of advanced control algorithms, optimizing either their energy efficiency or trajectory tracking performance. The overarching objective pertains to the conception, fabrication, and application of a heterogeneous robotic platform rooted in mobile robotics, with the explicit aim of characterizing pollution concentrations within zones of interest. Concerning the robot control paradigm, the utilization of the PIX-HAWK® controller is proposed. This controller is endowed with functionalities that facilitates global navigation. Furthermore, it permits the programming of low-level control laws by means of open-source tools and libraries. The incorporation of control laws founded upon the gradient descent method, in conjunction with the Kalman filter and a PID controller, is contemplated. This amalgamation is proposed to yield a suboptimal online auto-tunable discrete controller, affording the benefits of a closed-loop configuration that ensures robustness, stability, and self-tuning. The latter is specifically applied to the terrestrial robot, tailored to meet the demands commensurate with its intended tasks. Conversely, for the aerial robot, an integral suboptimal controller is proposed as a control strategy. The system integrates open hardware-based robotic equipment to collect solid soil samples via the off-road vehicle. Regarding readings of harmful agents in the air, a sensor system embarked upon the quadcopter is employed. The amalgamation of these samples and readings yields material and information, which can be analyzed and interpreted by experts in the realm of environmental pollution. This work does not encompass this stage; the scope is confined to the development of both robots and the collection of samples and readings to deduce the presence of contamination. Each robot

possesses an autonomous navigation system and can also be remotely manipulated via radiofrequency. This facilitates collaborative work between the robots, optimizing energy resources and trajectory management. Interaction with the user occurs through a SCADA system and a Unity®-based virtual environment, wherein real-time behavior of each robot can be observed.

Acrónimos

- **DH:** Denavit-Hartenberg
- **E:** Valor esperado
- **EKF:** Filtro de Kalman Extendido
- **FO:** Flujo óptico
- **GKF:** Filtro de Kalman Generalizado
- **GPS:** Global Positioning System
- **H:** Hamiltoniano
- **L:** Lagrangiano
- **LK:** LucasKanade (Método de LucasKanade)
- **LQG:** Linear Quadratic Gaussian
- **LQR:** Linear Quadratic Regulator
- **MSE:** Error cuadrático medio
- **N:** Número de iteraciones o instantes de tiempo
- **PID:** Proportional Integral Derivative
- **RF:** Radiofrecuencia
- **SCADA:** Supervisory Control And Data Acquisition
- **Tr:** Traza (trace en inglés)
- **UAV:** Unmanned Aerial Vehicle
- **ULV:** Unmanned Land Vehicle

Índice general

Resumen	V
Abstract	VII
Acronimos	IX
Índice general	I
Índice de figuras	V
Índice de cuadros	1
1. Introducción	2
1.1. Información preliminar.	5
1.1.1. Modelo matemático de un UAV	5
1.1.2. Modelo matemático de un ULV	6
1.1.3. Controladores aplicados a robots no tripulados tipo UAV y ULV	7
1.2. Planteamiento del problema	11
1.3. Justificación	11
1.4. Hipótesis	12
1.5. Objetivo general	12
1.5.1. Objetivos específicos	13
1.6. Metas.	13
1.7. Limitaciones	14
1.8. Experimentos realizados	15
1.9. Conclusión capítulo 1	15
2. Marco teórico	17
2.1. Control integral	17
2.2. Método de Denavit-Hartenberg	19
2.2.1. Selección de sistemas de referencia	21
2.2.2. Algoritmo de Denavit-Hartenberg	23
2.3. Controlador PID	25

2.4. Método de Lucas Kanade	29
2.4.1. Flujo óptico	29
2.5. Filtro de Kalman	33
2.6. Filtro de Kalman extendido	39
2.7. Método del gradiente descendente	44
2.8. Conclusión capítulo 2	48
3. Plataforma robótica móvil heterogénea	51
3.1. Descripción del sistema propuesto	51
3.1.1. Sistema de monitoreo de calidad de aire y visión	55
3.1.2. Sistema de recolección de muestras de suelo	56
3.2. Plataforma experimental UAV	58
3.3. Plataforma experimental ULV	62
3.4. Sistema SCADA	66
3.5. Conclusión capítulo 3	72
4. Modelado matemático de los robots móviles	75
4.1. Modelo matemático del vehículo aéreo	75
4.2. Modelo matemático robot todoterreno	84
4.3. Conclusión capítulo 4	94
5. Simulaciones y resultados experimentales	96
5.1. Simulación seguimiento de trayectoria UAV	96
5.2. Simulación seguimiento de trayectoria ULV	101
5.3. Simulación controlador subóptimo autosintonizable en motor del ULV	102
5.4. Resultados experimentales controlador subóptimo autosintonizable	108
5.5. Resultados del sistema de visión y sensor calidad de aire del UAV	112
5.6. Resultados sistema de recolección de muestras de suelo del ULV	116
5.7. Resultados experimentales mejora en la aproximación de posición del UAV	117
5.8. Resultados experimentales, control de posición y orientación del UAV	118
5.9. Resultados experimentales, control de posición y orientación del ULV	124
5.10. Conclusión capítulo 5	129
6. Conclusiones y trabajo futuro	132
6.1. Conclusión	132
6.2. Trabajo futuro	134
Bibliografía	136

A. Estrategias, métodos y algoritmos	144
A.1. Estrategia de control subóptimo autosintonizable	144
A.2. Estrategia de control aplicado al UAV	148
A.2.1. Modelo dinámico discretizado UAV	153
A.2.2. Control integral aplicado al modelo dinámico discreto del UAV	155
A.3. Algoritmos.	164
B. Planos y esquemas	168

Índice de figuras

1.1. Posiciones y ángulos de un UAV.	6
1.2. Posición y orientación de un ULV.	7
2.1. Esquema de control integral.	19
2.2. Convención Denavit-Hartenberg para un robot manipulador.	21
2.3. Convención para medir θ_i y α_i	21
2.4. Problema del flujo óptico.	29
2.5. Ventana de $n \times n$ píxeles a usar para el algoritmo de Lucas Kanade.	31
2.6. Pirámide de flujo óptico.	32
2.7. Esquema de sistema discretizado.	33
2.8. Representaciones del filtro de Kalman.	35
2.9. Algoritmo del filtro de Kalman.	35
2.10. Diagrama de flujo del proceso iterativo para el método de optimización.	48
3.1. Sistema de robótica móvil heterogénea y su aplicación al monitoreo ambiental.	53
3.2. Interacción entre el sistema SCADA, el UAV y el ULV.	54
3.3. Sistema de monitoreo de calidad de aire y de visión del UAV.	55
3.4. Sistema de recolección de suelo propuesto.	57
3.5. Sistema de recolección de suelo propuesto vista lateral izquierda posterior.	58
3.6. Plataforma experimental UAV.	59
3.7. Cuadricóptero basado en el drone SK450.	60
3.8. Esquema de variables de la planta UAV.	61
3.9. Plataforma experimental ULV.	63
3.10. Carro todo terreno basado en una plataforma shrimp rover.	64
3.11. Esquema de variables de la planta ULV.	65
3.12. Interfaz pantalla de configuración.	67
3.13. Interfaz monitoreo de variables del UAV.	68
3.14. Interfaz del sistema SCADA para ambos robots.	69
3.15. Interfaz monitoreo de variables del UAV LABview® y Unity®.	71

3.16. Interfaz monitoreo de variables del ULV LABvie® y Unity®.	71
3.17. Interfaz monitoreo de variables del ULV LABview®.	72
4.1. Esquema de helicóptero con cuatro rotores.	76
4.2. Posiciones y ángulos en un UAV.	82
4.3. Orientación del UAV.	83
4.4. Robot con arquitectura shrimp rover.	85
4.5. Marcos de coordenadas del shrimp rover.	86
4.6. Marcos de referencia aplicados al robot todo terreno propuesto.	87
4.7. Transformaciones para los marcos de referencia del Shrimp Rover.	87
5.1. Cuadro de bloques del modelo matemático de UAV simulado en Simulink®.	97
5.2. Bloque de simulación en Simulink® del modelo matemático del cuadricóptero más el bloque del controlador.	98
5.3. Posiciones y ángulos calculados cuando el modelo matemático es simulado en Simulink®.	99
5.4. Vistas de la trayectoria 3D de la simulación del UAV.	100
5.5. Resultados de simulación seguimiento de trayectoria rover.	101
5.6. Marcos de referencia para simulación de seguimiento de trayectoria del ULV.	102
5.7. Aproximación usando un filtro de Kalman simulado en MATLAB®.	103
5.8. Comportamiento del error al emplear el método del gradiente descendente simulado en MATLAB®.	104
5.9. Comportamiento de ganancias de PID usando método del gradiente descendente fuera de línea simulado en MATLAB®.	105
5.10. Comparación entre controlador PID con controlador PID subóptimo autosintonizable, simulados en MATLAB®.	106
5.11. Ganancias autoajustables del controlador PID subóptimo adaptable, simulados en MATLAB®.	106
5.12. Señales de control PID subóptimo autosintonizable contra PID, simulados en MATLAB®.	107
5.13. Comparación respuesta PID contra control subóptimo autosintonizable.	109
5.14. Comparación señal de control PID contra control subóptimo autosintonizable.	109
5.15. Comportamiento de ganancias de controlador subóptimo autosintonizable.	110
5.16. Transitorios del comportamiento de las ganancias de controlador subóptimo autosintonizable.	111
5.17. Resultados al aplicar Lucas Kanade en Raspberry® y enviar posición en x , y , z y calidad del aire (ca) mediante protocolo RS232.	112
5.18. Cadena de caracteres concatenada para ser enviada por protocolo RS232.	113

5.19. Gimbal del UAV armado.	114
5.20. Tarjeta electrónica RASPIX.	114
5.21. Pantalla de monitoreo de calidad del aire.	115
5.22. Sistema de toma de muestras de suelo.	116
5.23. Mejora en la aproximación de la posición $Z(t)$, mediante sensor YPLidar de 12m.	117
5.24. Mejora en las aproximaciones de las posiciones $X(t)$ e $Y(t)$, mediante sensor PX4-FLOW®.	118
5.25. Ganacias del controlador PID asignadas mediante LABview®.	119
5.26. Trayectoria de UAV para rutina de vuelo.	120
5.27. Trayectoria de UAV para rutina de vuelo vista aérea.	120
5.28. Posiciones y velocidades del UAV al aplicar controlador PID en posición y orientación.	121
5.29. Orientación y velocidades angulares del UAV al aplicar controlador PID en posición y orientación.	122
5.30. Torques y fuerza aplicados al UAV durante el seguimiento de trayectoria.	123
5.31. Trayectoria del ULV al aplicar controlador PID en posición y orientación.	124
5.32. Angulos y velocidades angulares del robot ULV.	125
5.33. Posiciones y velocidades del robot ULV.	126
5.34. Ángulos de articulaciones y posición del carro del ULV.	127
5.35. Velocidades angulares de los motores de tracción del ULV.	128
5.36. Señales de control del ULV.	129
A.1. Algoritmo propuesto para controlador subóptimo adaptable.	149
A.2. Diagrama de flujo interfaz gráfica.	164
A.3. Diagrama de flujo UAV.	165
A.4. Diagrama de flujo ULV.	166
A.5. Diagrama controlador PID híbrido autosintonizable.	167
B.1. XBee PIX-HAWK® pinout.	168
B.2. RPLidar PIX-HAWK® pinout.	168
B.3. Conexión sensor PX4-Flow y controlador PIX-HAWK®.	169
B.4. ESP32 wrover pin-out.	169
B.5. Sensor de altura del carro transversal del rover.	169
B.6. Driver de los motores del ULV.	170
B.7. Smart driver de los motores del ULV.	170
B.8. Conexión PIX-HAKW® motores para motricidad del ULV.	171
B.9. PIX-HAWK® elementos y componentes.	172
B.10. Movimientos del UAV en función de las velocidades de sus motores.	173

B.11.Movimientos del ULV en función del giro de sus motores.	174
B.12.Curvas de comportamiento gases detectados por el sensor MQ135.	175
B.13.Sensor MQ135.	175
B.14.Plano objeto 3D para LABview®.	176
B.15.Plataforma experimental UAV vistas.	177
B.16.Plataforma experimental UAV diseño.	178
B.17.Plataforma experimental ULV cotas horizontales.	179
B.18.Plataforma experimental ULV cotas verticales.	180
B.19.Plataforma experimental ULV diseño.	181
B.20.gimbal UAV diseño.	182
B.21.Esquema de tarjeta electrónica RAS-PIX.	183
B.22.Esquema de tarjeta electrónica usada para control de gimbal.	184

Índice de cuadros

2.1. Descripción de los parámetros Denavit-Hartenberg.	24
2.2. Filtro de Kalman Extendido	43
4.1. Tabla de Parámetros D-H del Rover.	88

Capítulo 1

Introducción

Este trabajo está enfocado al desarrollo de una plataforma robótica móvil heterogénea para monitoreo ambiental. En el presente capítulo se explica brevemente el contenido de este trabajo, el cual está dividido en seis capítulos y dos anexos, el primer capítulo es llamado “Introducción” en él se da a conocer de forma breve el estado del arte relacionado robots móviles, aéreo y terrestre, sus modelos, el planteamiento del problema, la justificación, la hipótesis, el objetivo general los específicos y la metodología a seguir para el desarrollo del trabajo.

El segundo capítulo denotado como “Marco Teórico”, aborda brevemente las principales herramientas teóricas empleadas en el desarrollo de los algoritmos y modelos empleados en del desarrollo del presente trabajo. Los cuales son el método de Denavit-Hartenberg, el controlador PID, el Método de Lucas-Kande, el filtro de Kalman y su extensión, además, del Método del gradiente descendente.

En el tercer capítulo “Plataforma Robótica Móvil Heterogénea”, se da una descripción del sistema de la plataforma propuesta, y sus componentes principales, el robot aéreo, el robot terrestre y el sistema SCADA.

En el cuarto capítulo, el lector podrá hallar los dos modelos matemáticos usados a lo largo de este trabajo, para manipular y simular el comportamiento de los dos robots.

El quinto capítulo que es llamado “Simulaciones y resultados experimentales”, muestra las simulaciones del seguimiento de trayectoria de ambos robots, la simulación de un

control subóptimo autosintonizable que es una contribución novedosa de este trabajo, los resultados experimentales en el seguimiento de la trayectoria del robot aéreo y del robot terrestre.

El último capítulo que es el sexto, muestra las conclusiones y trabajos futuros propuestos, derivados de los resultados experimentales.

Los dos anexos muestran las estrategias de control a usar, métodos, y modelos, y por últimos planos y esquemas.

El presente proyecto de investigación tiene como propósito aplicar leyes de control en una plataforma de robótica móvil heterogénea, la cual sirve como un sistema de monitoreo para tomar muestras y lecturas, que permitan detectar contaminantes ambientales de tipo aéreos y terrestres, empleando para ello dos robots: uno aéreo no tripulados (UAV), y vehículo terrestre también no tripulado (ULV), para tomar muestras en cada uno de los ambientes antes mencionados, que permitan generar datos estadísticos de la presencia de agentes nocivos en zonas de interés, lo anterior buscando que la plataforma pueda ser aplicada, en ecosistemas importantes para la conservación del medio ambiente, como presas, zonas mineras, ríos, lagos, etc. Con la intención de detectar la presencia de agentes contaminantes que puedan afectar directa o indirectamente a las personas o seres vivos que interactúan con ecosistemas claves para el desarrollo de la vida.

Para realizar esta investigación se proponen sistemas de recolección de muestras en cada uno de los robots, que interactúan con un sistema de Supervisión Control y Adquisición de Datos (Supervisory Control And Data Acquisition) SCADA para monitorear y almacenar información generando bases de datos utilizando los programas LABview® y Unity®, para el caso de las lecturas de contaminación aérea se embarca un sistema de monitoreo en un robot aéreo en configuración de cuatro rotores, y para la contaminación terrestre se crea un sistema que permite tomar muestras de suelo, este sistema se encuentra en el robot todoterreno, con arquitectura Shrimp Rover, posteriormente la información y muestras obtenidas por cada robot, respectivamente, podrán ser estudiadas, fuera de línea por personal especializado en contaminantes.

Con todo lo anterior se puede por ejemplo, generar mapas de contaminación de territo-

rios con alto índice de contaminación, primero acotándolo a zonas donde haya alto grado de daño, para aplicar acciones que permitan frenar o erradicar las afectaciones causadas por contaminantes que sean altamente agresivos para la salud o los ecosistemas en cuestión, y después buscando aplicarlo de forma tal que se generen procedimientos generalizados que permitan decrecer la contaminación.

Los altos índices de contaminación en la zona centro del país, causados principalmente por la ciudad de México afectan de forma muy agresiva a los estados vecinos a dicha región. A diario, la generación de contaminantes que dañan presas, lagunas, mantos acuíferos, suelo, aire, ríos, etc. es creciente, sin tener un registro claro y objetivo del grado de alteración nociva, provocado a especies de flora y fauna pertenecientes a dichas zonas, además de problemas de salud pública. Por ello, es importante aproximar de forma directa y cuantitativa el grado de afectación que los principales contaminantes están causando, crear un sistema de monitoreo basado en adquisición de datos, mediante la implementación de robots que puedan tomar muestras y lecturas directas de contaminantes en lugares de difícil acceso, coadyuva en la generación de planes y objetivos, prácticos y concretos, en el control y manejo de contaminantes a través de estrategias, basadas en análisis estadístico.

Por lo anterior la motivación del presente trabajo, está enfocada en la creación de un sistema de robótica móvil heterogénea que permita monitorear contaminantes presentes en ecosistemas de vital importancia. El enfoque teórico de la propuesta se soporta en la aplicación de leyes de control aplicadas a cada uno de los robots que forman parte del sistema, donde al aplicarlas se busca mejorar el desempeño de cada sistema robótico desde una perspectiva energética, brindado la posibilidad de incrementar los tiempos de operación, además de generar un trabajo colaborativo entre los robots para optimizar el uso de recursos, ya que por ejemplo, con el cuadricóptero se envían coordenadas a alcanzar al carro todoterreno, sin que este último tenga que hacer recorridos aleatorios antes de tomar muestras.

La aplicación de controladores optimizados permite disminuir considerablemente el consumo energético en sistemas complejos como un vehículo aéreo no tripulado (UAV)[70][71][75][80], y en vehículos terrestres no tripulados (ULV) tal y como es demostrado en [3] para el drone y en [58] para el carro todoterreno.

La integración de robots no tripulados en plataformas que solucionan problemas como la afectación de la contaminación es de alta relevancia, ya que con esto se demuestra que la aplicación de estas tecnologías es viable en problemas de la vida real, y permiten, mejorar en gran medida el desempeño de los sistemas de monitoreo, brindando la posibilidad de incrementar la habilitación de México en el combate de problemas ambientales, aplicando conocimientos científicos y tecnológicos, además, contar con esta plataforma permite que la toma de muestras y adquisición de datos no represente un riesgo para el usuario, ya que éste no tiene que estar en contacto directo con probables agentes nocivos.

1.1. Información preliminar.

En este apartado se muestran los modelos matemáticos a emplear para aplicar leyes de control, a cada uno de los robots que forman parte del sistema general. Los tipos de robots a usar son: Vehículo Aéreo no Tripulado del inglés Unmanned Aerial Vehicle (UAV) en modo mini helicóptero de 4 rotores y un Vehículo Terrestre no Tripulado, Unmanned Land Vehicles (ULV).

1.1.1. Modelo matemático de un UAV

El modelo no lineal simplificado de un UAV presentado en [4] está definido por el siguiente conjunto de ecuaciones no lineales:

$$\begin{aligned}
 m\ddot{x} &= -u \operatorname{sen}(\theta), \\
 m\ddot{y} &= u \cos(\theta) \operatorname{sen}(\phi), \\
 m\ddot{z} &= u \cos(\theta) \cos(\phi) - mg, \\
 \ddot{\phi} &= \tau_\phi, \\
 \ddot{\theta} &= \tau_\theta, \\
 \ddot{\psi} &= \tau_\psi,
 \end{aligned} \tag{1.1}$$

Donde la posición del UAV está dada por las coordenadas x , y , z ; y la orientación del robot es descrita por los ángulos θ , ϕ y ψ , siendo el primero el cabeceo (pitch), el segundo el ángulo de alabeo (roll) y el último la guiñada (yaw). u es la fuerza que empuja el robot hacia arriba; τ_θ , τ_ϕ y τ_ψ son los torques que permiten modificar la orientación relativa a

cada uno de los ángulos. En la Figura 1.1 son mostradas cada una de las variables de posición y orientación del este robot móvil [69].

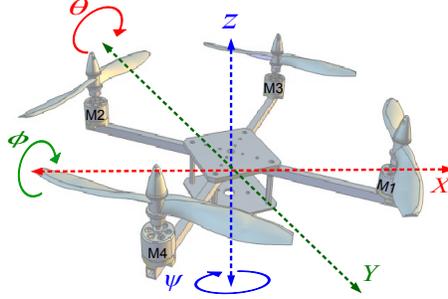


Figura 1.1: Posiciones y ángulos de un UAV.

1.1.2. Modelo matemático de un ULV

El modelo matemático empleado para un robot móvil con configuración diferencial es mostrado en [14], dicho modelo es el siguiente: sea $p \in \mathbb{R}^n$ un vector de coordenadas generalizadas y $q \in \mathbb{R}^m$ un vector con las variables de control, asumiendo que \dot{p} y \dot{q} son la respectivas derivadas de dichos vectores. El modelo directo se puede obtener mediante la matriz jacobiana dada por la siguiente ecuación:

$$\dot{p} = J(p)\dot{q} \quad (1.2)$$

La expresión anterior puede ser escrita como:

$$\dot{p} = f(p) + \sum_{i=1}^m g(p)_i \dot{q}_i \quad (1.3)$$

Cuya representación matricial es:

$$\dot{p} = \begin{bmatrix} -\text{sen}(\phi) \\ \cos(\phi) \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \bar{\omega} \quad (1.4)$$

donde v es la velocidad lineal del vehículo y $\bar{\omega}$ es la velocidad angular, la ecuación anterior se puede expresar de la siguiente manera:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} -\text{sen}(\phi) & 0 \\ \text{cos}(\phi) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \bar{\omega} \end{bmatrix} \quad (1.5)$$

Respecto a un plano de trabajo, las coordenadas de la posición del robot son x , y ; su orientación está determinada por el ángulo ϕ , V_I y V_D representan la velocidad de cada una de las ruedas del robot para el lado izquierdo y derecho, respectivamente, en la Figura 1.2, son ilustradas cada una de las variables antes mencionadas.

Leyes de control como el Regulador Cuadrático Óptimo (LQR), el Regulador Cuadrático Gaussiano (LQG), LQR más acción integral, entre otras, han sido empleadas recientemente en la manipulación de robots móviles [4], lo anterior es analizado con detalle en el siguiente apartado.

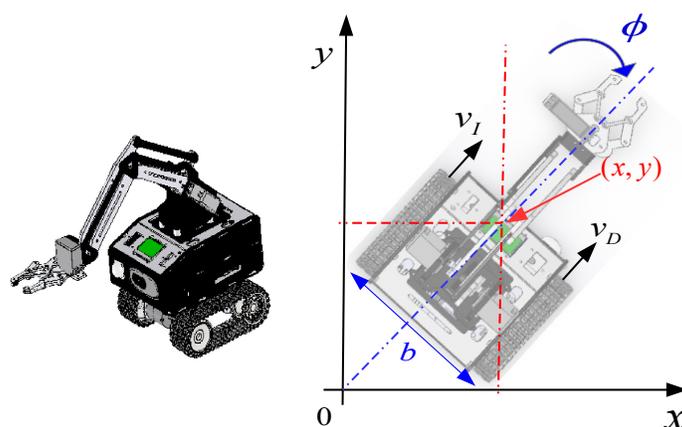


Figura 1.2: Posición y orientación de un ULV.

1.1.3. Controladores aplicados a robots no tripulados tipo UAV y ULV

Debido a que la aportación a realizar en este trabajo está enfocada al control de robots aéreo y terrestre móviles, con el objetivo de ser aplicados en un sistema de detección de contaminantes, a continuación se mencionan las principales aportaciones en el campo del control de robots y los controladores más empleados para ellos, buscando enfatizar la

aplicación de leyes aplicadas a los robots con las arquitecturas presentadas en este trabajo.

Primero se comentan estrategias de control aplicadas a los UAVs y posteriormente a los ULV. La aplicación de leyes de control óptimo aplicado a vehículos aéreos no tripulados (UAV) y vehículos terrestres no tripulados (ALV)[27], ha sido estudiada recientemente; sin embargo, esto representa todo un desafío para la teoría de control, el controlador Proporcional Integral Derivativo (PID) es empleado ampliamente en UAVs y ULVs. El controlador PID puede ser sintonizado de forma heurística, lo que permite que sea aplicado fácilmente, además, de poder ser aplicado en sistemas lineales como no lineales, para los primeros existen diversos métodos que permiten sintonizar este tipo de controlador. La presencia de incertidumbres causa que no pueda asegurarse que el error sea cero, y es uno de los problemas más perjudiciales; sin embargo, gracias a la acción integral de este controlador que permite corregir el error en estado estacionario [1] dando robustez al lazo de control.

Como es bien sabido, el PID es una poderosa herramienta, a pesar de que para algunas tareas específicas no es eficiente, puede ser implementado modificando los parámetros de control bajo un esquema adaptativo o agregando términos no lineales. La ventaja que presenta la acción integral del PID puede ser combinada con controladores óptimos, con el fin de mejorar el desempeño en la estabilización de sistemas.

Leyes de control que integran acción integral han sido exploradas en algunos trabajos reportados. Pude citarse [5] donde una estrategia de control no lineal integral predictiva es presentada para seguir una trayectoria empleando un mini robot aéreo de cuatro rotores, la estructura de control propuesta es un esquema de control predictivo, con el fin de seguir la referencia de una trayectoria junto con un controlador no lineal de horizonte infinito H_∞ , y estabilizar los movimientos rotacionales del vehículo considerado, en dicho trabajo son reportados resultados experimentales exitosos.

En [6] un controlador “backstepping” integral y un “motion planing” son combinados para estabilizar un helicóptero mediante la estabilización de dirección punto a punto, la simulación con el fin de mostrar el comportamiento del desempeño del sistema y su robustez son presentados. La misma técnica es mostrada en [38], con el objetivo de controlar la orientación, altura y posición, en despegue, hovering y aterrizaje son presentados en un

dispositivo OS4. Una comparación entre un controlador PID, un Regulador Cuadrático Lineal (LQR), y controladores no lineales (Adaptive Integral Backstepping Controller) se presentó en [8], se emplea una aproximación para el control no lineal basada en una metodología recursiva con el enfoque de Lyapunov, empleando la técnica backstepping y un esquema adaptativo, fueron publicados satisfactoriamente resultados de simulación y experimentales.

En [4] un control LQR fue empleado para estabilizar la altura y la orientación de un mini robot aéreo de ocho rotores, mediante una simulación numérica es demostrada la eficiencia de la estrategia de control bajo condiciones nominales, los autores implementaron el control LQR agregando una acción integral aplicada a la altura. En [10] es presentado un LQR con un término integral agregado, con el objetivo de mejorar el desempeño deseado del sistema de control, los estados no medibles son estimados mediante un observador, son mostrados resultados experimentales satisfactorios. Además de la aplicación de leyes de control aplicadas a UAVs, es posible encontrarlas aplicadas también a AUVs en [28] un controlador PID robusto es aplicado a un submarino, de una forma similar, un controlador PID auto ajustable, aplicado a un vehículo autónomo submarino en [13].

Los robots no tripulados brindan la ventaja de poder acceder a lugares de difícil acceso, evitando poner en riesgo al usuario, la aplicación de un UAV para detectar contaminante en aire es abordada en [12] y en [11], con los resultados obtenidos se crean mapas de contaminación aérea, con ello se muestra la factibilidad de aplicar robots móviles en entornos contaminados para generar información relativa a niveles de agentes contaminantes patógenos.

Con respecto a los ULV, se puede comentar que en la última década, ha habido un avance significativo en el desarrollo y aplicación de leyes de control en vehículos robóticos diseñados para operar en terrenos difíciles, particularmente en el diseño conocido como Shrimp Rover. Estos vehículos están ideados para funcionar eficazmente en entornos desafiantes, aprovechando arquitecturas innovadoras y estrategias de control.

El trabajo de Estier et al. [54] presenta una de las primeras arquitecturas del rover tipo Shrimp, resaltando su capacidad para misiones de largo alcance en Marte. Este enfoque ha sentado las bases para posteriores desarrollos en el diseño y control de estos rovers.

Yang et al. [55] proporcionan una revisión detallada de la suspensión tipo rocker-bogie, un elemento clave en el diseño de rovers para terrenos difíciles, destacando su importancia en el Shrimp Rover. Esta suspensión facilita la adaptación a terrenos irregulares, esencial para la recolección de muestras de suelo y el análisis de contaminación. Hashim [56] examina la representación gráfica y los perfiles de superficie terrestre en relación con un robot diseñado al estilo Shrimp, subrayando la importancia de una navegación precisa en terrenos impredecibles.

Shahnawaz et al. [57] realizan un análisis cinemático del ángulo de inclinación en un Shrimp Rover en terreno no estructurado, fundamental para mantener la estabilidad y eficiencia en la recogida de muestras en terrenos accidentados. Lopez-Franco et al. [58] describen un enfoque de control neural óptimo inverso para un robot tipo Shrimp, vital para la autonomía y adaptabilidad del rover en la ejecución de tareas. Dian et al. [59] y Lamon et al. [60] aportan mejoras en el control de torque de ruedas y sistemas de monitoreo móvil, respectivamente, mejorando el rendimiento del Shrimp Rover en terrenos desafiantes. Además, Michaud et al. [61] y Murambikar et al. [62] exploran el uso de energía solar en rovers y la fabricación de mecanismos tipo rocker bogie, ampliando la viabilidad y sostenibilidad de estos vehículos en misiones prolongadas.

Estos trabajos demuestran un progreso notable en el desarrollo de leyes de control y arquitecturas para el Shrimp Rover. Su habilidad para operar en terrenos de difícil acceso no sólo abre posibilidades para la exploración espacial, sino también para aplicaciones terrestres como la toma de muestras de suelo y el análisis de contaminación en entornos previamente inaccesibles.

Todos los trabajos relacionados con UAVs y ULVs antes mencionados, permiten intuir que es viable aplicar leyes de control a estas plataformas de robots no tripulados, buscando optimizar su desempeño y dando la posibilidad de ser aplicados, de forma autónoma, en el monitoreo de contaminantes en regiones de interés, a pesar de la topología del terreno o las distancias a cubrir en dicha tarea.

1.2. Planteamiento del problema

En el contexto actual, las grandes concentraciones de población están generando contaminantes que impactan negativamente tanto en la salud humana como en los ecosistemas naturales. Esto ha llevado a problemas medioambientales graves, como la extinción de especies, la contaminación de cuerpos de agua, el calentamiento global y la disminución de la capa de ozono. Sin una intervención efectiva, estos problemas podrían escalar a consecuencias catastróficas e irreversibles para todas las especies, incluyendo la humanidad.

Aunque existen sistemas de monitoreo de contaminación que utilizan robots aéreos no tripulados, estos sistemas tienen limitaciones significativas. No permiten una caracterización detallada en zonas específicas de interés, ya que las mediciones para identificar contaminantes suelen basarse en lecturas de áreas cercanas y aproximaciones estadísticas. Esto impide determinar con precisión el nivel de concentración de contaminantes nocivos en sitios críticos como presas, lagunas o ríos, donde la afectación a los seres vivos es más intensa.

Además, actualmente no hay un sistema eficiente de monitoreo que combine la vigilancia aérea y terrestre para abordar esta problemática de manera integral. En este trabajo, se propone la implementación de robots terrestres y aéreos para un monitoreo más preciso y detallado de la contaminación, lo que permitiría una mejor evaluación y mitigación de estos riesgos ambientales, anterior a nivel regional.

1.3. Justificación

Un sistema robotizado y automatizado capaz de monitorear y tomar muestras de regiones contaminadas, tanto desde perspectivas aéreas como terrestres, es esencial para generar estrategias efectivas destinadas a reducir o erradicar problemas de contaminación. Mediante el análisis estadístico de muestras cuantitativas, se puede determinar de manera objetiva el grado de afectación de diversos contaminantes aéreos y terrestres, lo que facilita la creación de mapas de contaminación basados en mediciones tomadas directamente en los lugares afectados.

Este enfoque permite a los expertos en medio ambiente, del estado de Hidalgo, desarrollar estrategias específicas para controlar y mitigar la presencia de contaminantes, lo que a su vez disminuiría sus efectos negativos en el medio ambiente. Además, la posibilidad de aplicación de leyes de control a robots que forman parte del sistema de monitoreo contribuirá a optimizar el uso energético e incrementar los tiempos de trabajo de los dispositivos, mejorando su rendimiento y reduciendo riesgos para los usuarios que toman las muestras, ya que no tendrían que estar en contacto directo con los contaminantes, usando robots no tripulados y técnicas de control, para manipular a éstos, permitiendo un mejor desempeño en sus tareas. En resumen, la implementación de un sistema robotizado y automatizado para el monitoreo y muestreo de regiones contaminadas es una herramienta clave para enfrentar problemas de contaminación de manera más efectiva, protegiendo tanto al medio ambiente como la salud de las personas involucradas en la recolección de muestras.

1.4. Hipótesis

La implementación de un sistema integrado de monitoreo ambiental, utilizando robots no tripulados aéreos y terrestres, permitirá una caracterización más precisa y detallada de los niveles de contaminación en áreas específicas, como presas, lagunas y ríos. Este enfoque no sólo mejorará significativamente la capacidad de identificar y cuantificar los contaminantes en comparación con los métodos actuales, sino que también garantizará la seguridad e integridad del usuario final al evitar la exposición directa a entornos potencialmente peligrosos. Como resultado, se espera contar con una plataforma robótica móvil heterogénea, que coadyuve a una gestión más eficaz de los riesgos ambientales y una mejor protección de la salud humana y los ecosistemas.

1.5. Objetivo general

Desarrollar una plataforma robótica heterogénea de monitoreo ambiental, mediante la integración de un sistema SCADA y dos robots móviles, uno aéreo y otro terrestre, no tripulados, para el monitoreo de calidad del aire y toma de muestras de suelo, que podrán ser analizadas fuera de línea por personal especializado.

1.5.1. Objetivos específicos

- Desarrollar una plataforma robótica aérea basada en un cuadricóptero que permita hacer lecturas sobre la calidad del aire utilizando un sistema de monitoreo integrado a bordo. Este sistema generará información estadística a partir de las lecturas realizadas durante el vuelo, de los niveles de CO₂ del ambiente. La plataforma también contará con sistema GPS y comunicación remota con un sistema SCADA por radiofrecuencia.
- Desarrollar un sistema robótico a ruedas que pueda recolectar muestras de tierra en diferentes ambientes y permita a personal capacitado analizar la presencia de contaminantes en dichas muestras. Esto se logrará mediante la aplicación de algoritmos de control que permitan su interacción con un sistema SCADA y la capacidad de tomar muestras de suelo y seguir trayectorias utilizando el sistema de posicionamiento global (GPS). Además, el robot contará con comunicación RF para interactuar remotamente con el usuario.
- Programar un sistema SCADA basado en LABview®(R), que permita la interacción, configuración y coordinación remota de dos robots: un cuadricóptero y un robot a ruedas, mediante comunicación inalámbrica. Este sistema permitirá que ambos robots trabajen colaborativamente, siendo la interfaz de comunicación y gestión entre ellos y el usuario.
- Sintetizar leyes de control y aplicarlas a las plataformas robóticas para lograr un control y desempeño adecuado en las tareas que deben realizar.

1.6. Metas.

Para lograr realizar el sistema se proponen las siguientes metas:

- Aprender a utilizar Python y la librería OpenCV para procesamiento de imágenes en Raspberry Pi 3B y luego integrar la computadora al robot terrestre para fines específicos en un plazo de 6 meses.
- Armado de plataformas experimentales (robots aéreo y terrestre), integrando tecnología existente y haciendo las adecuaciones requeridas para la integración de todo el sistema, en 3 años.

- Investigar el modelado matemático de los robots como sistemas dinámicos para sintetizar leyes de control en ellos.
- Adquirir y acondicionar señales de variables físicas para hacer aproximaciones del grado de contaminación del aire con un alto grado de confiabilidad implementado a bordo del UAV en un plazo de 1 año
- Programar interfaces ergonómicas de adquisición de datos basadas en un sistema SCADA programado en LABview®[®], capaces de interactuar con los 2 robots que forman parte del sistema de monitoreo y adquisición de datos en un plazo de 3 años.
- Desarrollar el sistema de recolección de muestras de suelo, del robot terrestre, en un plazo de 6 meses.
- Programar leyes de control en las tarjetas electrónicas de los robots, específicamente en la tarjeta PIX-HAWK®[®], para manipular su desempeño en un plazo de 12 meses.

1.7. Limitaciones

Para el presente trabajo se contempla crear un sistema de monitoreo y recolección de contaminantes ambientales empleando para ello un sistema heterogéneo basado en 2 robots; un drone y un carro todoterreno, que mediante una interfaz SCADA interactúan con el usuario. La intención es aplicar leyes de control a cada robot, para mejorar su desempeño y ahorro energético, mediante algoritmos como PID, LQR, LQG.

Las principales restricciones que se tienen son las siguientes: el alcance planeado para cada robot respecto a la ubicación del usuario es de 20 m, el UAV monitorea únicamente como contaminante aéreo, la cantidad de CO_2 , no recolecta muestras de aire, lo realiza en el lugar donde vuela mediante un sensor embarcado en él. Para el caso el vehículo todoterreno se realiza la recolección de muestra de tierra que son procesadas por personal capacitado en un laboratorio.

Las muestras a recolectar se encontrarán en un rango establecido por la resolución del sensor GPS del robot que envía la referencia al sistema SCADA, para este caso el robot aéreo, por ello dichas muestras se realizan en un punto correspondiente a la zona estable-

cida por dicho robot.

Los robots propuestos están basados en prototipos de bajo costo, por lo que el alcance, potencia y duración de baterías están limitados, por ello es importante aplicar leyes de control que permitan un ahorro energético, optimizando el desempeño de dichas plataformas.

El análisis de contaminantes requiere de la interpretación de personal capacitado en esa área, por ello este trabajo se desarrollará con la asesoría técnica de profesionales capacitados en ese ámbito.

Los algoritmos a desarrollar están basados en programación usando librerías de código abierto, por lo que todo material generado a través de éstas está sujeto a las licencias que las rijan.

1.8. Experimentos realizados

Los experimentos a realizar en el presente trabajo son listados a continuación:

- Implementación de sistema SCADA para configuración, monitoreo y control de los robots móviles.
- Realizar control de posición y orientación del robot aéreo.
- Realizar control de posición y orientación del robot terrestre.

1.9. Conclusión capítulo 1

En conclusión, este capítulo ha logrado abordar con éxito aspectos fundamentales en el campo de la robótica aérea y terrestre móvil no tripuladas. Se ha presentado una sólida introducción general, seguida de una detallada exposición de los modelos matemáticos usados en UAV y ULV, así como de los controladores aplicados a estos dispositivos. El planteamiento del problema, junto con la justificación, hipótesis y objetivos, ha establecido claramente la dirección y relevancia del estudio. La metodología propuesta, las metas

establecidas y las limitaciones reconocidas han demostrado un enfoque riguroso y realista. Finalmente, los experimentos previstos prometen aplicar y validar los conceptos teóricos en entornos prácticos, consolidando así la importancia de esta investigación en el ámbito de la robótica, aplicada al monitoreo de contaminantes.

Capítulo 2

Marco teórico

En este capítulo, se abordan los temas principales requeridos para el desarrollo de esta tesis, siguiendo un orden lógico y estructurado. Se inicia con el control integral, después, se muestra el Método de Denavit-Hartenberg, enfocándose en la selección de sistemas de referencia y en el propio algoritmo Denavit-Hartenberg. A continuación, se explora el Controlador PID, seguido de una detallada exposición del Método de Lucas-Kanade, con énfasis en el flujo óptico. Posteriormente, se profundiza en el Filtro de Kalman y su extensión, el Filtro de Kalman extendido. Finalmente, se examina el Método del gradiente descendente. Estos temas constituyen los pilares fundamentales sobre los cuales se construye el análisis y desarrollo de esta investigación.

2.1. Control integral

Considerando el sistema no lineal:

$$\dot{\bar{x}} = f(\bar{x}, \bar{u}), \quad (2.1)$$

$$\bar{y} = h(\bar{x}), \quad (2.2)$$

donde el estado $\bar{x} \in \mathbb{R}^n$, y el vector de control $\bar{u} \in \mathbb{R}^p$. Las variables f y h son funciones continuamente diferenciables en un dominio $D_{\bar{x}} \times D_{\bar{u}} \subset \mathbb{R}^n \times \mathbb{R}^p$, y $\bar{y} \in \mathbb{R}^p$ es la salida controlada. Sea $\bar{y}_R \in \mathbb{R}^p$ una referencia constante; el control integral es un estado de retroalimentación tal que

$$\bar{y}(t) \rightarrow \bar{y}_R, \quad t \rightarrow \infty \quad (2.3)$$

se supone que la salida controlada \bar{y} puede ser medida. La tarea de regulación se logrará estabilizando el sistema en un punto de equilibrio donde $\bar{y} = \bar{y}_R$. Para mantenerlo en esa condición de equilibrio, existe un par $(\bar{x}_{ss}, \bar{u}_{ss}) \in D_{\bar{x}} \times D_{\bar{u}}$ tal que:

$$0 = f(\bar{x}_{ss}, \bar{u}_{ss}), \quad (2.4)$$

$$0 = h(\bar{x}_{ss}) - \bar{y}_R. \quad (2.5)$$

Se asume que estas ecuaciones tienen una solución única $(\bar{x}_{ss}, \bar{u}_{ss})$. Ahora, la acción integral se incluye de la siguiente manera: consideremos el error de seguimiento $\bar{e} = \bar{y} - \bar{y}_R$. Entonces, se define la siguiente equivalencia

$$\dot{\sigma} = \bar{e} = \bar{y} - \bar{y}_R, \quad (2.6)$$

Así, el control se obtendrá como una función de retroalimentación de \bar{x} y $\bar{\sigma}$ de tal manera que en el bucle cerrado haya un punto de equilibrio $(\bar{x}, \bar{\sigma})$ con $\bar{x} = x_{ss}$. Suponiendo que el sistema es linealizable alrededor de $x_{ss}, \bar{\sigma}, u_{ss}$, se sigue que:

$$\dot{\bar{\xi}} = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix} \bar{\xi} + \begin{bmatrix} B \\ 0 \end{bmatrix} v \triangleq \mathcal{A}\bar{\xi} + \mathcal{B}v, \quad (2.7)$$

con

$$\bar{\xi} = \begin{bmatrix} \bar{x} - x_{ss} \\ \bar{\sigma} - \sigma_{ss} \end{bmatrix}, \quad (2.8)$$

$v = \bar{u} - u_{ss}$ donde:

$$A = \left. \frac{\partial f}{\partial \bar{x}} \right|_{(\bar{x}, \bar{u})=(x_{ss}, u_{ss})}, B = \left. \frac{\partial f}{\partial \bar{u}} \right|_{(\bar{x}, \bar{u})=(x_{ss}, u_{ss})}, \quad (2.9)$$

$$C = \left. \frac{\partial h}{\partial \bar{x}} \right|_{\bar{x}=x_{ss}}$$

Suponiendo que el par (A, B) es controlable y

$$\text{rank} \begin{bmatrix} A & B \\ C & 0 \end{bmatrix} = n + p. \quad (2.10)$$

Entonces $(\mathcal{A}, \mathcal{B})$ es controlable [1]. Por lo tanto, se diseña una matriz \mathcal{K} tal que $\mathcal{A} + \mathcal{BK}$ sea de Hurwitz [1]. Considere la partición para la matriz \mathcal{K} como $[\mathcal{K}_1 \ \mathcal{K}_2]$. La señal de control se define entonces por:

$$\bar{u} = \mathcal{K}_1(\bar{x} - x_{ss}) + \mathcal{K}_2(\bar{\sigma} - \sigma_{ss}) + u_{ss}. \quad (2.11)$$

No es difícil verificar que el sistema no lineal en bucle cerrado tiene un único punto de equilibrio $(x_{ss}, \bar{\sigma}_{ss})$ [1]. Como se demuestra en [1], el punto de equilibrio $(x_{ss}, \bar{\sigma}_{ss})$ es exponencialmente estable, y todas las soluciones que comienzan lo suficientemente cerca de este punto de equilibrio se acercan a él a medida que t tiende al infinito. Entonces $\bar{y}(t) - \bar{y}_R \rightarrow 0$ a medida que $t \rightarrow \infty$. Siguiendo estas ideas, en esta contribución, la matriz \mathcal{K} se diseña utilizando el enfoque de control óptimo. Para la altitud, aprovechamos el hecho de que en el modelo matemático para la altitud del cuadricóptero, se puede estabilizar mediante linealización exacta, y entonces, no es necesario linealizar la dinámica no lineal. La acción integral se utiliza para minimizar los efectos de las perturbaciones externas en el rendimiento del vehículo. La Figura 2.1 muestra el esquema básico para el control de la acción integral.

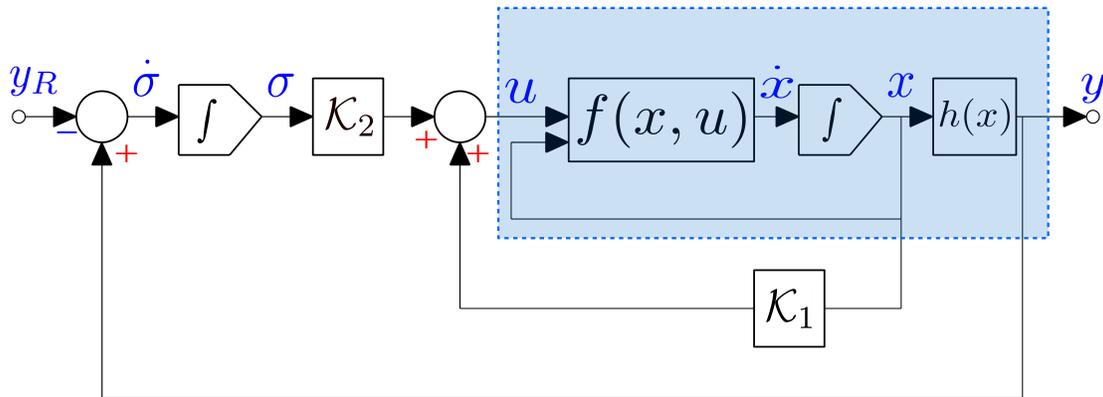


Figura 2.1: Esquema de control integral[1].

2.2. Método de Denavit-Hartenberg

El método de Denavit-Hartenberg es una herramienta ampliamente conocida en el área de ingeniería, ya que ofrece un procedimiento sencillo para obtener el modelo cinemático directo cuya estructura queda en términos de las transformaciones homogéneas. Jaques Denavit y Richard S. Hartenberg en 1955 presentaron un procedimiento para obtener una

representación mínima de la orientación y traslación de robots manipuladores. Consiste en determinar una tabla de parámetros relacionados con los eslabones del robot. La convención Denavit-Hartenberg toma como referencia el diagrama de un robot manipulador en cadena cinemática abierta, como se muestra en la Figura 2.2.

Las variables articulares en la representación Denavit-Hartenberg se denotan con θ_i para el tipo rotacional, prismática o lineal por d_i ; este parámetro d_i también hace el papel de representar el ancho del servomotor de la articulación rotacional más el espesor de la placa metálica del eslabón, en este caso se denota por el símbolo β_i ; la longitud del eslabón se representa con l_i y el ángulo de separación entre los ejes z_i y z_{i-1} se denota con α_i .

El ángulo θ_i es el ángulo entre los ejes x_{i-1} y x_i medido alrededor del eje z_{i-1} . El parámetro d_i es la distancia desde el origen del sistema de referencia $i-1$ hasta la intersección del eje x_i con el eje z_{i-1} . La medición de d_i se realiza a lo largo del eje z_{i-1} , como se indica en la Figura 2.3.

Además de las variables articulares θ_i y d_i , hay dos parámetros constantes que describen características específicas del eslabón i -ésimo. Estos parámetros son:

El parámetro l_i se define como la distancia a lo largo del eje x_i desde el origen del sistema de referencia coordinado $i-1$ hasta la intersección del eje z_{i-1} con el eje x_i . El otro parámetro es el ángulo α_i entre los ejes z_i y z_{i-1} , el cual se denota por α_i . Su medición se realiza respecto a un plano normal a x_i . Una medición de ángulo positivo para α_i se toma en dirección del eje z_{i-1} hacia z_i .

Por ejemplo, para un robot con 6 articulaciones rotacionales, se requieren 24 elementos $(l_i, \alpha_i, \beta_i, \theta_i)$ para describir completamente su modelo cinemático [33].

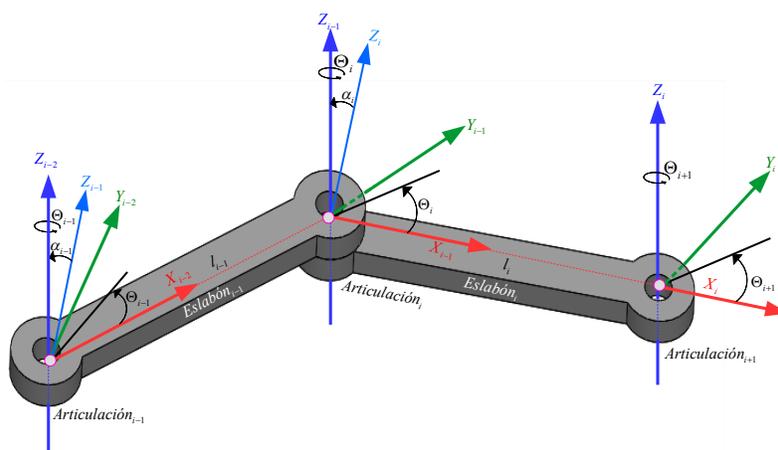


Figura 2.2: Convención Denavit-Hartenberg para un robot manipulador [33].

2.2.1. Selección de sistemas de referencia

En la metodología Denavit-Hartenberg, primero se describirá la convención para asignar los sistemas de referencia cartesianos asociados a los eslabones del robot. En este punto, es necesario aclarar que en la literatura de robótica, la convención Denavit-Hartenberg no es única, depende de la selección de los sistemas de referencia cartesianos en las articulaciones y eslabones, así como en sus eslabones adyacentes.

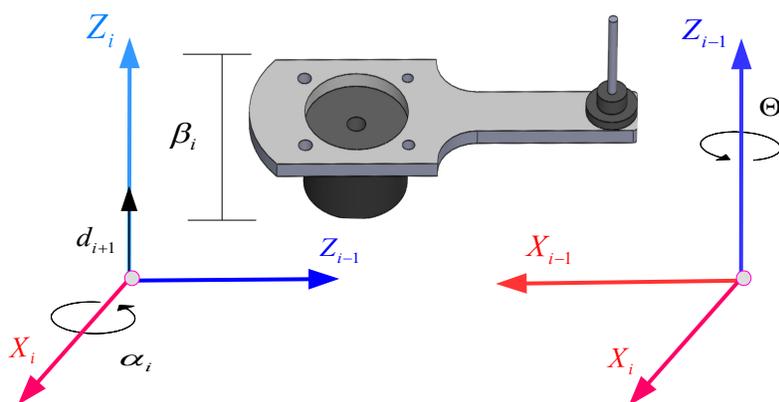


Figura 2.3: Convención para medir θ_i y α_i [33].

La cinemática directa del robot proporciona las coordenadas cartesianas del extremo final del robot con respecto a un sistema de referencia cartesiano fijo $\Sigma_0(x_0, y_0, z_0)$. En la Figura 2.2 se muestra la asignación de sistemas de referencia para las articulaciones

$i - 1$ -ésima, i -ésima e $i + 1$ de un robot manipulador.

En general, se tiene el siguiente procedimiento:

1. El eje z_i se asigna rígidamente a la articulación $i + 1$. Es decir, z_0 es el eje de la articulación 1, z_1 es el eje de la articulación 2, y así sucesivamente.
2. Localizar el origen o_i del sistema de referencia $\Sigma_i(x_i, y_i, z_i)$ en la intersección del eje z_i con la normal común a los ejes z_{i-1} y z_i .
3. Seleccionar el eje x_{i-1} sobre la normal que une los ejes z_{i-1} y z_i en dirección de la articulación $i - 1$ hacia la articulación i .
4. Definir el ángulo de torsión α_i . Este es el ángulo entre los ejes z_i y z_{i-1} y se mide con valor positivo en el sentido de las manecillas del reloj sobre el eje x_i .
5. Seleccionar el eje y_i usando la regla de la mano derecha.

De acuerdo con esta convención previamente descrita, a continuación se resumen los parámetros del i -ésimo eslabón:

1. l_i es la longitud del i -ésimo eslabón, es la distancia del eje z_{i-1} hacia el eje z_i medida sobre el eje x_{i-1} .
2. α_i es el ángulo de torsión, el cual representa el ángulo entre los ejes z_{i-1} y z_i medido en el sentido de las manecillas del reloj sobre el eje x_i .
3. d_i se emplea en articulaciones lineales o prismáticas y representa el desplazamiento lineal. Cuando la articulación es rotacional, entonces representa el offset o espesor del servomotor (la distancia de x_{i-1} a x_i medida sobre el eje z_{i-1}), se denota por β_i .
4. θ_i es el desplazamiento rotacional de x_{i-1} a x_i medido alrededor del eje z_{i-1} . El signo positivo de θ_i es el sentido contrario a las manecillas del reloj.

Obsérvese que l_i y β_i siempre serán positivos, puesto que corresponden a longitudes, mientras que α_i , d_i , θ_i representan cantidades con signo [33].

2.2.2. Algoritmo de Denavit-Hartenberg

A continuación se describe el procedimiento para encontrar la cinemática directa a través de la convención Denavit-Hartenberg:

1. Localizar la dirección de los ejes z_0, z_1, \dots, z_{n-1} .
2. Establecer el sistema de referencia cartesiano fijo $\Sigma_0(x_0, y_0, z_0)$ cuyo origen es colocado sobre el sistema de referencia en la base del robot. Los ejes x_0, y_0 son determinados de acuerdo con la regla de la mano derecha.

Una vez que el sistema de referencia $\Sigma_0(x_0, y_0, z_0)$ ha sido establecido, se inicia un proceso iterativo en el cual se define el sistema de referencia $\Sigma_i(x_i, y_i, z_i)$ usando el sistema de referencia $\Sigma_{i-1}(x_{i-1}, y_{i-1}, z_{i-1})$, iniciando con el sistema de referencia $\Sigma_1(x_1, y_1, z_1)$. En la Figura 2.2 se muestra el procedimiento.

3. Llevar a cabo los pasos 3 al 5 para las articulaciones $i = 1, \dots, n - 1$.
 - a) Localizar el origen o_i en la intersección de la normal común que une al eje z_i con el eje z_{i-1} .
Si el eje z_i intercepta al eje z_{i-1} , colocar o_i en la intersección.
Para el caso en que los ejes z_i y z_{i-1} son paralelos:
Si la articulación i -ésima es rotacional, colocar el origen o_i sobre la articulación i -ésima, tal que $d_i = 0$.
Si la articulación i -ésima es prismática, colocar el origen o_i en un punto límite físico de la articulación i -ésima, por ejemplo, en un punto extremo.
 - b) Seleccionar el eje x_i a lo largo de la normal común que une a los ejes z_{i-1} y z_i , en dirección de la articulación $i - 1$ hacia la articulación i .
 - c) Determinar y_i por la regla de la mano derecha.
4. Establecer el sistema de referencia del extremo final $\Sigma_n(x_n, y_n, z_n)$.
Si la articulación n -ésima es rotatoria, entonces alinear el eje z_n con el eje z_{n-1} .
Si la articulación n -ésima es prismática, entonces seleccionar el eje z_n de forma arbitraria. El eje x_n debe cumplir el paso 4.
5. Establecer la Tabla 2.1 de parámetros de eslabones.

Parámetro	Significado
l_i	Longitud del eslabón i -ésimo
d_i	Articulaciones lineales o prismáticas
α_i	Ángulo entre z_{i-1} y z_i respecto a x_i
θ_i	Ángulo entre x_{i-1} y x_i alrededor de z_{i-1}

Tabla 2.1: Descripción de los parámetros Denavit-Hartenberg.

6. Obtener las matrices de transformaciones homogéneas (H_i^{i-1}) para $i = 1, 2, \dots, n-1$.

En la representación Denavit-Hartenberg, cada transformación homogénea H_i^{i-1} se representa por el producto de cuatro transformaciones básicas:

$$H_i^{i-1} = H_{Rz_{i-1}}(\theta_i) \cdot H_{Tz_{i-1}}(d_i(\beta_i)) \cdot H_{Tx_{i-1}}(l_i) \cdot H_{Rx_{i-1}}(\alpha_i), \quad (2.12)$$

donde:

$$H_{Rz_{i-1}}(\theta_i) = \begin{bmatrix} \cos(\theta_i) & -\text{sen}(\theta_i) & 0 & 0 \\ \text{sen}(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.13)$$

$$H_{Tz_{i-1}}(d_i(\beta_i)) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i(\beta_i) \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.14)$$

$$H_{Tx_{i-1}}(l_i) = \begin{bmatrix} 1 & 0 & 0 & l_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.15)$$

$$H_{Rx_{i-1}}(\alpha_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_i) & -\text{sen}(\alpha_i) & 0 \\ 0 & \text{sen}(\alpha_i) & \cos(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.16)$$

y finalmente:

$$H_i^{i-1} = \begin{bmatrix} \cos(\theta_i) & -\text{sen}(\theta_i) \cos(\alpha_i) & \text{sen}(\theta_i) \text{sen}(\alpha_i) & l_i \cos(\theta_i) \\ \text{sen}(\theta_i) & \cos(\theta_i) \cos(\alpha_i) & -\cos(\theta_i) \text{sen}(\alpha_i) & l_i \text{sen}(\theta_i) \\ 0 & \text{sen}(\alpha_i) & \cos(\alpha_i) & d_i(\beta_i) \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.17)$$

La transformación homogénea total se obtiene como $H_{n0} = H_{10} \cdot H_{21} \cdot \dots \cdot H_{n-2,n-1} \cdot H_{n-1,n}$.

La cinemática directa es la forma general de transformaciones homogéneas que concatenan los sistemas de referencia cartesianos asociados a los eslabones del robot, todos relativos al sistema de referencia fijo Σ_0 [33].

2.3. Controlador PID

Sea $e(t)$ la señal de error, definida como la diferencia entre la referencia $r(t)$ y el valor medido $y(t)$, en un sistema de control en lazo cerrado. Un controlador PID (Proporcional Integral Derivativo) consiste en la sumatoria de tres acciones aplicadas a dicha señal de error, éstas son la acción proporcional, la acción integral y la acción derivativa, las cuales son descritas a continuación.

Acción proporcional: La acción proporcional actúa de forma gradual en función de la magnitud de la señal de error, como lo indica la siguiente expresión:

$$u(t) = K_p e(t) = K_p [r(t) - y(t)], \quad (2.18)$$

donde K_p es la ganancia proporcional, la cual puede definirse como la función de transferencia de un controlador meramente proporcional, es decir:

$$C(s) = U(s)/E(s) = K_p, \quad (2.19)$$

Acción integral: La acción integral genera una señal de salida proporcional a la integral de la señal de error, ésta está dada por la siguiente ecuación:

$$u(t) = K_i \int_0^t e(\tau) d\tau, \quad (2.20)$$

siendo K_i la ganancia integral. En cierta forma esta acción, está relacionada con los valores anteriores del error, ya que podría decirse que es la acumulación de errores anteriores y está en función de éstos. La función de transferencia para la acción integral es:

$$C(s) = U(s)/E(s) = \frac{K_i}{s}, \quad (2.21)$$

Acción Derivativa: Mientras la acción proporcional está basada en los valores actuales de la señal de error y acción integral está basada en los valores pasados de dicha señal, la acción derivativa actúa sobre valores futuros. Una ley de control derivativa ideal puede ser expresada como:

$$u(t) = K_d \frac{de(t)}{dt}, \quad (2.22)$$

K_d es la ganancia derivativa. La correspondiente función de transferencia es:

$$C(s) = U(s)/E(s) = K_d s, \quad (2.23)$$

Defina:

$$K_i = \frac{K_p}{T_i}, \quad (2.24)$$

$$K_d = K_p T_d,$$

donde T_i es el tiempo de integración y T_d es el tiempo de derivación, ambos constantes. La función de transferencia de un controlador PID puede escribirse como:

$$C(s) = K_p + \frac{K_i}{s} + K_d s = K_p \left(1 + \frac{1}{T_i s} + T_d s \right), \quad (2.25)$$

Implementación digital: Para aplicar un control PID de forma digital, se requiere discretizar las leyes de control antes mencionadas. Considere la expresión en tiempo continuo de un controlador PID en su forma ideal:

$$u(t) = K_P \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right), \quad (2.26)$$

Defina ΔT como el tiempo de muestreo. La acción integral en (2.26) puede ser escrita como:

$$\int_0^{tk} e(\tau) d\tau = \sum_{i=1}^k e(t_i) \Delta t, \quad (2.27)$$

donde $e(t_i)$ es el error del sistema en tiempo continuo, en el i -ésimo instante de muestreo. La acción derivativa discretizada vía una aproximación de Euler resulta:

$$\frac{de(t_k)}{dt} = \frac{e(t_k) - e(t_{k-1})}{\Delta t}, \quad (2.28)$$

De lo anterior, la ley de control discretizada es:

$$u(t_k) = K_P \left(e(t_k) + \frac{\Delta t}{T_i} \sum_{i=1}^k e(t_i) + \frac{T_d}{\Delta t} (e(t_k) - e(t_{k-1})) \right), \quad (2.29)$$

De esta manera, el valor de la variable de control es determinada directamente. Alternativamente, la variable de control en el tiempo t_k puede ser calculada empleando su valor anterior $u(t_{k-1})$. Substrayendo $u(t_{k-1})$ de $u(t_k)$, se obtiene:

$$u(t_k) - u(t_{k-1}) = K_P \left[\left(1 + \frac{\Delta t}{T_i} + \frac{T_d}{\Delta t} \right) e(t_k) + \left(-1 - \frac{2T_d}{\Delta t} \right) e(t_{k-1}) + \frac{T_d}{\Delta t} e(t_{k-2}) \right], \quad (2.30)$$

La ecuación (2.30) puede ser reescrita como:

$$u(t_k) - u(t_{k-1}) = K_1 e(t_k) + K_2 e(t_{k-1}) + K_3 e(t_{k-2}), \quad (2.31)$$

siendo:

$$K_1 = K_P \left(1 + \frac{\Delta t}{T_i} + \frac{T_d}{\Delta t} \right), \quad (2.32)$$

$$K_2 = -K_p \left(1 + \frac{2T_d}{\Delta t} \right), \quad (2.33)$$

$$K_3 = K_p \frac{T_d}{\Delta t}, \quad (2.34)$$

Definiendo el operador q^{-1} como el operador de recorrido hacia atrás en tiempo discreto, es decir:

$$q^{-1}u(t_k) = u(t_{k-1}), \quad (2.35)$$

El controlador PID discretizado puede ser escrito como:

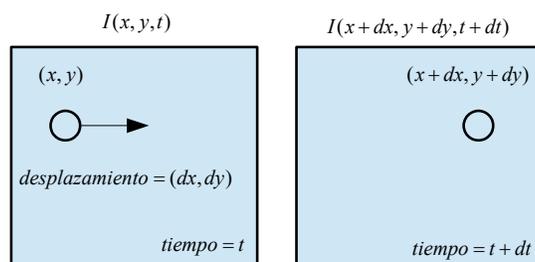
$$C(q^{-1}) = \frac{K_1 + K_2q^{-1} + K_3q^{-2}}{1 - q^{-1}}, \quad (2.36)$$

donde K_1 , K_2 y K_3 pueden ser vistos como los parámetros de sintonización [20].

2.4. Método de Lucas Kanade

2.4.1. Flujo óptico

El flujo óptico, en una comprensión de alto nivel, es el movimiento de objetos entre fotogramas consecutivos de secuencia, causado por el movimiento relativo entre el objeto y la cámara. El problema del flujo óptico se puede expresar como:



problema del flujo óptico

Figura 2.4: Problema del flujo óptico [23].

Donde, entre fotogramas consecutivos, se puede expresar la intensidad de la imagen (I) en función del espacio (x, y) y el tiempo (t) . En otras palabras, si se toma la primera imagen $I(x, y, t)$ y se mueven sus píxeles por (dx, dy) durante t tiempo, se obtiene la nueva imagen $I(x + dx, y + dy, t + dt)$ [23].

Inicialmente se asume que las intensidades de los píxeles de un objeto son constantes entre cuadros consecutivos.

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t), \quad (2.37)$$

En segundo lugar se aplica la aproximación de la serie de Taylor para el lado derecho de la ecuación anterior eliminando términos comunes.

$$I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t) + \frac{\partial I}{\partial x} \delta x + \frac{\partial I}{\partial y} \delta y + \frac{\partial I}{\partial t} \delta t + \dots \quad (2.38)$$

$$\Rightarrow \frac{\partial I}{\partial x} \delta x + \frac{\partial I}{\partial y} \delta y + \frac{\partial I}{\partial t} \delta t = 0, \quad (2.39)$$

En tercer Lugar, se divide todo por dt para derivar la ecuación de flujo óptico:

$$\frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v + \frac{\partial I}{\partial t} = 0, \quad (2.40)$$

donde $u = dx/dt$ y $v = dy/dt$, dI/dx , dI/dy y dI/dt son los gradientes de la imagen a lo largo del eje horizontal, el eje vertical y el tiempo. Por tanto, se concluye con el problema del flujo óptico, es decir, resolviendou(dx/dt) yv(dy/dt) para determinar el movimiento en el tiempo. Puede notar que no se puede resolver directamente la ecuación de flujo óptico para u y v ya que sólo hay una ecuación para dos variables desconocidas. Se implementan algunos métodos, como el de Lucas Kanade, para abordar este problema.

En visión artificial, el método Lucas Kanade es un método diferencial ampliamente usado para estimar el flujo óptico desarrollado por Bruce D. Lucas y Takeo Kanade. Asume que el flujo es esencialmente constante en la vecindad de un píxel en consideración, y resuelve las ecuaciones básicas de flujo óptico para todos los píxeles vecinos, aplicando el criterio de mínimos cuadrados[23].

Al combinar información de varios píxeles cercanos, el método Lucas Kanade frecuentemente resuelve la ambigüedad inherente de la ecuación de flujo óptico. También es más robusto frente al ruido en la imagen que otros métodos locales que se concentran en un punto. Por otro lado, siendo un método puramente local, no provee información de flujo en el interior de regiones uniformes en la imagen, como sí lo hace el método global de Horn-Schunck.

Lucas y Kanade propusieron una técnica eficaz para estimar el movimiento de características interesantes, comparando dos fotogramas consecutivos en su artículo Una técnica de registro de imágenes iterativas con una aplicación a la visión estéreo. El método Lucas Kanade funciona bajo los siguientes supuestos:

1. Dos fotogramas consecutivos están separados por un pequeño incremento de tiempo, dt de modo que los objetos no se desplazan significativamente.

2. Un marco representa una escena natural con objetos texturizados que exhiben tonos de gris que cambian suavemente.

Primero, bajo estos supuestos, se puede tomar una pequeña ventana de 3×3 (vecindad) alrededor de las características detectadas por el método de Shi-Tomasi y asumir que los nueve puntos tienen el mismo movimiento.

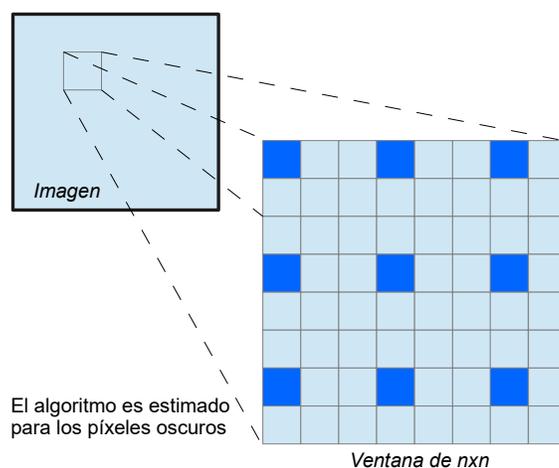


Figura 2.5: Ventana de $n \times n$ píxeles a usar para el algoritmo de Lucas Kanade[23].

Lo anterior puede ser representado como:

$$\begin{aligned}
 I_x(q_1)V_x + I_y(q_1)V_y &= -I_t(q_1), \\
 I_x(q_2)V_x + I_y(q_2)V_y &= -I_t(q_2), \\
 &\vdots \\
 I_x(q_n)V_x + I_y(q_n)V_y &= -I_t(q_n),
 \end{aligned}
 \tag{2.41}$$

donde q_1, q_2, \dots, q_n denota los píxeles dentro de la ventana (por ejemplo, $n = 9$ para una ventana de 3×3) y $I_x(q_i)$, $I_y(q_i)$, y $I_t(q_i)$ denotan las derivadas parciales de la imagen I con respecto a la posición (x, y) y el tiempo t , para el píxel q_i en el instante actual.

Ésta es sólo la Ecuación de flujo óptico para cada uno de los n píxeles. El conjunto de ecuaciones se puede representar en la siguiente forma de matriz donde $Av = b$:

$$A = \begin{bmatrix} I_x(q_1) & I_y(q_1) \\ I_x(q_2) & I_y(q_2) \\ \vdots & \vdots \\ I_x(q_n) & I_y(q_n) \end{bmatrix}, \quad v = \begin{bmatrix} V_x \\ V_y \end{bmatrix}, \quad b = \begin{bmatrix} -I_t(q_1) \\ -I_t(q_2) \\ \vdots \\ -I_t(q_n) \end{bmatrix}, \quad (2.42)$$

Se debe tener en cuenta que el problema del flujo óptico, exige resolver dos variables desconocidas con una ecuación. Dichas incógnitas son (V_x y V_y) con nueve ecuaciones. En segundo lugar, para abordar el problema, se aplica el ajuste por mínimos cuadrados para obtener el siguiente problema de dos ecuaciones, dos incógnitas:

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum_i I_x(q_i)^2 & \sum_i I_x(q_i)I_y(q_i) \\ \sum_i I_y(q_i)I_x(q_i) & \sum_i I_y(q_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i I_x(q_i)I_t(q_i) \\ -\sum_i I_y(q_i)I_t(q_i) \end{bmatrix}, \quad (2.43)$$

En pocas palabras, se identifican algunas características interesantes para rastrear y calcular iterativamente los vectores de flujo óptico de estos puntos. Sin embargo, la adopción del método Lucas Kanade sólo funciona para movimientos pequeños y falla cuando hay un gran movimiento. Por tanto, la implementación del método Lucas Kanade adopta la siguiente forma piramidal.

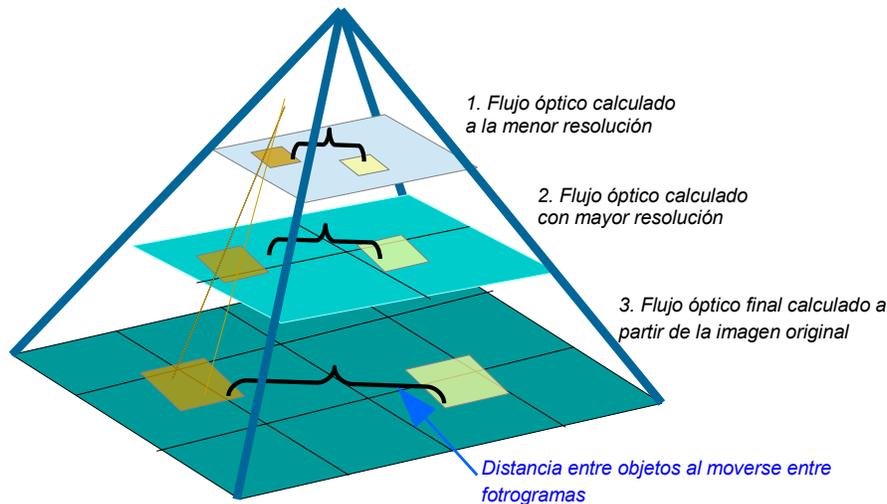


Figura 2.6: Pirámide de flujo óptico[23].

2.5. Filtro de Kalman

Debido a que los modelos matemáticos son sólo una aproximación del comportamiento de los sistemas dinámicos, y a que las señales medidas son afectadas por ruido, existe incertidumbre en las medidas y a pesar de usar modelos determinísticos, éstos se ven afectados por procesos estocásticos, el filtro de Kalman permite estimar los estados de un sistema estocástico. Puede definirse como un algoritmo de procesamiento de datos óptimo recursivo. Óptimo en el sentido de que minimiza un criterio determinado y porque incorpora toda la información que se le suministra para determinar el filtrado. Recursivo porque no precisa mantener los datos previos, los que facilita su implementación en sistemas de procesamiento en tiempo real. Este filtro ha sido diseñado para ser implementado en sistemas discretos. El objetivo del filtro de Kalman es estimar los estados de una forma óptima, de manera que se minimiza un índice de error cuadrático medio.

Un sistema lineal discretizado empleando un tiempo de muestreo ΔT puede ser representado como:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + v(k), \\ y(k) &= Cx(k) + w(k), \end{aligned} \tag{2.44}$$

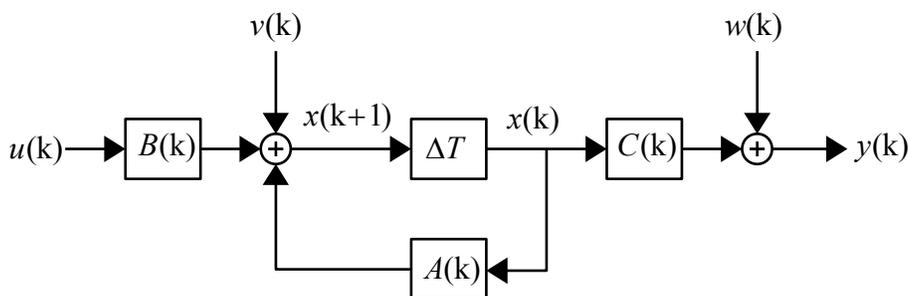


Figura 2.7: Esquema de sistema discretizado [26].

Siendo:

El estado del sistema $x(k)$ pertenece a \mathbb{R}^n , la salida $y(k)$ pertenece a \mathbb{R}^m , el proceso estocástico asociado a la medida $w(k)$ pertenece a \mathbb{R}^p , y el proceso estocástico asociado

al sistema $v(k)$ pertenece a \mathbb{R}^q . Las matrices A , B , y C son matrices determinísticas de dimensiones $n \times n$, $n \times m$, y $p \times n$, respectivamente.

Para implementar este filtro, es necesario suponer las siguientes consideraciones:

$$\begin{aligned}
E[x(0)] &= x_0, \quad \text{donde } x_0 \in \mathbb{R}^n, \\
E[w(k)] &= 0 \quad \forall k, \quad \text{donde } w(k) \in \mathbb{R}^p, \\
E[v(k)] &= 0 \quad \forall k, \quad \text{donde } v(k) \in \mathbb{R}^q, \\
E[x(0), w(k)] &= 0 \quad \forall k, \quad \text{donde } x(0) \in \mathbb{R}^n \text{ y } w(k) \in \mathbb{R}^p, \\
E[x(0), v(k)] &= 0 \quad \forall k, \quad \text{donde } x(0) \in \mathbb{R}^n \text{ y } v(k) \in \mathbb{R}^q, \\
E[v(k), w(j)] &= 0 \quad \forall k, \quad \text{donde } v(k), w(j) \in \mathbb{R}^p, \\
E[w(k), w(k)] &= R(k), \quad \text{donde } R(k) \text{ es una matriz de } \mathbb{R}^{p \times p}, \\
E[w(k), w(j)] &= 0 \quad \forall k \neq j, \quad \text{donde } w(k), w(j) \in \mathbb{R}^p, \\
E[v(k), v(k)] &= Q(k), \quad \text{donde } Q(k) \text{ es una matriz de } \mathbb{R}^{q \times q}, \\
E[v(k), v(j)] &= 0 \quad \forall k \neq j, \quad \text{donde } v(k), v(j) \in \mathbb{R}^q, \\
E[x(0), x(k)] &= P_0 \quad \forall k, \quad \text{donde } x(0), x(k) \in \mathbb{R}^n \text{ y } P_0 \text{ es una matriz de } \mathbb{R}^{n \times n}.
\end{aligned}$$

donde $E[\cdot]$ es la esperanza matemática, Q , R y P son las matrices de covarianza las cuales son diagonales y por tanto simétricas. Sea $\hat{x}(k)$ la estimación del estado $x(k)$ en el instante k . Definiendo el error $e(k)$ como la diferencia entre el valor real del estado y la estimación se tiene:

$$e(k) = x(k) - \hat{x}(k), \quad (2.45)$$

El objetivo es minimizar:

$$P(n) = E \{ e(n)e(n)^T \}, \quad (2.46)$$

$P(n)$ es la matriz de covarianza del error. Dependiendo del valor que tome n en la ecuación (2.46), se tienen distintas representaciones del filtro de Kalman. Sí $n = k + 1$, el filtrado es de predicción, si $n = k$ el filtrado es de alisado.

El objetivo del filtro de Kalman consiste en determinar los valores de $\hat{x}(k)$ a partir de las señales contaminadas $y(0)$, $y(1)$, ..., $y(k)$ para que la matriz de covarianza del error

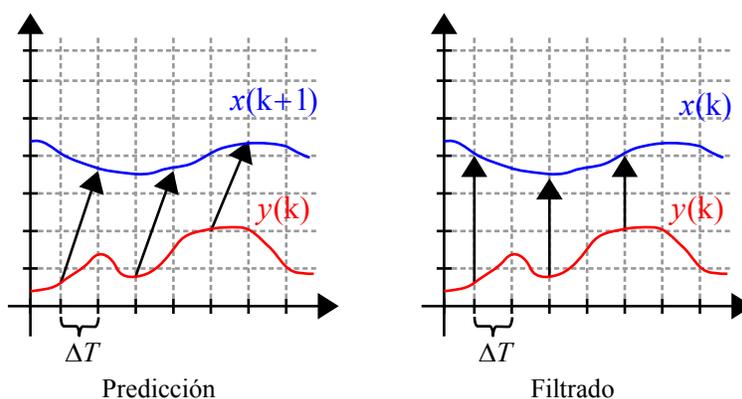


Figura 2.8: Representaciones del filtro de Kalman[26].

$P(k)$ sea mínima, o dicho de otra forma, determinar los valores de $\hat{x}(k+1)$ a partir de las medidas afectadas por el ruido de la observación $y(k+1)$ para que la matriz $P(k+1)$ sea mínima.

El filtrado de $\hat{x}(k+1)$, se realiza a partir del estado anterior y contemplando un factor de corrección que está en función del error. El algoritmo se compone de dos pasos los cuales son ejecutados de forma iterativa. Predicción antes de tener la medida $y(k+1)$ y corrección o actualización del estado. En la siguiente figura se puede ver el proceso de cálculo.

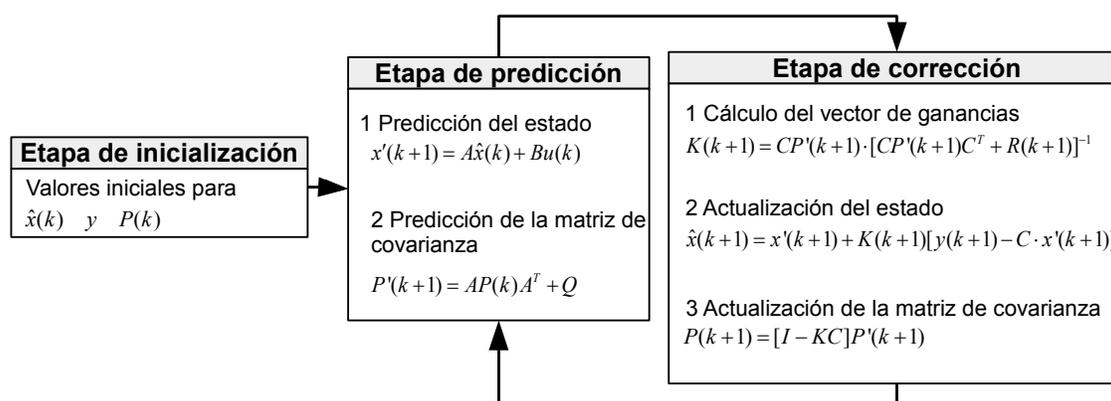


Figura 2.9: Algoritmo del filtro de Kalman [26].

Paso 1 Predicción-Extrapolación-Estimación.

Se calcula una predicción del estado $x(k+1)$, sea $x'(k+1)$ dicha predicción, el valor de la predicción es hallado a partir del valor más actualizado del estado.

$$x'(k+1) = A\hat{x}(k) + Bu(k), \quad (2.47)$$

Se predice el valor de la matriz de covarianza del error previo a la medida $P'(k+1)$. De acuerdo a la definición de error dada en (2.46), el error está dado por:

$$e'(k+1) = x(k+1) - x'(k+1), \quad (2.48)$$

Lo anterior implica que $e'(k+1)$ en esta etapa de predicción vale:

$$\begin{aligned} e'(k+1) &= Ax(k) + Bu(k) + v(k) - A\hat{x}(k) - Bu(k), \\ &= A[x(k) - \hat{x}(k)] + v(k), \end{aligned} \quad (2.49)$$

La matriz de covarianza $P'(k+1)$ por tanto es:

$$P'(k+1) = E \left\{ [A(x(k) - \hat{x}(k)) + v(k)] [A(x(k) - \hat{x}(k)) + v(k)]^T \right\}, \quad (2.50)$$

implica:

$$P'(k+1) = E \left\{ [A(x(k) - \hat{x}(k))] [A(x(k) - \hat{x}(k))]^T \right\} + E \left\{ v(k)v(k)^T \right\}, \quad (2.51)$$

Ya que $P(k) = E[e(k)e(k)^T]$ y que $Q = E[v(k)v(k)^T]$ la ecuación anterior puede reescribirse como:

$$P'(k+1) = AP(k)A^T + Q, \quad (2.52)$$

Paso 2 Innovación-Actualización-Corrección.

El valor del estado es calculado a partir del estado anterior y de un factor de corrección que está en función del error. Esto es:

$$\hat{x}(k+1) = x'(k+1) + K(k+1)[y(k+1) - Cx'(k+1)], \quad (2.53)$$

Donde el factor de corrección es:

$$K(k+1)[y(k+1) - Cx'(k+1)], \quad (2.54)$$

y $y(k+1)$ es el último valor observado, $x'(k+1)$ es el valor más actualizado disponible del estado, el cual, fue calculado en la fase de predicción. Se busca el valor $K(k+1)$ para hallar un valor óptimo de $\hat{x}(k+1)$ tal que la matriz de covarianza del error sea mínima. Por definición se tiene que el error es:

$$e(k+1) = x(k+1) - \hat{x}(k+1), \quad (2.55)$$

Sustituyendo el valor de $\hat{x}(k+1)$ en la ecuación(2.55) se tiene:

$$e(k+1) = x(k+1) - x'(k+1) - K[y(k+1) - Cx'(k+1)], \quad (2.56)$$

La observación $y(k+1)$ tiene el valor:

$$y(k+1) = Cx(k+1) + w(k+1)], \quad (2.57)$$

Por lo tanto:

$$e(k+1) = x(k+1) - x'(k+1) - KCx(k+1) - Kw(k+1) - KCx'(k+1), \quad (2.58)$$

Al agrupar términos se obtiene:

$$e(k+1) = [I - KC]((x(k+1) - x'(k+1)) - Kw(k+1)), \quad (2.59)$$

Ahora aplicando la definición de la ecuación (2.46).

$$P(k+1) = E \left\{ \langle [I - KC]((x(k+1) - x'(k+1)) - Kw(k+1)) \rangle \langle [I - KC]((x(k+1) - x'(k+1)) - Kw(k+1))^T \rangle \right\} \quad (2.60)$$

$$P(k+1) = E \left\{ \langle [I - KC]((x(k+1) - x'(k+1))) \rangle \langle [I - KC]((x(k+1) - x'(k+1)))^T \rangle + E \left\{ Kw(k+1) w(k+1)^T K^T \right\} \right\}, \quad (2.61)$$

$E \left\{ Kw(k+1) w(k+1)^T K^T \right\}$ es la matriz de covarianza asociada a la medida $R(k+1)$. Sustituyendo dicha matriz en la expresión anterior y factorizando se tiene:

$$P(k+1) = [I - KC]E \left\{ \langle (x(k+1) - x'(k+1)) \rangle \langle (x(k+1) - x'(k+1))^T \rangle \right\} [I - KC]^T + KR(k+1)K^T, \quad (2.62)$$

Anteriormente se definió que:

$$P'(k+1) = E \left\{ \langle (x(k+1) - x'(k+1)) \rangle \langle (x(k+1) - x'(k+1))^T \rangle \right\}, \quad (2.63)$$

Con lo que:

$$P(k+1) = [I - KC]P'(k+1)[I - KC]^T + KRK^T, \quad (2.64)$$

Haciendo cambio de variable, $P'(k+1) \triangleq P1$ y $P'(k) \triangleq P$, la ecuación (2.64) puede ser reescrita como:

$$P = P1 + KCP + PC^T K^T + KCPC^T K^T + KRK^T, \quad (2.65)$$

$$TrP = TrP1 + TrKCP + TrPC^T K^T + TrKCPC^T K^T + TrKRK^T, \quad (2.66)$$

Donde Tr es la traza de la matriz. Diferenciando respecto a K para obtener el mínimo de la expresión e igualamos a cero.

$$\frac{\partial \text{Tr}P}{\partial K} = -2\text{Tr}CP + 2\text{Tr}CPC^T + 2KR = 0, \quad (2.67)$$

$$K = CP[CP^T + R]^{-1}, \quad (2.68)$$

es decir:

$$K(k+1) = CP'(k+1) \cdot [CP'(k+1)C^T + R(k+1)]^{-1}, \quad (2.69)$$

Se puede comprobar que es el mínimo a través del Hessiano. Sustituyendo (2.69) en (2.64):

$$P(k+1) = [I - KC]P'(k+1), \quad (2.70)$$

Inicialización del algoritmo.

Para iniciar el algoritmo es necesario conocer las siguientes condiciones de contorno: Un valor inicial del estado $x(0)$. El valor de la matriz de covarianza del error, que para $P(0)$. Se puede asumir que $P(0) = Q$. Y los valores de las matrices de covarianza asociadas al sistema y a la medida Q y R [26].

2.6. Filtro de Kalman extendido

El problema de filtrado de Kalman considerado hasta este punto, ha abordado la estimación de un vector de estado en un modelo lineal de un sistema dinámico. Sin embargo, si el modelo es no lineal, se puede extender el uso del filtrado de Kalman a través de un procedimiento de linealización. El filtro resultante se conoce como filtro de Kalman extendido (EKF, por sus siglas en inglés) [37].

Representación espacio estado del sistema:

$$\mathbf{x}_{k+1} = \mathbf{F}_{k+1,k}\mathbf{x}_k + \mathbf{w}_k, \quad (2.71)$$

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k, \quad (2.72)$$

donde \mathbf{w}_k y \mathbf{v}_k son procesos de ruido gaussianos independientes de media cero y matrices de covarianza \mathbf{Q}_k y \mathbf{R}_k , respectivamente.

Filtro en adelante:

Inicialización: Para $k = 0$, establece

$$\hat{\mathbf{x}}_0 = E[\mathbf{x}_0], \quad (2.73)$$

$$\mathbf{P}_0 = E[(\mathbf{x}_0 - E[\mathbf{x}_0])(\mathbf{x}_0 - E[\mathbf{x}_0])^T], \quad (2.74)$$

Cálculo: Para $k = 1, 2, \dots$, se obtiene:

$$\hat{\mathbf{x}}_k^{f-} = \mathbf{F}_{k,k-1} \hat{\mathbf{x}}_{k-1}^{f-}, \quad (2.75)$$

$$\mathbf{P}_k^{f-} = \mathbf{F}_{k,k-1} \mathbf{P}_{k-1}^f \mathbf{F}_{k,k-1}^T + \mathbf{Q}_{k-1}, \quad (2.76)$$

$$\mathbf{G}_k^f = \mathbf{P}_k^{f-} \mathbf{H}_k^T [\mathbf{H}_k \mathbf{P}_k^{f-} \mathbf{H}_k^T + \mathbf{R}_k]^{-1}, \quad (2.77)$$

$$\hat{\mathbf{x}}_k^f = \hat{\mathbf{x}}_k^{f-} + \mathbf{G}_k^f (\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^{f-}), \quad (2.78)$$

Suavizador recursivo

Inicialización: Para $k = N$, sea

$$\mathbf{P}_N = \mathbf{P}_N^f, \quad (2.79)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^f, \quad (2.80)$$

Cálculo: Para $k = N - 1, N - 2$, se calcula

$$\mathbf{A}_k = \mathbf{P}_k^f \mathbf{F}_{k+1,k}^T [\mathbf{P}_{k+1}^{f-}]^{-1}, \quad (2.81)$$

$$\mathbf{P}_k = \mathbf{P}_k^f - \mathbf{A}_k (\mathbf{P}_{k+1}^{f-} - \mathbf{P}_{k+1}) \mathbf{A}_k^T, \quad (2.82)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^f + \mathbf{A}_k(\hat{\mathbf{x}}_{k+1} - \hat{\mathbf{x}}_{k+1}^{f-}), \quad (2.83)$$

La extensión es factible gracias a que el filtro de Kalman se describe en términos de ecuaciones de diferencias en el caso de sistemas en tiempo discreto. Para acondicionar el sistema buscando aplicar el filtro de Kalman extendido, considere el sistema dinámico no lineal descrito por el modelo de espacio de estados [37]:

$$\mathbf{x}_{k+1} = \mathbf{f}(k, \mathbf{x}_k) + \mathbf{w}_k, \quad (2.84)$$

$$\mathbf{y}_k = \mathbf{h}(k, \mathbf{x}_k) + \mathbf{v}_k, \quad (2.85)$$

Donde, como antes, \mathbf{w}_k y \mathbf{v}_k son procesos independientes de ruido Gaussiano blanco con media cero y matrices de covarianza \mathbf{R}_k y \mathbf{Q}_k , respectivamente. Aquí, sin embargo, la función $\mathbf{f}(k; \mathbf{x}_k)$ representa una función de matriz de transición no lineal posiblemente variante en el tiempo. De manera similar, la función $\mathbf{h}(k; \mathbf{x}_k)$ representa una matriz de medición no lineal que también puede ser variable en el tiempo. La idea básica del filtro de Kalman extendido es linealizar el modelo de espacio de estados de las Ecuaciones 2.84 y 2.85 en cada instante de tiempo alrededor de la estimación más reciente del estado, que se toma ya sea como $\hat{\mathbf{x}}_k$ o $\hat{\mathbf{x}}_k^-$, dependiendo de la función particular que se esté considerando. Una vez obtenido un modelo lineal, se aplican las ecuaciones estándar del filtro de Kalman. De manera más explícita, la aproximación se realiza en dos etapas:

Etapla 1: Las siguientes dos matrices se construyen:

$$\mathbf{F}_{k+1,k} = \left. \frac{\partial \mathbf{f}(k, \mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_k}, \quad (2.86)$$

$$\mathbf{H}_k = \left. \frac{\partial \mathbf{h}(k, \mathbf{x}_k)}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_k}, \quad (2.87)$$

Que son, las entradas ij -ésimas de $\mathbf{F}_{k+1,k}$ es igual a la derivada parcial del i -ésimo componente de $\mathbf{F}(k, x)$ respecto a la j -ésima componente de \mathbf{x} . Similarmente, la entrada ij -ésima de \mathbf{H}_k es igual a la derivada parcial del i -ésimo componente de $\mathbf{H}(k, \mathbf{x})$ respecto a la j -ésima componente de \mathbf{x} . En el primer caso, las derivadas se evalúan en $\hat{\mathbf{x}}_k$, mientras que en el segundo caso, las derivadas se evalúan en $\hat{\mathbf{x}}_k$. Las entradas de las matrices $\mathbf{F}_{k+1,k}$ y \mathbf{H}_k son todas conocidas (es decir, computables), ya que $\hat{\mathbf{x}}_k$ y $\hat{\mathbf{x}}_k^-$ están disponibles en el tiempo k .

Etapa 2: Una vez evaluadas las matrices $\mathbf{F}_{k+1,k}$ y \mathbf{H}_k , se emplean en una aproximación de Taylor de primer orden de las funciones no lineales $\mathbf{F}(k, \mathbf{x}_k)$ y $\mathbf{H}(k, \mathbf{x}_k)$ alrededor de $\hat{\mathbf{x}}_k$ y $\hat{\mathbf{x}}_k$, respectivamente. Específicamente, $\mathbf{F}(k, \mathbf{x}_k)$ y $\mathbf{H}(k, \mathbf{x}_k)$ se aproximan de la siguiente manera [37]:

$$\mathbf{F}(k, \mathbf{x}_k) \approx \mathbf{F}(\mathbf{x}, \hat{\mathbf{x}}_k) + \mathbf{F}_{k+1,k}(\mathbf{x}, \hat{\mathbf{x}}_k), \quad (2.88)$$

$$\mathbf{H}(k, \mathbf{x}_k) \approx \mathbf{H}(\mathbf{x}, \hat{\mathbf{x}}_k) + \mathbf{H}_{k+1,k}(\mathbf{x}, \hat{\mathbf{x}}_k), \quad (2.89)$$

Con las expresiones aproximadas anteriores en mente, ahora se puede proceder a aproximar las ecuaciones de estado no lineales 2.84 y 2.85 como se muestra a continuación:

$$\mathbf{x}_{k+1} \approx \mathbf{F}_{k+1,k}x_k + \mathbf{w}_k + \mathbf{d}_k, \quad (2.90)$$

$$\mathbf{y}_k \approx \mathbf{H}_k x_k + \mathbf{v}_k, \quad (2.91)$$

Donde se han introducido dos nuevas cantidades:

$$\mathbf{y}'_k = \mathbf{y}_k - \mathbf{h}(x, \hat{x}_k) + \mathbf{H}_k \hat{\mathbf{x}}_k, \quad (2.92)$$

$$\mathbf{d}_k = \mathbf{f}(\mathbf{x}, \hat{\mathbf{x}}_k) - \mathbf{F}_{k+1,k} \hat{\mathbf{x}}_k, \quad (2.93)$$

Las entradas en el término \mathbf{y}'_k son todas conocidas en el tiempo k , y por lo tanto, \mathbf{y}'_k puede ser considerado un vector de observación en el tiempo k . Del mismo modo, las entradas en el término \mathbf{d}_k son todas conocidas en el tiempo k .

Tabla 2.2: Filtro de Kalman Extendido [37].

Etapa	Descripción
Modelo	$\mathbf{x}_{k+1} = \mathbf{f}(k, \mathbf{x}_k) + \mathbf{w}_k$ $\mathbf{y}_k = \mathbf{h}(k, \mathbf{x}_k) + \mathbf{v}_k$ <p>donde \mathbf{w}_k y \mathbf{v}_k son procesos de ruido independientes, de media cero y distribución Gaussiana, con matrices de covarianza \mathbf{Q}_k y \mathbf{R}_k, respectivamente.</p>
Definiciones	$\mathbf{F}_{k+1,k} = \left. \frac{\partial \mathbf{f}(k, \mathbf{x})}{\partial \mathbf{x}} \right _{\mathbf{x}=\mathbf{x}_k}$ $\mathbf{H}_k = \left. \frac{\partial \mathbf{h}(k, \mathbf{x})}{\partial \mathbf{x}} \right _{\mathbf{x}=\mathbf{x}_k}$
Inicialización	$\hat{\mathbf{x}}_0 = E[\mathbf{x}_0]$ $\mathbf{P}_0 = E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T]$
Cálculo	<p>Para $k = 1, 2, \dots$, calcular:</p> <p>Propagación de la estimación del estado</p> $\hat{\mathbf{x}}_k = \mathbf{f}(k, \hat{\mathbf{x}}_{k-1})$ <p>Propagación de la covarianza del error</p> $\mathbf{P}_k = \mathbf{F}_{k,k-1} \mathbf{P}_{k-1} \mathbf{F}_{k,k-1}^T + \mathbf{Q}_{k-1}$ <p>Matriz de ganancia de Kalman</p> $\mathbf{G}_k = \mathbf{P}_k \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k \mathbf{H}_k^T + \mathbf{R}_k)^{-1}$ <p>Actualización de la estimación del estado</p> $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k + \mathbf{G}_k (\mathbf{y}_k - \mathbf{h}(k, \hat{\mathbf{x}}_k))$ <p>Actualización de la covarianza del error</p> $\mathbf{P}_k = (\mathbf{I} - \mathbf{G}_k \mathbf{H}_k) \mathbf{P}_k$

La Tabla 2.2 resume las recurrencias involucradas en el cálculo del filtro de Kalman extendido.

2.7. Método del gradiente descendente

Considerando el siguiente sistema no lineal causal y discretizado:

$$x(i+1) = f(x(i), u(i), \alpha(i)), \quad (2.94)$$

donde el control depende del estado y los parámetros del controlador teniendo:

$$u_{(i)} = (x(i), \alpha(i)),$$

definiendo los vectores, para el estado, el control y los parámetros del controlador, los siguientes:

$$x(i) = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, u(i) = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}, \alpha(i) = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_p \end{bmatrix}, \quad (2.95)$$

El índice de desempeño J :

$$J = \phi(x(N), \alpha) + \sum_{i=0}^{N-1} L(x(i), u(i), \alpha), \quad (2.96)$$

$$J = x(N)^T H x(N) + \sum_{i=0}^{N-1} \left\{ x(i)^T Q x(i) + u(i)^T R u(i) \right\}, \quad (2.97)$$

El índice de rendimiento J mide el desempeño del sistema de control. Se compone de dos términos: el primero es el término terminal, que representa la importancia de alcanzar un estado final deseado, y el segundo es la suma de los términos cuadráticos de los estados del sistema x y las entradas de control u , ponderados por las matrices Q y R , respectivamente.

H , Q y R son matrices de ponderación:

$$\begin{aligned} H, Q &\geq 0, \\ R &> 0, \end{aligned} \quad (2.98)$$

Las matrices de ponderación H y Q deben ser semidefinidas positivas, y la matriz R debe ser definida positiva. Estas condiciones aseguran que el índice de rendimiento J sea

positivo y que el problema de control óptimo tenga una solución única [32].

La función de costo V es:

$$V(x(k), \alpha, u(k, N-1)) = \varphi(x(N), \alpha) + L(x(k), u(k), \alpha) + \sum_{i=k+1}^{N-1} L(x(i), u(i), \alpha), \quad (2.99)$$

$$V(x(k+1)) = \varphi(x(N), \alpha) + \sum_{i=k+1}^{N-1} L(x(i), u(i), \alpha), \quad (2.100)$$

$$V(x(k), \alpha, u(k, N-1), k) = V(x(k+1), \alpha, u[k+1, N-1], k+1) + L(x(k), u(k), \alpha), \quad (2.101)$$

La función de costo V representa el costo acumulado desde el tiempo k hasta el final del horizonte de control N . Está compuesta por el término terminal $\varphi(x(N), \alpha)$ y la suma de los costos instantáneos $L(x(k), u(k), \alpha)$, en cada paso de tiempo.

La relación entre la función de costo y el índice de desempeño está dada por:

$$V(x(k), \alpha, u[k, N-1]) = \underbrace{x(N)^T H x(N)}_{\varphi} + \underbrace{\sum_{i=k+1}^{N-1} \{x(i)^T Q x(i) + u(i)^T R u(i)\}}_L + \underbrace{x(k)^T Q x(k) + u(k)^T R u(k)}_{L(k)}, \quad (2.102)$$

La ecuación 2.102, muestra como la función de costo V se relaciona con el índice de desempeño J y se compone de tres partes el término terminal, la suma de los costos instantáneos desde el tiempo $k+1$ hasta $N-1$ y el costo instantáneo en el instante k . La idea detrás de este enfoque es minimizar la función de costo V para encontrar el control óptimo u y el estado del sistema x . Este es el fundamento del control óptimo y puede resolverse utilizando técnicas como la programación dinámica, el principio del mínimo de "principio del mínimo de Pontryagin", que es una técnica importante en la teoría del control óptimo.

Este es el fundamento del control óptimo y puede resolverse utilizando técnicas como la programación dinámica, el principio del mínimo de Pontryagin o la programación cuadrática. Estas técnicas buscan encontrar el control óptimo u y el estado del sistema x que minimizan el índice de desempeño J sujeto a las restricciones del sistema (como las

ecuaciones de estado y las restricciones de entrada de control).

El principio del mínimo de Pontryagin es especialmente útil cuando se trata de sistemas de control continuos y proporciona una herramienta poderosa para encontrar soluciones analíticas y numéricas a problemas de control óptimo. Este principio se basa en la idea de que, en un sistema de control óptimo, el Hamiltoniano mínimo (una función que combina el Lagrangiano y las ecuaciones de estado) debe ser mínimo con respecto a las entradas de control óptimas.

En resumen, minimizar el índice de desempeño J es el objetivo principal en la teoría del control óptimo, y diferentes técnicas, como la programación dinámica, el principio del mínimo de Pontryagin y la programación cuadrática, pueden utilizarse para encontrar la solución óptima.

$$V(x(0), \alpha^{j+1}, u^{j+1}[0, N-1]) = V(x(0), \alpha^j, u^j[0, N-1]) + \frac{\partial}{\partial \alpha}(V(x(0), \alpha^j, u^j[0, N-1]))\delta\alpha + \dots + \frac{\partial}{\partial u[0, N-1]}(V(x(0), \alpha^j, u^j[0, N-1]))\delta u[0, N-1], \quad (2.103)$$

Estas ecuaciones muestran una relación iterativa para actualizar los parámetros de control u y los parámetros de diseño α en un proceso de optimización. La idea es mejorar iterativamente el índice de desempeño J ajustando α y u en cada paso, utilizando la información del gradiente de la función de costo V . A continuación se muestran los pasos para la optimización interactiva:

Actualización de α :

$$\delta\alpha = \alpha^{j+1} - \alpha^j, \quad (2.104)$$

$$\delta\alpha = -\bar{\varepsilon} \left[\frac{\partial}{\partial \alpha}(V(x(0), \alpha^j, u^j[0, N-1])) \right]^T, \quad (2.105)$$

En cada iteración, se actualiza α utilizando el gradiente negativo de la función de costo V con respecto a α , multiplicado por un factor de aprendizaje $\bar{\varepsilon}$.

Actualización de u :

$$\delta u[0, N-1] = -\varepsilon \left[\frac{\partial}{\partial u[0, N-1]}(V(x(0), \alpha^j, u^j[0, N-1])) \right], \quad (2.106)$$

De manera similar, en cada iteración, se actualiza u utilizando el gradiente negativo de la función de costo V con respecto a u , multiplicado por un factor de aprendizaje ε .

Condiciones de mejora:

$$V(x(0), \alpha^{j+1}, u^{j+1}[0, N-1]) \leq V(x(0), \alpha^j, u^j[0, N-1]), \quad (2.107)$$

Esta condición asegura que, después de cada actualización, la función de costo V no aumenta, es decir, el índice de desempeño J mejora o se mantiene igual en cada iteración.

Función de costo en el tiempo k :

$$V(x(k), k) = x(N)^T H x(N) + \sum_{i=k}^{N-1} \left\{ x(i)^T Q x(i) + u(i)^T R u(i) \right\}, \quad (2.108)$$

Esta ecuación representa la función de costo V en un tiempo específico k , que está relacionada con el índice de desempeño J y las matrices de ponderación Q y R que penalizan el estado del sistema x y las entradas de control u , respectivamente. Además, la matriz H penaliza el estado final del sistema.

Con base en las ecuaciones de actualización y las condiciones de mejora, el objetivo del proceso iterativo es encontrar las secuencias óptimas de parámetros de control u y diseño α que minimizan el índice de desempeño J y, por lo tanto, mejoran el comportamiento del sistema de control en general[32].

El método del gradiente descendente se puede resumir en los siguientes pasos:

1. Inicializar α^j , $u^j[0, N-1]$ y $x[0, N]$.
2. Calcular J^j usando la ecuación: $J = \varphi(x(N), \alpha) + \sum_{i=0}^{N-1} L(x(i), \alpha, u(i))$.
3. Calcular las derivadas parciales $V_x^j(k)$, $V_\alpha^j(k)$, $F_\alpha^j(k)$ y $L_\alpha^j(k)$, siguiendo las ecuaciones proporcionadas.
4. Actualizar $u^{j+1}(k)$ y α^{j+1} usando las ecuaciones: $u^{j+1}(k) = u^j(k) - \varepsilon \left[\frac{\partial V(k)}{\partial u^j(k)} \right]^T$ y $\alpha^{j+1} = \alpha^j - \bar{\varepsilon} V_\alpha^T(0)$.
5. Calcular $x(k+1) = F(x(k), \alpha^{j+1}, u^{j+1}(k))$ y actualizar $x^{j+1}[0, u]$.

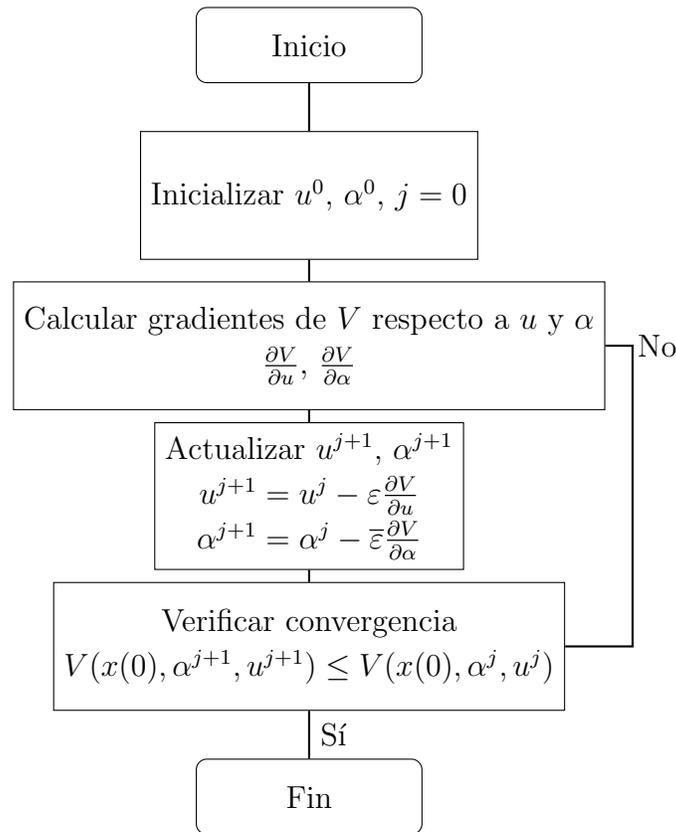


Figura 2.10: Diagrama de flujo del proceso iterativo para el método de optimización [32].

6. Calcular J^{j+1} y comparar con J^j . Si $J^{j+1} < J^j$, continuar con el siguiente paso. De lo contrario, reducir ε y $\bar{\varepsilon}$ a la mitad y volver al paso 4.
7. Comprobar si $J^{j+1} - J^j > \beta$, donde $\beta > 0$. Si se cumple, detener el proceso. De lo contrario, incrementar j y repetir desde el paso 2 [32].

2.8. Conclusión capítulo 2

En este capítulo, se ha desarrollado un marco teórico integral que sienta las bases para una de las principales contribuciones de esta tesis: la creación de un Controlador Proporcional-Integral-Derivativo (PID) auto-sintonizable en línea y subóptimo. Este enfoque innovador aprovecha las ventajas del controlador PID, el Filtro de Kalman y el método del gradiente descendente para optimizar el algoritmo de control en tiempo real.

El capítulo comienza con una explicación detallada del método de Denavit-Hartenberg, crucial para la selección de sistemas de referencia y el algoritmo Denavit-Hartenberg, proporcionando una base sólida para el entendimiento de la estructura y la cinemática de los sistemas robóticos.

El Controlador PID, abordado posteriormente, es un pilar fundamental en el control de procesos y sistemas. La inclusión de este tema en el marco teórico subraya su relevancia para el desarrollo del controlador auto-sintonizable.

Se introduce el método de Lucas-Kanade, centrado en el flujo óptico, aportando una perspectiva importante sobre el análisis de movimiento y seguimiento en tiempo real, esencial para la adaptación y optimización de los sistemas de control.

El Filtro de Kalman y su extensión, el Filtro de Kalman extendido, son discutidos en detalle. Estos filtros son fundamentales para la estimación y corrección en tiempo real de estados en sistemas dinámicos, lo cual es crucial para el desarrollo del controlador PID auto-sintonizable.

Finalmente, el método del gradiente descendente se explora como una técnica de optimización eficaz, vital para ajustar en tiempo real los parámetros del controlador y mejorar su rendimiento.

En resumen, este capítulo no solo proporciona una revisión exhaustiva de conceptos y técnicas clave en el campo del control y la robótica, sino que también establece las bases teóricas para la innovadora propuesta de esta tesis: un controlador PID auto-sintonizable en línea y subóptimo, representando un avance significativo en la optimización de algoritmos de control en tiempo real.

Capítulo 3

Plataforma robótica móvil heterogénea

En este capítulo se aborda una descripción detallada de la plataforma robótica móvil heterogénea propuesta, y los componentes principales que la integran, que son: el robot aéreo, el robot todoterreno, el sistema de monitoreo de calidad de aire, el sistema de toma de muestras de suelo y la interfaz que forma parte del Sistema de Supervisión, Control y Adquisición de Datos (SCADA).

3.1. Descripción del sistema propuesto

En el presente proyecto se propone la creación de un sistema de monitoreo de calidad del aire, y recolección de muestras de suelo. Para realizar la caracterización de la calidad del aire se propone emplear un Vehículo Aéreo no Tripulado que permita realizar lecturas a diferentes alturas, de forma que se genere un modelo no basado únicamente en una lectura estática, lo que da la posibilidad de crear modelos a partir de múltiples lecturas permitiendo aumentar la probabilidad de observar el comportamiento de la calidad del aire a lo largo de una trayectoria desarrollada por el robot. Las muestras de suelo podrán ser tomadas por un robot basado en ruedas con una arquitectura que le permita trabajar en diferentes terrenos, su alcance es de 20 m alrededor de una estación terrena, está última cuenta con una PC, en la cual ejecuta un programa que permite al usuario interactuar con los robots y sus sensores de éstos.

El sistema de monitoreo, y toma de muestras para detección de contaminantes aéreos

y terrestres, es a continuación mostrado y explicado de forma general. Dicho sistema se compone principalmente de los siguientes subsistemas:

Computadora con un sistema SCADA (S1): Es considerada como la estación terrena, permite configurar, coordinar y monitorear los robots, visualizar las variables de interés para observar el desempeño de éstos. Dichos robots son: un cuadricóptero (S3), y un carro todoterreno (S4), la interfaz ayuda al usuario a ver las posiciones y orientaciones de los robots y la lectura de los sensores embebidos en cada uno de ellos, dichos sensores son explicados a detalle más adelante.

Tarjetas de comunicación (S2): Se compone principalmente de una computadora de bajo costo, que mediante una aplicación permite monitorear vía inalámbrica los datos generados por los robots que forman parte del grupo de exploración, la información es enviada a la PC principal (S1) mediante protocolo RF para mostrar los datos colectados en el sistema SCADA. Abordo de cada uno de los robots, está una tarjeta de comunicación, por lo que cabe mencionar que cada uno de ellos se comunica directamente a la computadora principal, además, pueden comunicarse entre sí, permitiendo generar trabajo colaborativo entre cada uno de los robots pertenecientes a la plataforma heterogénea.

Cuadricóptero (S3): Es un vehículo aéreo no tripulado que permite hacer lecturas de la calidad del aire a diferentes alturas de vuelo, cuenta con un sistema de visión artificial que sirve para realizar tareas como detección de cambio de color en zonas sobre el suelo, que presente algún tono inherente a posibles contaminantes, mediante comunicación inalámbrica es posible enviar coordenadas al robot terrestre (S4) para realizar tomas de muestras por parte de este último, permitiendo agilizar y optimizar el proceso de exploración, y cálculo de trayectorias para la toma de muestras.

Robot Terrestre (S4): A través de un sistema de perforación de suelo y succión de partículas, que tiene incorporado, este robot es capaz de realizar recolección de muestras de suelo, lo anterior en lugares que impliquen un alto riesgo, o difícil acceso, mediante un sistema de ruedas el robot puede moverse en terrenos de difícil acceso, al igual que el UAV, éste puede interactuar, intercambiando información y comandos con el sistema principal (S1) y el drone (S3).

Sistema de radiocontrol manual (S5): Mediante el uso de 2 radios RC se puede operar manualmente a cada uno de los robots (cuadricóptero y robot terrestre). Con esto se busca tener un sistema de seguridad por si alguno de los robots deja de operar automáticamente con la finalidad de proteger la integridad del usuario y del equipo, además por medio del radiocontrol se puede calibrar y ajustar el comportamiento de cada robot. En la Figura 3.1 es mostrado un esquema general del sistema propuesto.

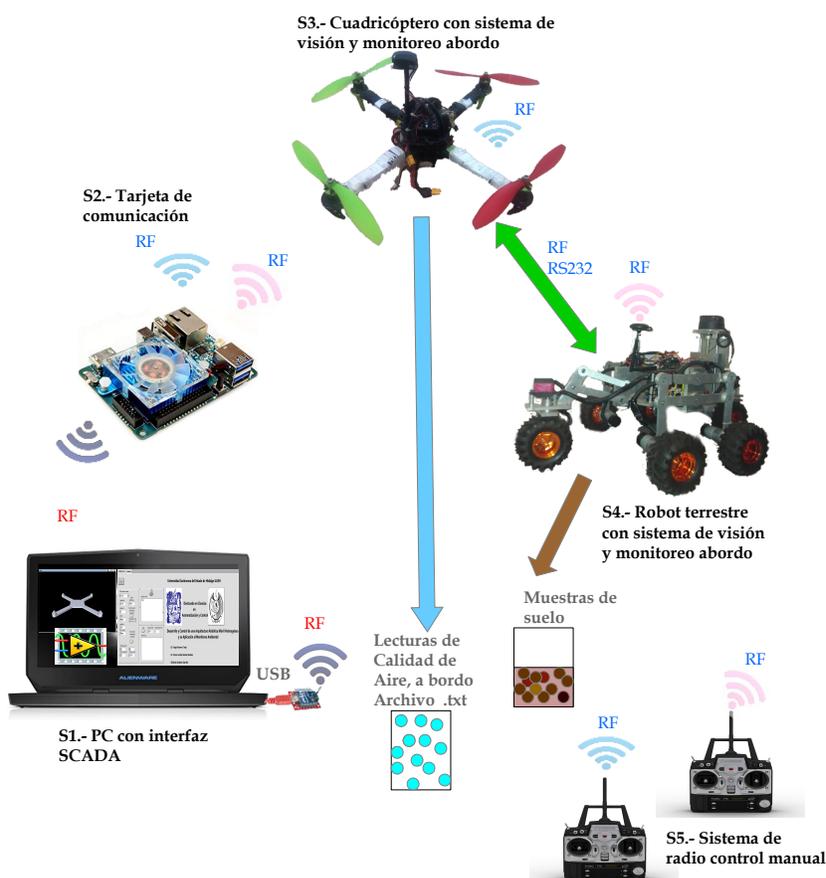


Figura 3.1: Sistema de robótica móvil heterogénea y su aplicación al monitoreo ambiental.

La interacción entre los dos robots que integran el sistema consiste, en que el UAV realiza un vuelo, siguiendo una trayectoria configurada por el usuario y que contempla, la zona de interés en la que se harán las tomas de muestras de suelo, así como el monitoreo

de la calidad de aire, realizado por el UAV, lo que implica que en dicha trayectoria se debe considerar una área de tierra, en donde se encuentre la zona a estudiar, con las restricciones planteadas en las limitaciones del presente trabajo.

Mediante una cámara ESP32 con conexión WI-FI, embarcada en el cuadricóptero, el usuario determina si en alguna zona es altamente probable que exista la presencia de contaminantes y mediante la interfaz creada en LABview® envía al dron un comando, que permite que este robot mande a su vez, al otro robot (ULV) coordenadas GPS, usando comunicación inalámbrica por protocolo RS232, a través de los modems X-Bee®, así, si durante el vuelo del UAV se determina que en determinada posición del plano de trabajo es importante tomar una muestra de suelo, se envía al ULV dicha coordenada para que dicho robot, siga una trayectoria que permita llegar a la zona donde se tomará dicha muestra, ver la Figura(3.2).

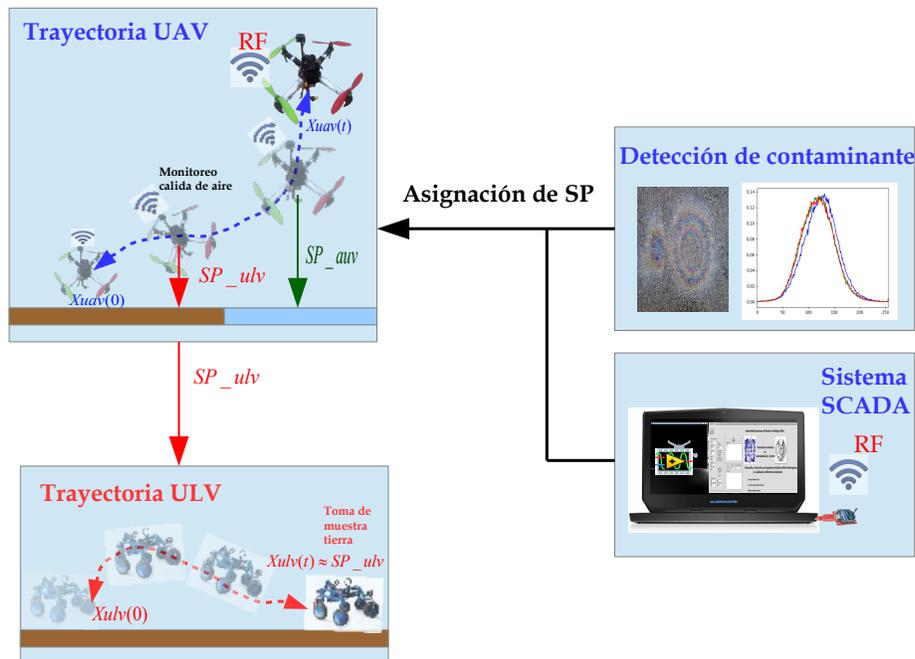


Figura 3.2: Interacción entre el sistema SCADA, el UAV y el ULV.

A continuación se hace una descripción del sistema de monitoreo a emplear en el cuadricóptero para calidad del aire, y el sistema de recolección de muestras de suelo, empleado

en el robot terrestre, además, del sistema de visión artificial a emplear en cada robot.

3.1.1. Sistema de monitoreo de calidad de aire y visión

Para el monitoreo de calidad de aire se integra en el robot aéreo un sensor MQ135, el cual permite hacer lecturas de dicha variable, éste genera una señal analógica con un rango de trabajo de 0 a 5 V de CD, dicha señal es leída por la tarjeta PIX-HAWK®, a través de su entrada analógica, a su vez el UAV envía la información a la interfaz del sistema SCADA, por protocolo RS232 inalámbrico mediante un módem X-Bee® S1 Pro, permitiendo crear un archivo con información de las lecturas para una posterior interpretación.

El sensor MQ135 (Figura B.13) permite ajustar su salida mediante un potenciómetro de sensibilidad, incorporado en la PCB de éste, en la Figura (B.12) se puede ver las curvas de comportamiento y los gases que puede detectar, para el presente trabajo se hace especial énfasis en la detección de dióxido de carbono CO_2 .

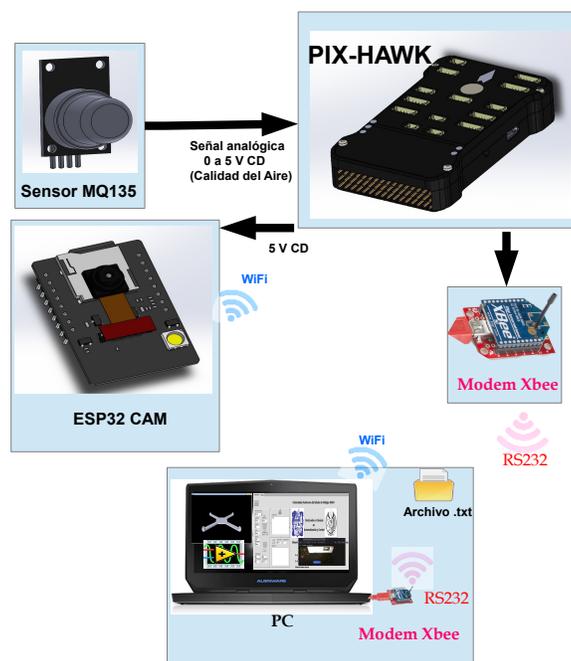


Figura 3.3: Sistema de monitoreo de calidad de aire y de visión del UAV.

Para enviar imágenes y vídeo del UAV a la PC se usa una cámara ESP32® abordo del drone, dicha cámara tiene características como su bajo peso (25 gramos), tamaño reducido (27*40.5*6 mm), consumo relativamente bajo (250 mA) y conexión por protocolo WI-FI y Bluetooth (Wifi 802.11b/g/n, Bluetooth 4.2), que la hacen adecuada para este fin, la cámara es alimentada por el PIX-HAWK®, y ésta envía directamente a la interfaz SCADA vídeo e imágenes. El sistema de monitoreo de calidad de aire y de visión del UAV son mostrados en el diagrama mostrado en la Figura(3.3).

3.1.2. Sistema de recolección de muestras de suelo

El sistema de recolección de muestras de suelo, se basa en un perforador y una aspiradora de pequeña escala, para tomar partículas sólidas de algún punto de interés. Cuenta con un motor de corriente continua, acoplado a un encoder incremental, con lo que se puede tener un control en lazo cerrado, con una resolución de 0.1 mm, en el avance o retroceso del perforador, dicho motor está dispuesto en la parte superior y acoplado a un tornillo sin fin con cuerda acme, que desplaza un carro verticalmente, en dicho carro es sujetado un segundo motor de CD, que tiene acoplada una broca en su armadura, permitiendo realizar operación de perforación, con un taladro de pequeña escala y bajo consumo de corriente.

La herramienta abrasiva a usar para perforar, tiene una medida de 1/4" de diámetro y puede ser ajustada en altura, a través del mecanismo de traslado del carro, el principio de funcionamiento de dicho mecanismo es mediante baleros lineales que corren en dos guías aceradas de 8 mm, mediante una tuerca con cuerda acme, acoplada al carro se logra hacer el desplazamiento de éste, lo anterior, porque dicha tuerca es encastrada al carro mediante un sujetador; dicha tuerca y sujetador están dispuestos en la parte posterior y al centro del carro de traslado vertical. El motor encargado de generar el movimiento rotacional que es convertido a traslación del carro, por el mecanismo propuesto, es acoplado al tornillo sin fin mediante una banda dentada con paso estándar, que permite, mediante dos poleas dentadas, que el movimiento sea transmitido del motor al carro.

Las dos barras aceradas están sujetas por soportes de aluminio encastrados a la base del mecanismo, la base al igual que el carro de traslación son propuestos de material de aluminio para aligerar el peso, la tornillería a emplear, es tornillo de 1/8 ", con una longitud de 3/4"de largo, cuerda estándar con tuerca de seguridad. La base da soporte

a todo el sistema, y ésta es fijada al robot, mediante tornillería con las características antes mencionadas. La aspiradora propuesta es de dimensiones reducidas y comercial, ésta permite succionar las partículas de tierra y polvo generadas al taladrar el suelo, lo anterior mediante un tubo de bronce, ubicado paralelamente a la broca, y dispuesto a su lado derecho respecto a la vista de frente, este tubo es conectado a la aspiradora mediante una manguera flexible permitiendo su desplazamiento sin afectar al mecanismo general, las muestras de tierra son almacenadas en un contenedor que posteriormente puede ser removido del robot para realizar las pruebas pertinentes ver Figura 3.5 . La arquitectura propuesta para el robot terrestre es de un shrimp rover, esto debido a la versatilidad y adaptabilidad que presenta para terrenos irregulares, en la imagen mostrada en la Figura 3.4, es apreciado el robot propuesto más el sistema de recolección diseñado para tomar muestras de suelo mediante succión de partículas, dispersadas a partir de taladrar.

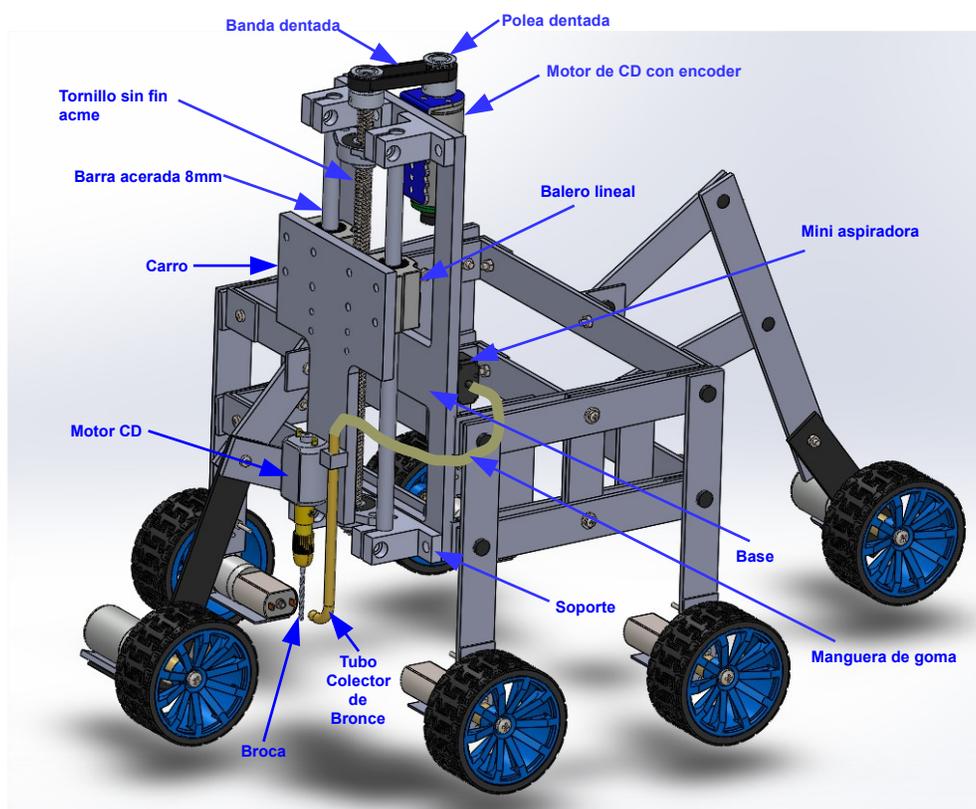


Figura 3.4: Sistema de recolección de suelo propuesto.

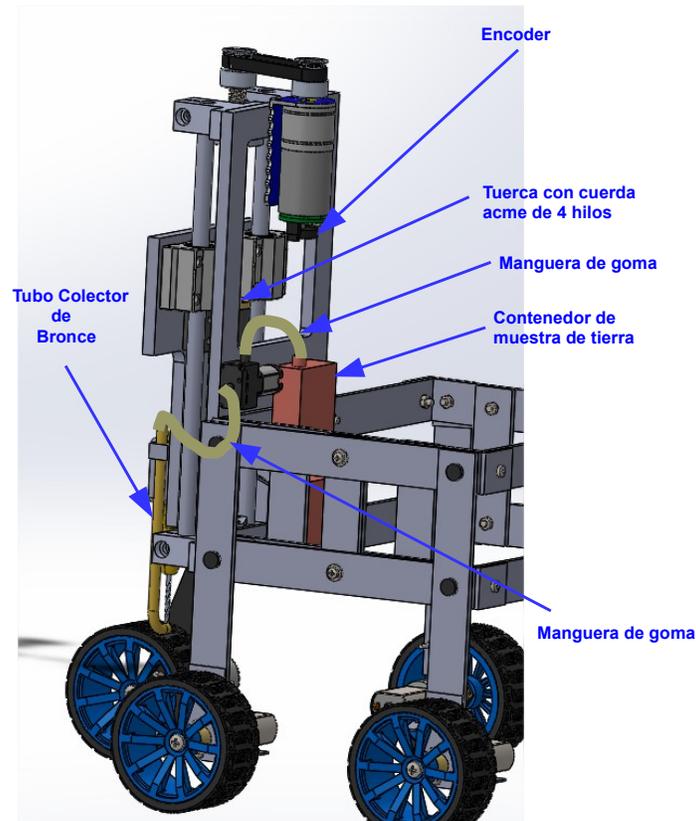


Figura 3.5: Sistema de recolección de suelo propuesto vista lateral izquierda posterior.

En el siguiente apartado son mostrados los componentes de las plataformas robóticas experimentales, que forman parte del sistema propuesto, primero se aborda una descripción detallada del vehículo aéreo no tripulado (UAV), seguida de una narrativa análoga para el vehículo terrestre no tripulado (ULV), finalmente se describe la interfaz gráfica creada en LABview® para el sistema SCADA.

3.2. Plataforma experimental UAV

La plataforma experimental diseñada para el vehículo aéreo no tripulado está basada en una arquitectura en configuración cuadricóptero, ésta es creada empleando un frame comercial, específicamente el del drone SK450, usa como controlador de vuelo un PIX-HAWK PX4, y un radio de control Futaba®, para su manejo manual a distancia.



Figura 3.6: Plataforma experimental UAV.

Embarcado en el UAV se encuentra un módulo Xbee S1 que permite realizar comunicación serial inalámbrica entre el robot aéreo mediante el PIX-HAWK® y una computadora. La PC cuenta con un programa desarrollado en LABview® que permite configurar e interactuar con el drone para obtener información de éste o manipular parámetros de su firmware. Los componentes principales que forman parte del sistema en conjunto son la PC, el UAV y el radio control, todos ellos son mostrados en la Figura 3.6.

El UAV posee una configuración de mini-helicóptero de 4 rotores, mostrada en la Figura 3.7, el controlador de vuelo usado en la plataforma es un PIX-HAWK PX4 de 3D robotics®, que incluye los accesorios mostrados en la Figura B.9 B.1, se usa un Xbee S1 para hacer transmisión y recepción de datos mediante protocolo RS232 a 115200 bps, la conexión entre dicho módem y el controlador de vuelo es mostrada en la Figura B.1.

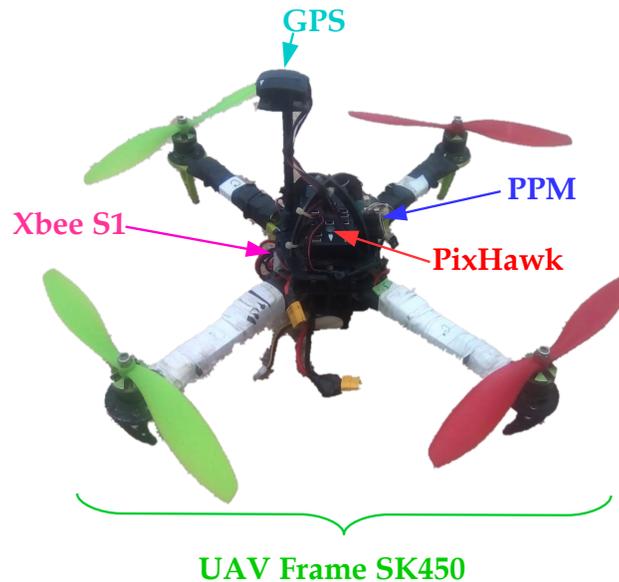


Figura 3.7: Cuadricóptero basado en el drone SK450.

El diseño y conceptualización del armado, de la estructura mecánica y los componentes del drone, son mostrados en los planos [B.15](#), [B.16](#), además en ellos pude apreciarse las medidas máximas del robot, teniendo aproximadamente 52 cm de largo y ancho, y una altura de 19.5 cm, el peso del sistema incluida la batería la cual es de 2.2 A es de 0.9 Kg. A pesar de que el sistema está basado en una plataforma comercial estos diseños son clave para la mejora de éste, ya que por ejemplo pude ser diseñada una carcasa de protección para el robot, a partir de ellos, lo que brinda la posibilidad de modificar el diseño mecánico y aumentar seguridad del sistema.

Las variables que forman parte de la interacción de los elementos de la plataforma experimental, son mostradas en la Figura [3.8](#). La PC contiene una interfaz gráfica creada en LABview® la cual interactúa a su vez con Unity®, para manipular un objeto 3D que permita observar el comportamiento del UAV en el espacio, como entradas la PC recibe a través de protocolo RS232 Serial mediante un modem Xbee S1, conectado por USB, una cadena de caracteres que contiene el estado del sistema así como las señales de control de fuerza y torques. Dichas señales se emplean para generar un archivo de texto que permite guardar las trayectorias y controles del sistema, además mediante el estado se genera la

manipulación de un objeto 3D que imita al drone, dicho objeto está creado con formato .BXF en Blender[®] y es exportado a partir de un archivo generado en Solid Works[®] con formato .WRL, en el plano B.14 es mostrado el diseño realizado para dicho cuerpo 3D.

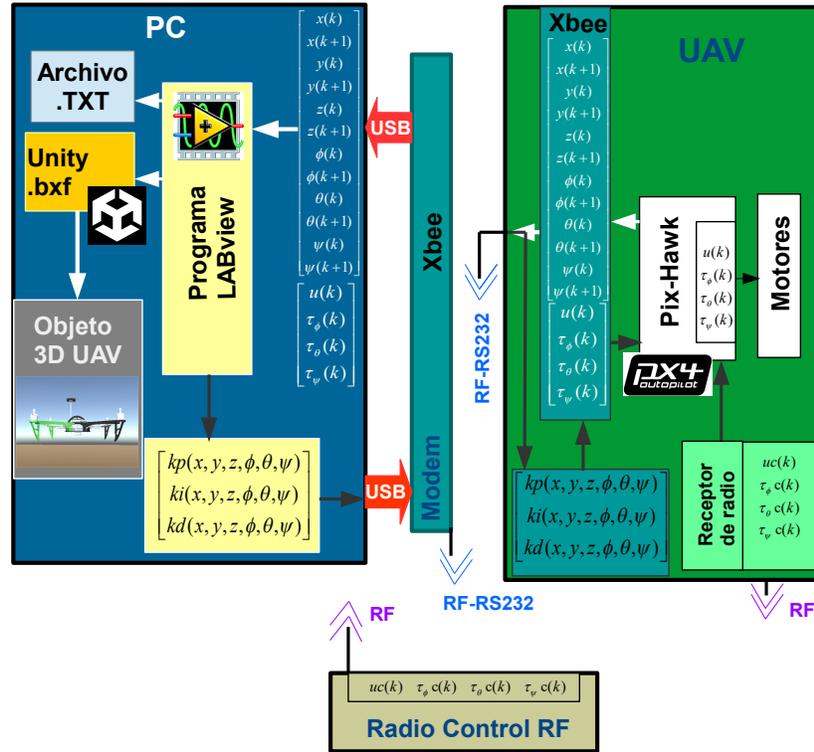


Figura 3.8: Esquema de variables de la planta UAV.

Mediante la interfaz de LABview[®] se ejecuta un sistema SCADA, para monitoreo del UAV; sin embargo, también se puede manipular el comportamiento del firmware del PIX-HAWK[®] mediante el envío de datos por el mismo protocolo antes mencionado, en la Figura 3.8 Se muestra por ejemplo, el envío de ganancias para un controlador PID, aplicado en la posición y orientación del UAV.

El UAV puede recibir señales de control o ajuste de *off-set* del radiocontrol mediante el receptor que posee, estas señales son los tres torques y la fuerza ($\tau_\phi C(k)$, $\tau_\theta C(k)$, $\tau_\psi C(k)$ y $uC(k)$) que el usuario puede manipular manualmente, con ellas y el algoritmo de control, el controlador de vuelo PIX-HAWK[®] hace el cálculo de las señales de control a ser

aplicadas a los motores del drone, dichas señales son: $\tau_\phi(k)$ $\tau_\theta(k)$ $\tau_\psi(k)$ y $u(k)$.

El controlador de vuelo PIX-HAWK® recibe ganancias o variables de configuración, para que éste pueda adaptarse a las necesidades que el usuario establezca mediante la interfaz gráfica, dichas señales llegan a él de forma análoga que las que envía a la PC, es decir, por medio de una cadena de caracteres, con determinado formato que permite decodificar y ajustar valores, de por ejemplo, las ganancias del controlador como se aprecia en el diagrama 3.8. Para hacer la configuración de dichos parámetros por seguridad se realizan, cuando el UAV no está en vuelo y como candado de seguridad se debe considerar el estado en la perilla del radio, si ésta tiene determinado valor y el drone no está armado pueden hacerse cambios en las constantes del controlador, de lo contrario no se podrá.

El diagrama de flujo mostrado en la Figura A.3, ilustra el comportamiento que tiene el algoritmo cargado en el UAV de forma general, como puede verse en él, la primera etapa corresponde a la configuración que realiza la PC al UAV, por medio de comunicación serial inalámbrica, seguidamente, si el radio se encuentra prendido y mandando señal al receptor, es posible realizar la configuración de los parámetros del controlador y éstos son asignados mediante un botón de validación, lo anterior se puede realizar mientras no se presione el botón armar, ya que a partir de ello el PIX-HAWK® entrará a su rutina de control (orientación) y aproximación de posición del UAV, aplicando para este caso un controlador PID (mostrado en Apéndices) y un filtro de Kalman, además en dicha rutina se contempla el envío de datos hacia la PC, para que la interfaz SCADA pueda acceder a los datos de trayectoria y señales de control, esto se realiza cíclicamente hasta que la comunicación es perdida o el usuario decide aterrizar el drone.

3.3. Plataforma experimental ULV

La plataforma experimental, creada para el vehículo terrestre no tripulado (ULV), se basa en una arquitectura de tipo Shrimp Rover [27], que es un sistema encontrado en la literatura y que ha demostrado por su estructura, versatilidad y adaptabilidad a terrenos irregulares [28], lo que le permite moverse en entornos de difícil acceso para otros robots[29]. De forma análoga al UAV, esta plataforma también cuenta con un radiocontrol y puede interactuar con el sistema SCADA, siendo su controlador un PIX-HAWK PX4®), cuenta, además, con un sistema de radar basado en un RPLIDAR A2M8® (ver Figura B.2), que le permite recalcular su trayectoria en caso de que el robot encuentre

un obstáculo con las dimensiones necesarias para que éste no pueda pasar sobre él. Esto último debido a que el robot incluye en sus componentes electrónicos un sistema de navegación global (GPS), que es usado para calcular la posición inicial del robot y como sensor de posición para calcular el error en tiempo real de la ubicación del rover respecto a la referencia deseada, que envía el UAV. En la Figura 3.9, pueden verse los componentes de la plataforma experimental del ULV y las partes que la integran.

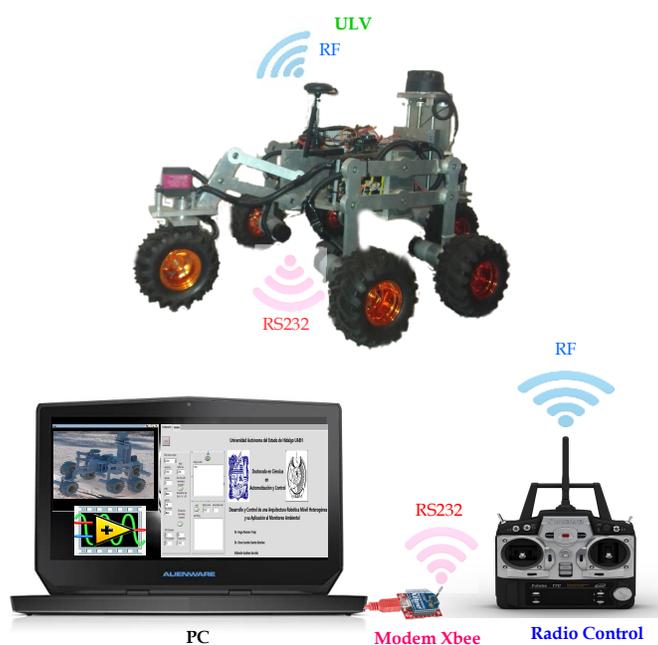


Figura 3.9: Plataforma experimental ULV.

El diseño asistido por computadora del robot todoterreno es mostrado en los planos de las Figuras B.17, B.18 y B.19, en ellas puede apreciarse detalladamente, la incorporación de sistema de recolección de muestras de suelo, el cual está basado en un carro transversal orientado hacia la parte baja del robot, y ubicado al centro de éste. Las direcciones de las ruedas delantera y trasera son manipuladas por servomotores, los cuales obtienen la señal de control directamente de una salida PWM del controlador PIX-HAWK®), cada rueda tractora de robot, que en total tiene 6, es manipulada por un driver cuyo controlador principal es un ESP32®, el cual recibe una señal de referencia originada por un PWM del PIX-HAWK®, de esta forma se aplica un algoritmo de control PID adaptable en cada rueda generando como salida del ESP32® dos señales analógicas una para cada dirección

del motor, estas señales son enviadas a un puente H ULM297 que permitir manipular la velocidad y dirección de cada motor (ver Figura B.19).

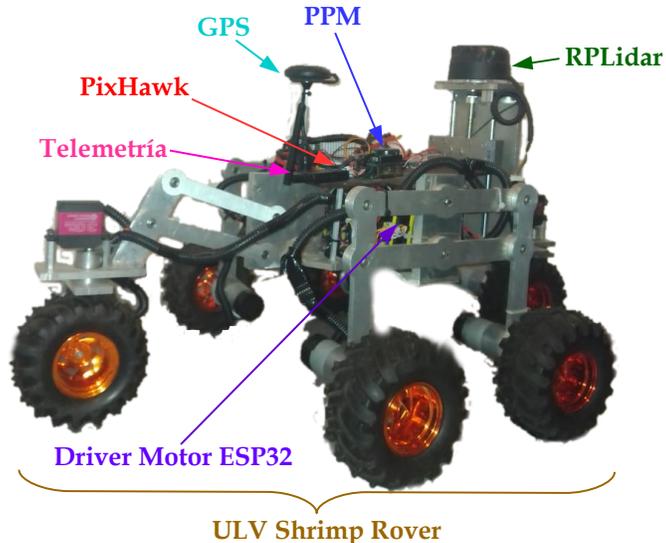


Figura 3.10: Carro todo terreno basado en una plataforma shrimp rover.

Las variables que forman parte de la interacción de los elementos del robot shrimp rover, se pueden observar sobre la Figura 3.11. La PC contiene una interfaz gráfica creada en LABview® la cual interactúa a su vez con un programa ejecutable creado en Unity®, que permite manipular el ensamble de un objeto 3D que imita el comportamiento del ULV en el espacio de trabajo, como entradas la PC recibe a través de protocolo RS232 Serial mediante un modem Xbee S1, conectado por USB, una cadena de caracteres que contiene el estado del sistema así como las señales de control. Dichas señales se emplean para generar un archivo de texto que permite guardar las trayectorias y controles del sistema, además mediante el estado se genera la manipulación de un objeto 3D que imita al robot terrestre, dicho objeto está creado con formato .BXF en Blender® y es exportado a partir de un archivo .WRL generado en Solid Works®, en el plano B.19 es mostrado el diseño realizado para dicho ensamble 3D.

El sistema SCADA creado en LABview®, también permite el monitoreo de las variables correspondientes al ULV, de forma analoga a como se realiza con el UAV, brindando la posibilidad de, además, manipular el comportamiento del firmware del PIX-HAWK® encargado embarcado en el vehículo todoterreno, esto, mediante el envío de datos por

del mismo protocolo (RS232 analámbrico), en la Figura 3.11, se muestra por ejemplo, el envío de ganancias para un controlador PID, aplicado en la posición y orientación del ULV, desde la PC hacia el robot, y el estado y señales de control desde el sentido contrario.

El rover puede recibir señales de control o ajuste, del radio RF, estas señales son: la correspondiente al control de ángulo de orientación del robot $\tau_\theta c(k)$, la de control de velocidad de desplazamiento lineal $uc(k)$, la activación del perforador $ud(k)$ y la señal de control para el sistema de succión de muestras $uv(k)$, que el usuario puede manipular manualmente por medio del radiocontrol, con ellas y el algoritmo de control, el controlador PIX-HAWK® hace el cálculo de las señales de control a ser aplicadas a los motores de ruedas de tracción ($\omega_n(k)$ $n = 1, 2, \dots, 6$), de dirección ($\varphi_1(k)$, $\varphi_2(k)$), de succión $v(k)$, de velocidad de avance del perforador $D_d(k)$ y de velocidad del rotor $\omega_d(k)$ del motor de perforación del ULV.

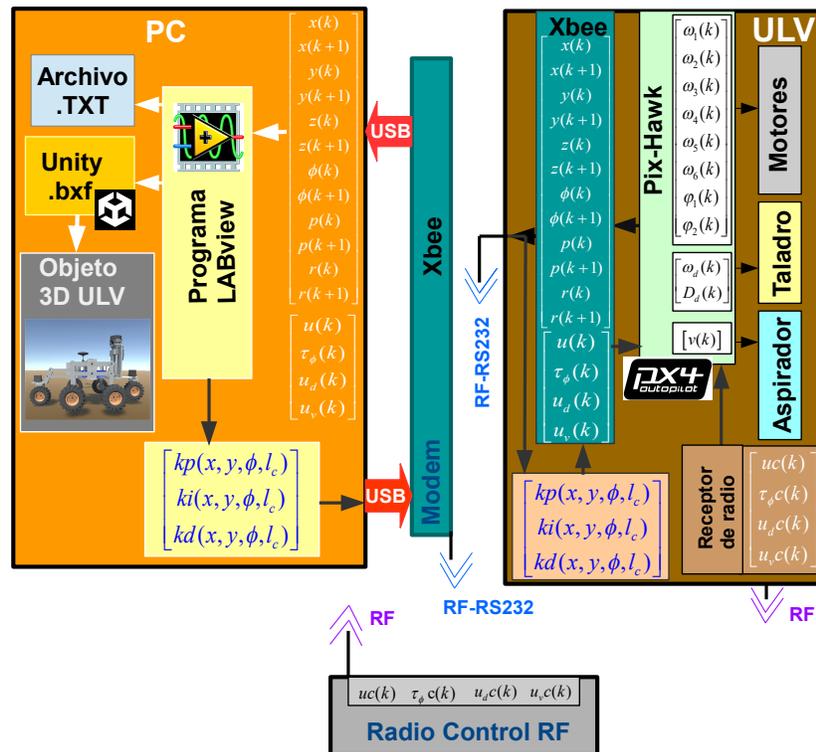


Figura 3.11: Esquema de variables de la planta ULV.

El controlador PIX-HAWK® recibe ganancias o variables de configuración, para que éste pueda adaptarse a las necesidades que el usuario establezca mediante la interfaz gráfica del sistema SCADA, dichas señales llegan a él desde la PC por protocolo RS232 inalámbricamente, a través de una cadena de caracteres, con determinado formato que permite decodificar y ajustar valores, de por ejemplo las ganancias del controlador como se aprecia en el diagrama 3.11. Para hacer la configuración de dichos parámetros por seguridad se realizan, cuando el ULV no está en operación y como candado de seguridad se debe considerar el estado en la perilla del radio, si ésta tiene determinado valor y el rover no está armado pueden hacerse cambios en las variables correspondientes al algoritmo de control o incluso modificar que tipo de algoritmo se usará, por ejemplo un controlador PID o un controlador LQR, etc.

3.4. Sistema SCADA

El sistema SCADA (Supervisory Control and Data Acquisition, es decir, Supervisión, Control y Adquisición de Datos) se basa en una interfaz que contiene tres ventanas principales, una para configuración de parámetros de comunicación y control, otra para el monitoreo y observación del comportamiento del UAV y la última para el monitoreo de las variables del ULV.

En la Figura 3.12 se muestra la pantalla que corresponde a la configuración de comunicación y ganancias del controlador, como se ve en ella se pueden ajustar los parámetros de comunicación para el protocolo RS232 serial inalámbrico usado con el Xbee®, para este caso particular dicha conexión se realiza a 115200 bps, sin control de flujo ni paridad, con un bit de parada y un formato de 8 bits por dato. Teniendo una conexión full dúplex. Esto es asignado en la región denominada “*Configuración de comunicación RS232*”.

Las ganancias del controlador (para el caso de lo mostrado en la Figura 3.12 un PID) son asignadas y enviadas en el apartado “*Asignación de ganancias*”, para el ejemplo mostrado permiten ajustar valores a los controladores de posición y orientación, contar con esta posibilidad en la interfaz del sistema SCADA permite modificar las ganancias de los controladores tanto para posición como orientación, ambos robots, sin necesidad de modificar el firmware del controlador programable ubicado en cada robot.

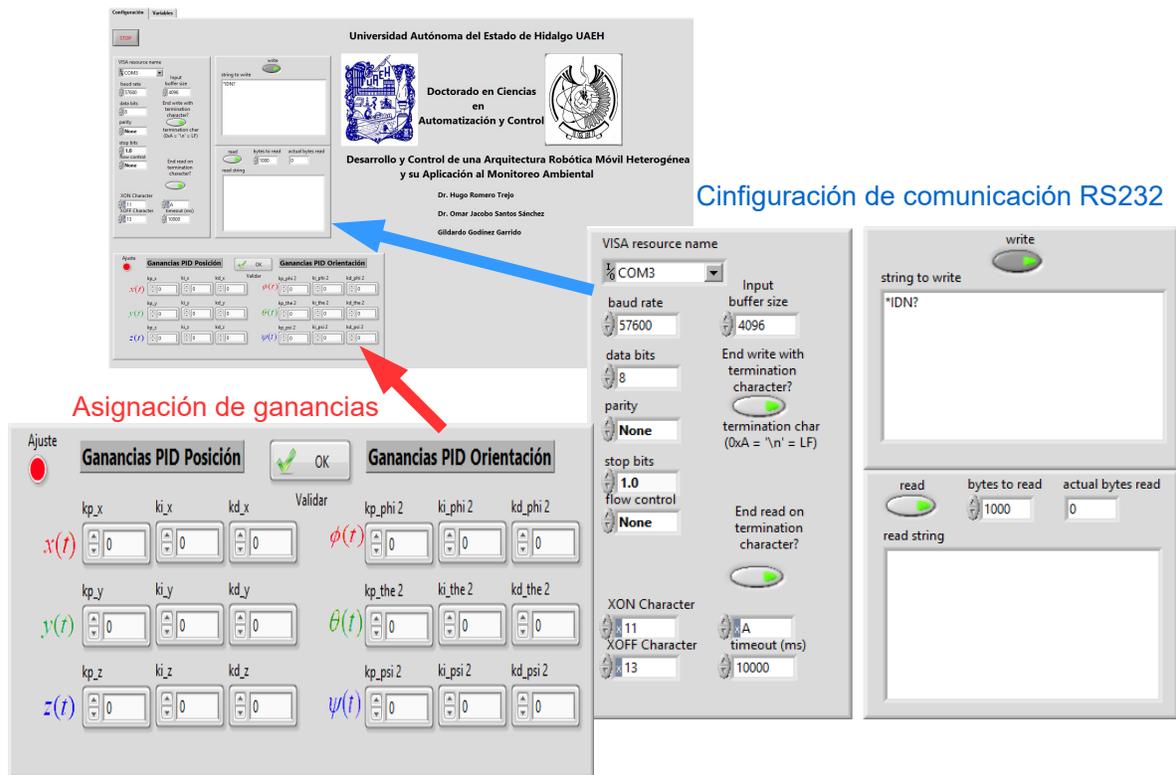


Figura 3.12: Interfaz pantalla de configuración.

En la segunda pantalla de la interfaz (Figura 3.13) se muestra el monitoreo de las variables de estado del UAV y de las cuatro señales de control generadas por el controlador, además se incluye un objeto 3D en formato VRML que permite que el usuario aprecie de manera gráfica el movimiento del UAV en el espacio, esto se logra mediante la conversión de los ángulos cabeceo, alabeo y guiñada a un cuaternión, ya que la visualización se realiza empleando MATLAB®[®], y se requiere dar la orientación en con ese formato. Las variables son mostradas de acuerdo a su unidad y ordenadas en tríadas en las gráficas mostradas en la interfaz, así los tres ángulos de orientación son graficados en conjunto, las tres posiciones, las velocidades para orientación y posición, los tres torques y la única variable que se grafica sola es la fuerza $u(t)$. Todos estos datos son almacenados de forma automática en un archivo que tiene por nombre “*datos.TXT*”, el cual posteriormente permite su interpretación y generación de gráficas para uso general.

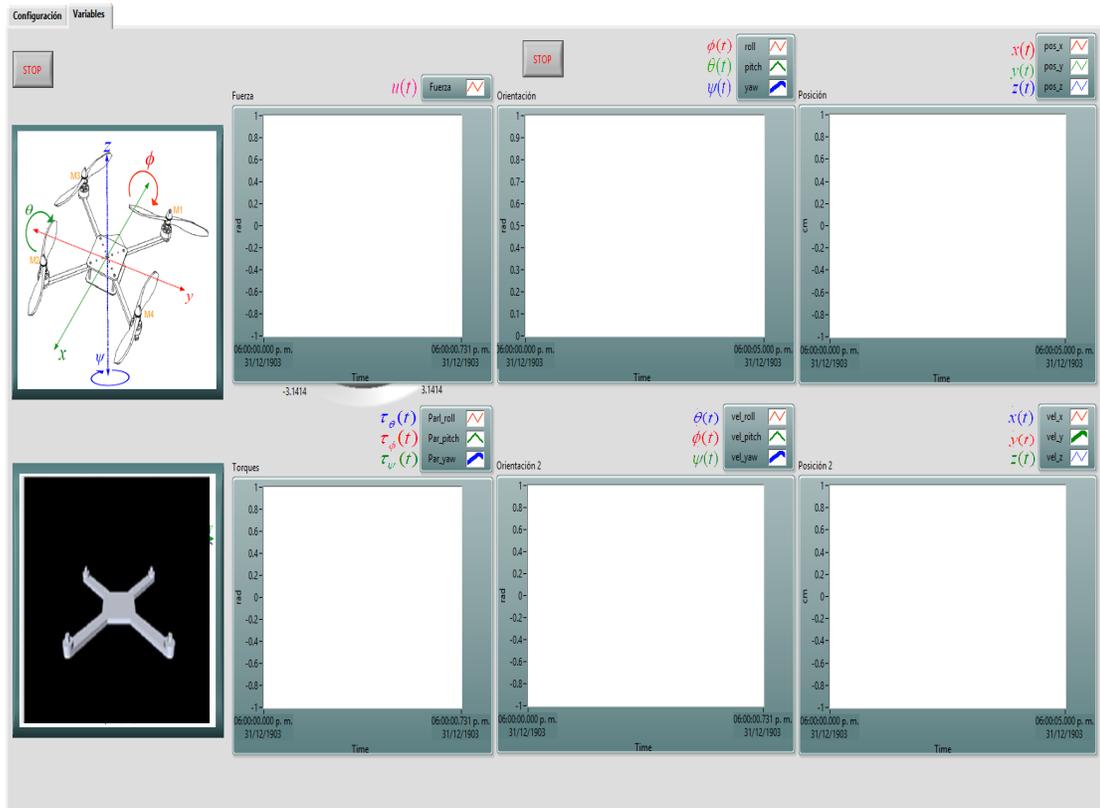


Figura 3.13: Interfaz monitoreo de variables del UAV.

El comportamiento general del algoritmo de la interfaz gráfica que rige al sistema SCADA es mostrado en la Figura A.2, como puede observarse en él, primero se realiza la configuración de la comunicación seguida de su validación, a continuación se configuran las ganancias del controlador que se asignarán al UAV, mientras la comunicación entre la PC y el dron no presente algún problema, se realizan las siguientes tareas, lectura del estado del UAV, generación del archivo para guardar trayectorias y señales de control, cálculo de cuaternion a partir de los ángulos de Euler, modificación de objeto 3D que imita el movimiento del UAV mediante la interacción con MATLAB® en un archivo VRML.

Debido a la potencia y prestaciones que el programa Unity® tiene, se robusteció la interfaz gráfica integrando un ambiente virtual, creado en este programa, con la finalidad de mejorar la interacción con el usuario, y generar la posibilidad de crear simulaciones

previas y validación de trayectorias, en dicho ambiente virtual al ejecutar la tarea de cada robot. La Figura 3.14 muestra el sistema SCADA, al mismo tiempo, como puede verse, además, se incluye a la derecha una imagen del robot aéreo, en ella se muestra en la parte superior los marcos de referencia que tiene cada robot, correspondientes a cada uno de los grados de libertad inherentes a dichos marcos, para que el usuario, comprenda a que corresponde cada una de las variables indicadas en los gráficos, y se pueden notar la ruta donde se graba el archivo de texto y las cadenas de caracteres de entrada y salida correspondiente a los datos enviados desde la PC a los robots y viceversa, así como, los parámetros de configuración del protocolo de comunicación, todos esto puede ser ajustado por el usuario, con la finalidad de adaptar a las necesidades de la o las tareas a ejecutar. Esta ventana se encuentra en la viñeta nombrada “Configuración”, en las otras dos viñetas “UAV”, y “ULV”, se puede acceder respectivamente a los datos enviados por cada robot.

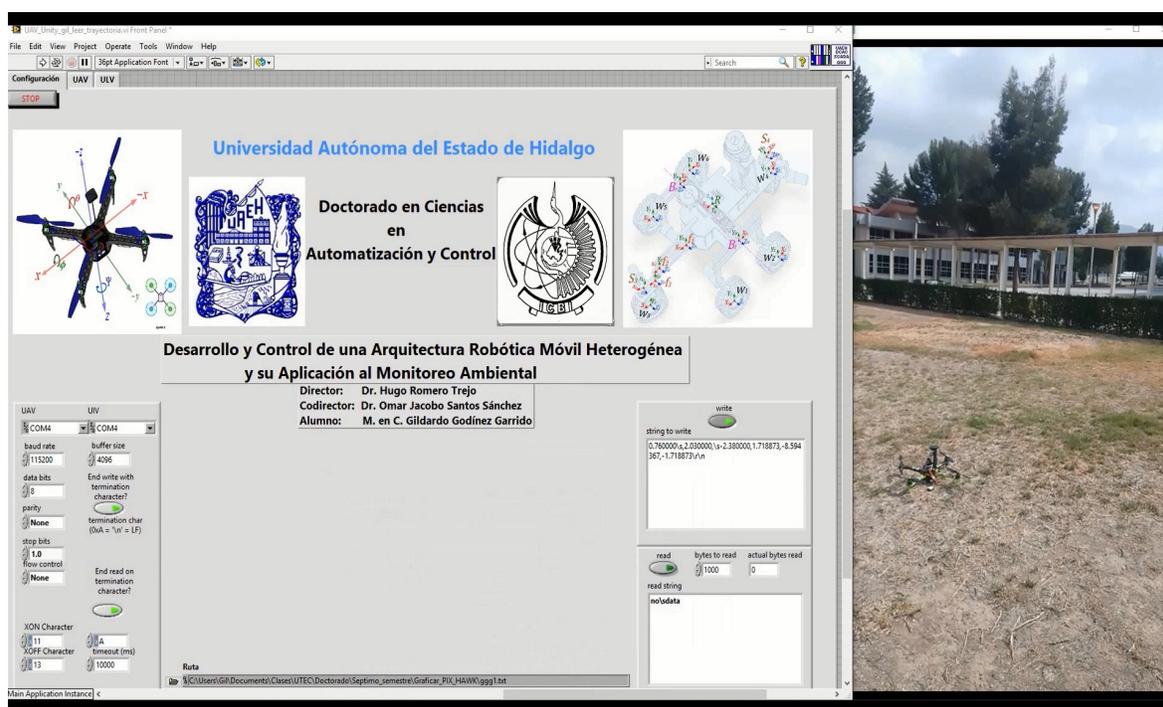


Figura 3.14: Interfaz del sistema SCADA para ambos robots.

En caso de seleccionar la viñeta “UAV”, se puede acceder a la información del robot aéreo, la Figura 3.15, muestra un ejemplo de algunas variables correspondientes al cuadricóptero, correspondientes a la posición y orientación del robot, así como, un ambiente 3D donde puede apreciarse el desempeño y trayectoria seguida por el robot, e instrumentos de navegación, se observan, un indicador de orientación y una brújula, los cuales obtienen los datos en tiempo real del sistema de navegación inercial del robot.

Si la viñeta seleccionada es “ULV”, entonces, se muestran indicadores correspondientes a la instrumentación del carro todoterreno, la Figura 3.15, ilustra los controles usados para mover el robot todoterreno en el ambiente virtual, además, en dicho ambiente pueden apreciarse ambos robots, lo que demuestra que la interface es capaz de interactuar simultáneamente con ambas plataformas, para cada uno de los grados de libertad del rover se muestra en LABview® a cada elemento del robot corresponde, permitiendo que la interacción sea intuitiva e inmersiva.

La Figura 3.16 permite observar ambos robots en el ambiente virtual y al mismo tiempo, los controles manuales creados en sistema SCADA para manipular el objeto 3D correspondiente al ULV, en dicho ambiente; mientras que en la Figura 3.17, se pueden observar los diferentes instrumentos virtuales empleados para el monitoreo de las variables del robot todoterreno, como por ejemplo gráficas para monitoreo de las posiciones y ángulos, así como las velocidades de éstas y éstos, una gráfica polar para el sistema de detección de obstáculos en tiempo real con el sensor RPLidar, e indicadores para saber en qué momentos son activados los actuadores del sistema de succión y avance del carro que permite tomar las muestras de suelo, esto por medio de LEDs indicadores que están en la interfaz. Puede notarse también el plano de trabajo XY, que permite ubicar la posición del robot, en su área de trabajo. El ambiente virtual permite monitorear el comportamiento del robot en tiempo real, cuando el usuario está interactuando con cada uno de los robots mediante el sistema SCADA, pero además, permite que la fuera de línea se replique el comportamiento de los robots, mediante la lectura de los archivos que se generaron en cada experimento, o directamente de los datos extraídos de la memoria micro SD del controlador.

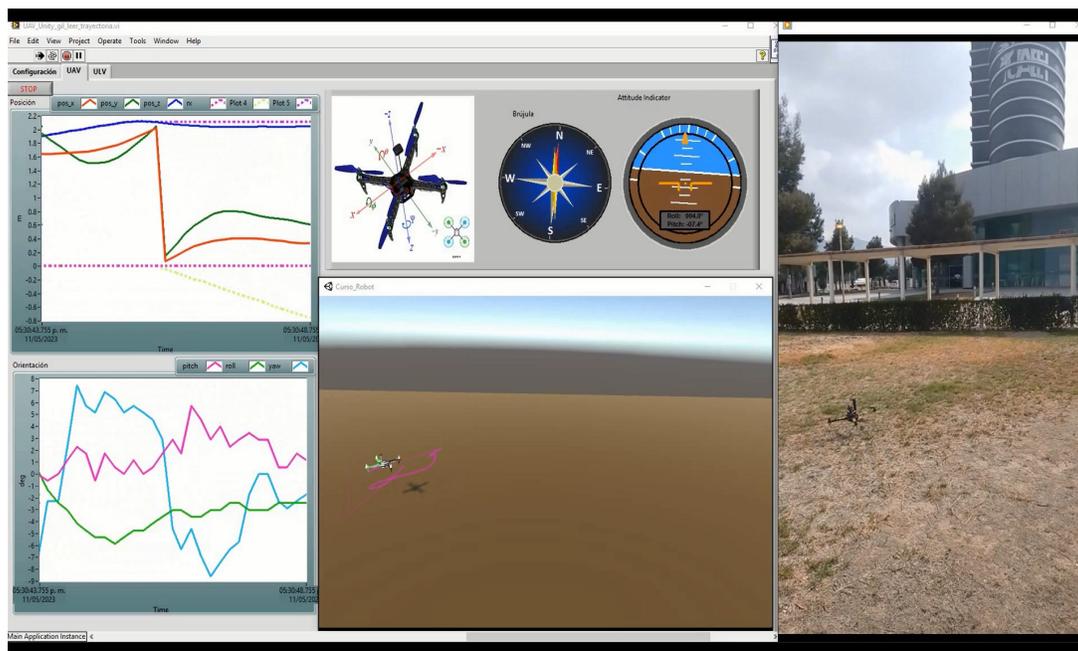


Figura 3.15: Interfaz monitoreo de variables del UAV LABview® y Unity®.

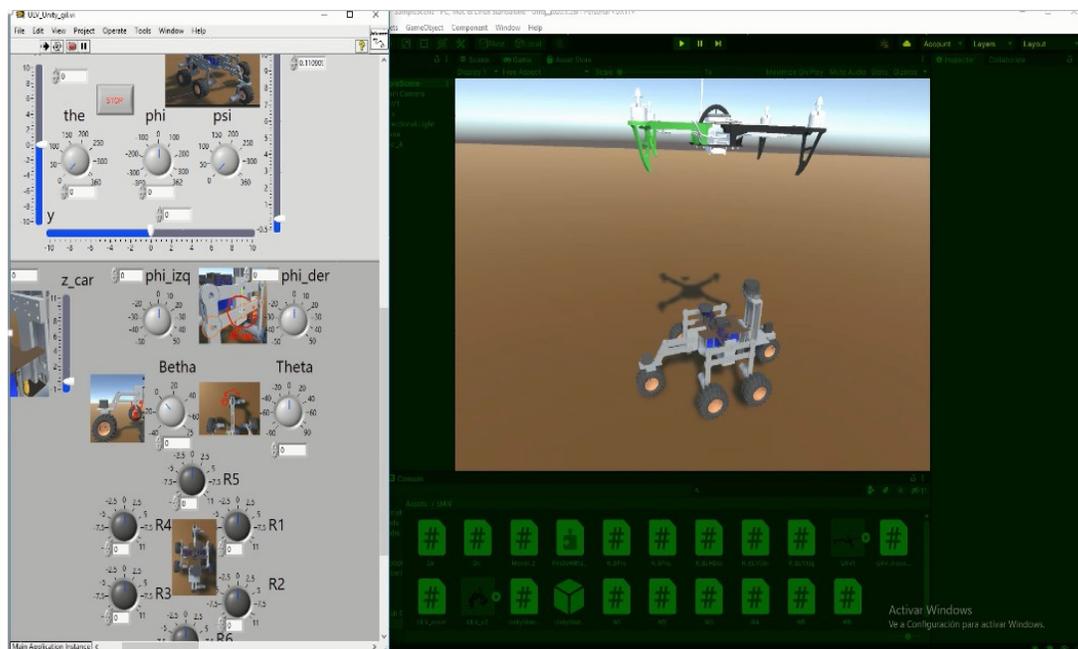


Figura 3.16: Interfaz monitoreo de variables del ULV LABvie® y Unity®.

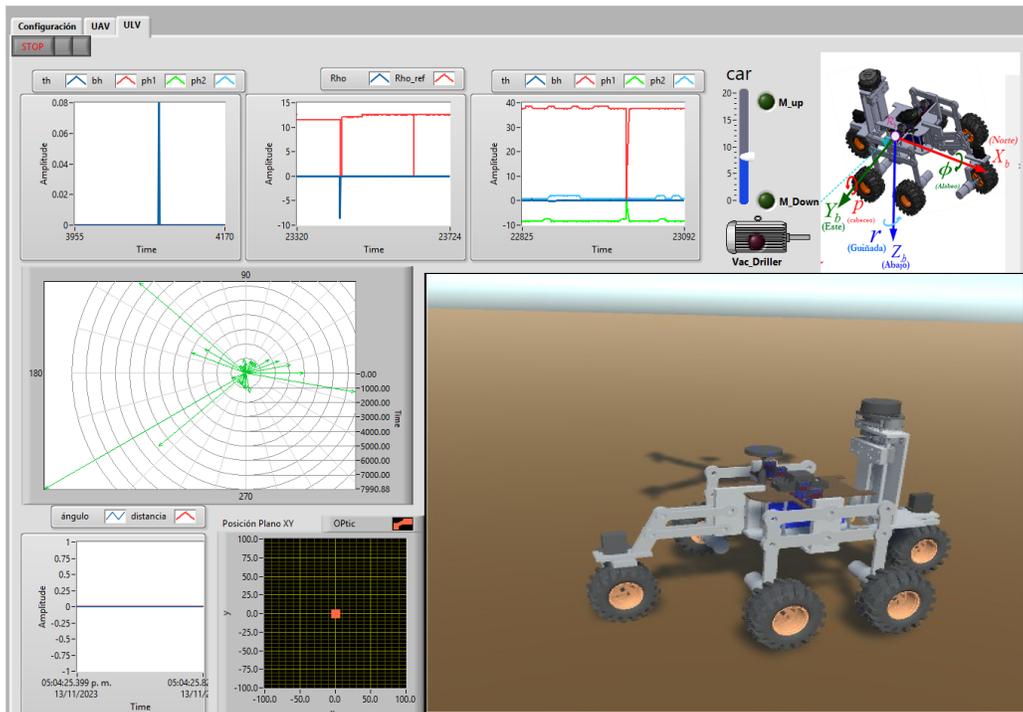


Figura 3.17: Interfaz monitoreo de variables del ULV LABview®.

3.5. Conclusión capítulo 3

En este capítulo se presenta detalladamente la Plataforma Robótica Móvil Heterogénea, una propuesta innovadora y propia que juega un papel crucial en el logro de los objetivos general y específicos de esta investigación. Cada uno de los sistemas descritos dentro de esta plataforma no solo refleja un diseño original, sino que también es esencial para el monitoreo de contaminantes en el aire y la recolección de muestras de suelo, resaltando la versatilidad y la eficacia de la propuesta.

La Descripción del Sistema Propuesto introduce la estructura general y las funcionalidades de la plataforma, sentando las bases para un entendimiento integral del proyecto. Dentro de esta sección, se hace especial hincapié en el Sistema de Monitoreo de Calidad de Aire y Visión, y en el Sistema de Recolección de Muestras de Suelo. Estos sistemas no solo demuestran una aplicación práctica y directa de la robótica en la monitorización ambiental, sino que también destacan por su diseño innovador y su capacidad para reco-

lectar datos valiosos de manera eficiente.

La Plataforma Experimental UAV (Vehículo Aéreo No Tripulado) y la Plataforma Experimental ULV (Vehículo Ligero No Tripulado) son exploradas en detalle, mostrando cómo estas tecnologías pueden ser empleadas para la recopilación de datos ambientales. Estas plataformas son fundamentales para el proyecto, ya que permiten realizar tareas de monitoreo y recolección de datos en diferentes entornos y condiciones.

Finalmente, se presenta el Sistema SCADA (Control de Supervisión y Adquisición de Datos), una parte integral del proyecto que facilita la supervisión, el control y la adquisición de datos en tiempo real. Este sistema no solo mejora la eficiencia y la efectividad de la plataforma robótica, sino que también asegura la precisión y la fiabilidad de los datos recogidos.

En resumen, este capítulo demuestra cómo cada componente de la Plataforma Robótica Móvil Heterogénea contribuye de manera significativa al logro de los objetivos planteados en esta investigación. Con su enfoque innovador y sus diseños propios, esta plataforma representa un avance significativo en la robótica aplicada al monitoreo ambiental, especialmente en la medición de contaminantes en el aire y la toma de muestras de suelos.

Capítulo 4

Modelado matemático de los robots móviles

Los modelos matemáticos usados para la simulación y diseño de los algoritmos de control usados en cada robot, son mostrados en este capítulo, primero el modelo del UAV es abordado, y a continuación del carro todoterreno, ambos modelos son obtenidos de trabajos previos, los cuales han demostrado en múltiples aplicaciones ser funcionales y coherentes con la dinámica de cada robot respectivamente.

4.1. Modelo matemático del vehículo aéreo

En este trabajo se emplea el modelo simplificado no lineal mostrado en la Figura [3], el cual es un modelo dinámico de un helicóptero con cuatro rotores, este modelo ha sido obtenido aplicando el método de Euler-Lagrange. Sean \mathcal{R}_w y \mathcal{R}_b dos sistemas de coordenadas. \mathcal{R}_w es definido como el sistema de coordenadas inercial de referencia y \mathcal{R}_b es otro sistema de coordenadas ligado al cuerpo del robot volador. El vector de coordenadas generalizado para el robot es $v \in \mathbb{R}^6$, descrito como:

$$v = \left(\hat{x} \ y \ z \ \psi \ \theta \ \phi \right)^T, \quad (4.1)$$

Donde los vectores $(\hat{x} \ y \ z)$ representa la posición del centro de masa del robot volador respecto al sistema \mathcal{R}_w . El vector compuesto por $(\psi \ \theta \ \phi)$ define los tres ángulos de Euler,

el cabeceo, el alabeo y la guiñada respectivamente. Con ellos es representada la orientación del robot.

De manera que se tienen dos subsistemas, un subsistema ξ referente a la parte traslacional y otro η para la parte rotacional, es decir, $\xi = \begin{pmatrix} \hat{x} & y & z \end{pmatrix}^T \in \mathbb{R}^3$, $\eta = \begin{pmatrix} \psi & \theta & \phi \end{pmatrix}^T \in \mathbb{R}^3$, ver Figura 4.1.

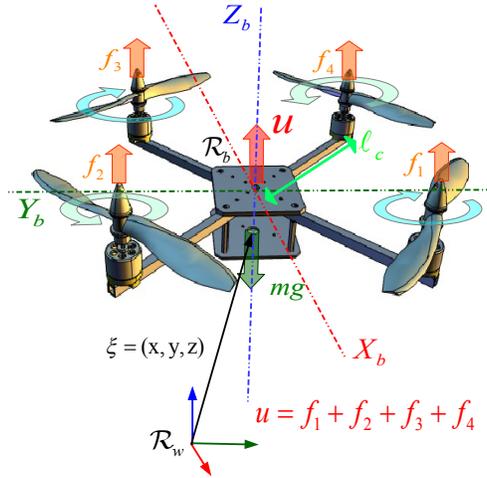


Figura 4.1: Esquema de helicóptero con cuatro rotores.

La energía cinética de traslación del helicóptero está dada por la expresión:

$$T_{tras} = \frac{m}{2} \dot{\xi}^T \dot{\xi}, \quad (4.2)$$

Donde m representa la masa del helicóptero, contemplando ahora la parte rotacional, la energía cinética es dada por:

$$T_{rot} = \frac{1}{2} \Omega^T \mathbb{I} \Omega, \quad (4.3)$$

Donde Ω es el vector de velocidades angulares e \mathbb{I} es la matriz de inercia. Sea W el vector de velocidades angulares expresado respecto al marco de referencia \mathcal{R}_b , éste está ligado directamente con las velocidades generalizadas $\dot{\eta}$ por la relación cinemática estándar [30].

$$\Omega = W_\eta \dot{\eta}, \quad (4.4)$$

Reemplazando (4.33) en (4.3) se tiene:

$$T_{rot} = \frac{1}{2} \dot{\eta}^T W_\eta^T \mathbb{I} W_\eta \dot{\eta}, \quad (4.5)$$

Siendo.

$$W_\eta = \begin{bmatrix} -\text{sen}(\theta) & 0 & 1 \\ \text{sen}(\phi) \cos(\theta) & \cos(\phi) & 0 \\ \cos(\phi) \cos(\theta) & -\text{sen}(\phi) & 0 \end{bmatrix}, \quad \mathbb{I} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}, \quad (4.6)$$

Entonces el vector Ω está dado por:

$$\Omega = \begin{bmatrix} \dot{\phi} - \text{sen}(\psi) \\ \dot{\psi} \text{sen}(\phi) \cos(\theta) + \dot{\theta} \cos(\phi) \\ \dot{\psi} \cos(\phi) \cos(\theta) - \dot{\theta} \text{sen}(\phi) \end{bmatrix}, \quad (4.7)$$

Definiendo:

$$\mathbb{J}(\eta) \triangleq W_\eta^T \mathbb{I} W_\eta, \quad (4.8)$$

Implica que la ecuación (4.5) se puede escribir como:

$$T_{rot} = \frac{1}{2} \dot{\eta}^T \mathbb{J} \dot{\eta}, \quad (4.9)$$

Donde \mathbb{J} es la matriz de inercia para la energía cinética de rotación, expresada en bajo las coordenadas generalizadas η .

La fuerza que contribuye a la energía potencial presentada en el sistema dinámico es la fuerza U debida a la aceleración gravitacional, expresada como:

$$U = mgz, \quad (4.10)$$

Siendo z la altura del helicóptero. Considerando las ecuaciones anteriores el Lagrangiano está dado por:

$$\begin{aligned} \mathcal{L}(g, \dot{g}) &= T_{tras} + T_{rot} - U, \\ &= \frac{m}{2} \dot{\xi}^T \dot{\xi} + \frac{1}{2} \dot{\eta}^T \mathbb{J} \dot{\eta} - mgz, \end{aligned} \quad (4.11)$$

Para obtener el modelo dinámico del helicóptero, se consideran las ecuaciones de Euler-Lagrange, tomando en cuenta el vector de fuerzas externas generalizado, esto es:

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{v}} \right) - \left(\frac{\partial \mathcal{L}}{\partial v} \right) = F, \quad (4.12)$$

Donde $F = (F_\xi, \tau)$. F_ξ es el vector de fuerzas aplicado al robot aéreo. Dichas fuerzas son debidas a las entradas de control aplicadas para producir movimientos de traslación y τ es el vector de momentos generalizados producidos por las entradas de control aplicadas para efectuar movimientos de rotación. Todas las otras fuerzas aerodinámicas presentadas en el sistema no son tomadas en cuenta debido a que son despreciables comparadas con las fuerzas producidas por la entradas de control. Las fuerzas aplicadas al mini helicóptero respecto al marco de referencia \mathcal{R}_b son:

$$F_b = \begin{bmatrix} 0 \\ 0 \\ u \end{bmatrix}, \quad (4.13)$$

donde:

$$u = f_1 + f_2 + f_3 + f_4, \quad (4.14)$$

$$f_i = k_i \omega_i^2, \quad i = 1, \dots, 4, \quad (4.15)$$

Donde $k_i > 0$ es un parámetro que depende de la densidad del aire ρ_a , del área A_h de la hélice, de su forma y del ángulo de sus palas. ω_i es la velocidad angular de cada motor

i ($M_i, i=1, \dots, 4$), este conjunto de fuerzas expresado en \mathcal{R}_w se transforma en:

$$F_\xi = \mathbf{R}F_b, \quad (4.16)$$

Donde $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ es la matriz de rotación del mini helicóptero de cuatro rotores, ésta es:

$$\mathbf{R} = \begin{bmatrix} c(\theta)c(\psi) & s(\phi)s(\theta)c(\psi) - s(\psi)c(\phi) & s(\phi)s(\psi) + s(\theta)c(\phi)c(\psi) \\ s(\psi)c(\theta) & s(\phi)s(\theta)s(\psi) + c(\phi)c(\psi) & s(\theta)s(\psi)c(\phi) - s(\phi)c(\psi) \\ -s(\theta) & s(\phi)c(\theta) & c(\phi)c(\theta) \end{bmatrix}, \quad (4.17)$$

Esta representación de la matriz \mathbf{R} es muy empleada en aeronáutica [30]. El vector de momentos generalizado sobre las variables de η es:

$$\tau = \begin{bmatrix} \tau_\psi \\ \tau_\theta \\ \tau_\phi \end{bmatrix}, \quad (4.18)$$

siendo:

$$\begin{aligned} \tau_\psi &= \sum_{i=1}^4 \tau_{M_i}, \\ \tau_\theta &= (f_2 - f_4)\ell_c, \\ \tau_\phi &= (f_3 - f_1)\ell_c, \end{aligned} \quad (4.19)$$

ℓ_c es la distancia entre los ejes de los motores y el centro de gravedad del helicóptero y τ_{M_i} es el momento producido por el motor M_i alrededor del centro de gravedad del vehículo.

Debido a que el Lagrangiano no contiene términos en la energía cinética combinando $\dot{\xi}$ con $\dot{\eta}$, las ecuaciones de Euler-Lagrange pueden ser divididas en las dinámicas para las coordenadas de ξ y las coordenadas de η .

La ecuación de Euler-Lagrange para el movimiento de traslación es:

$$\frac{d}{dt} \left[\frac{\partial \mathcal{L}_{tras}}{\partial \dot{\xi}} \right] - \left(\frac{\partial \mathcal{L}_{tras}}{\partial \xi} \right) = F_{\xi}, \quad (4.20)$$

se obtiene:

$$m \ddot{\xi} + \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} = F_{\xi}, \quad (4.21)$$

De manera similar, para las coordenadas de η se puede escribir:

$$\frac{d}{dt} \left[\frac{\partial L_{rot}}{\partial \dot{\eta}} \right] - \left(\frac{\partial L_{rot}}{\partial \eta} \right) = \tau, \quad (4.22)$$

obteniendo:

$$\begin{aligned} \tau &= \frac{d}{dt} \left[\dot{\eta}^T \mathbb{J} \frac{\partial \dot{\eta}}{\partial \dot{\eta}} \right] - \left(\frac{1}{2} \frac{\partial}{\partial \eta} (\dot{\eta}^T \mathbb{J} \dot{\eta}) \right), \\ &= \mathbb{J} \ddot{\eta} + \dot{\mathbb{J}} \dot{\eta} - \frac{1}{2} \frac{\partial}{\partial \eta} (\dot{\eta}^T \mathbb{J} \dot{\eta}), \end{aligned} \quad (4.23)$$

Definiendo el vector de Coriolis como:

$$\mathcal{C}(\eta, \dot{\eta}) = \dot{\mathbb{J}} \dot{\eta} - \frac{1}{2} \frac{\partial}{\partial \eta} (\dot{\eta}^T \mathbb{J} \dot{\eta}), \quad (4.24)$$

Reemplazando el vector $\mathcal{C}(\eta, \dot{\eta})$ en (4.23), se tiene:

$$\mathbb{J} \ddot{\eta} + \mathcal{C}(\eta, \dot{\eta}) = \tau, \quad (4.25)$$

Reescribiendo (4.23), se obtiene:

$$\mathbb{J} \ddot{\eta} + \left(\dot{\mathbb{J}} - \frac{1}{2} \frac{\partial}{\partial \eta} (\dot{\eta} \mathbb{J}) \right) \dot{\eta} = \tau, \quad (4.26)$$

$$\mathbb{J} \ddot{\eta} + \mathbb{C}(\eta, \dot{\eta}) \dot{\eta} = \tau \quad (4.27)$$

Donde $\mathbb{C}(\eta, \dot{\eta})$ es la matriz de coriolis, en ella están comprendidos los términos giroscópicos y centrífugos relacionados a la dependencia de η sobre \mathbb{J} . Con el fin de simplificar el análisis se propone un cambio de variable en la entrada.

$$\dot{\eta} = \tilde{\tau}, \quad (4.28)$$

lo que implica que:

$$\mathbb{J} \tilde{\tau} + \mathbb{C}(\eta, \dot{\eta}) \dot{\eta} = \tau, \quad (4.29)$$

$\tilde{\tau} = [\tilde{\tau}_\psi, \tilde{\tau}_\theta, \tilde{\tau}_\phi]^T$ es el vector con las nuevas entradas, las cuales son aplicadas directamente a los momentos angulares generalizados τ_ψ , τ_θ y τ_ϕ .

Por lo anterior el sistema puede ser reescrito como [31]:

$$\begin{aligned} F_\xi &= m\ddot{\xi} + mgE_z, \\ \dot{\eta} &= \tilde{\tau}, \end{aligned} \quad (4.30)$$

F_ξ está definido como:

$$F_\xi = \begin{bmatrix} -u \operatorname{sen}(\theta) \\ u \operatorname{sen}(\phi) \operatorname{cos}(\theta) \\ u \operatorname{cos}(\phi) \operatorname{cos}(\theta) \end{bmatrix}, \quad (4.31)$$

Finalmente al multiplicar se obtiene que un robot aéreo puede ser representado por el siguiente sistema de ecuaciones no lineales:

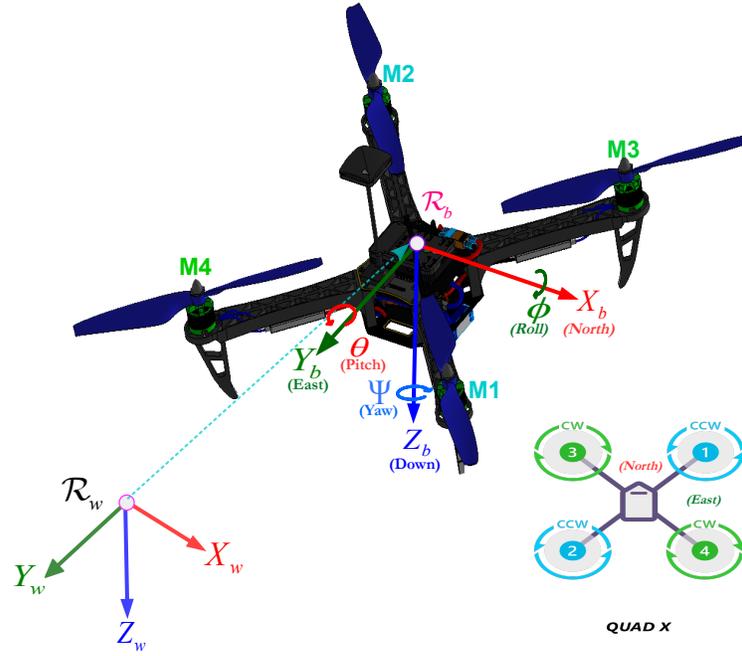


Figura 4.2: Posiciones y ángulos en un UAV.

$$\begin{aligned}
 m\ddot{x} &= -u \sin(\theta), \\
 m\ddot{y} &= u \cos(\theta) \sin(\phi), \\
 m\ddot{z} &= u \cos(\theta) \cos(\phi) - mg, \\
 \ddot{\phi} &= \tau_\phi, \\
 \ddot{\theta} &= \tau_\theta, \\
 \ddot{\psi} &= \tau_\psi,
 \end{aligned} \tag{4.32}$$

Donde \hat{x} , y son los desplazamientos en el plano horizontal y z es la posición vertical, ψ es el ángulo de guiñada en torno al eje z , θ es el ángulo de cabeceo respecto al eje y , ϕ es el ángulo de alabeo entorno al eje x . Las entradas de control son: u , $\tilde{\tau}_\phi$, $\tilde{\tau}_\theta$, y $\tilde{\tau}_\psi$. u es la fuerza generada por los motores para elevar el UAV. $\tilde{\tau}_\phi$, $\tilde{\tau}_\theta$, y $\tilde{\tau}_\psi$ son los torques generados en torno a cada eje respectivamente, como se muestra en las figuras 4.2 y 4.3.

$$\Omega = W_\eta \dot{\eta}, \tag{4.33}$$

La figura 4.3 ilustra las posiciones y los ángulos del UAV. Para hallar la representación

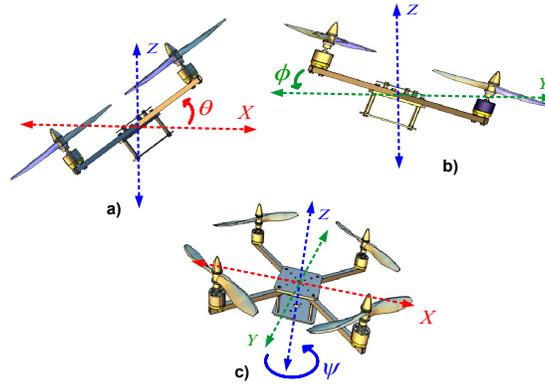


Figura 4.3: Orientación del UAV.

espacio estado del robot, a partir de (4.32) se definen las siguientes variables de estado:

$$x_1 = \hat{x}, \quad x_2 = \dot{\hat{x}}, \quad x_3 = y, \quad x_4 = \dot{y}, \quad (4.34)$$

$$x_5 = z, \quad x_6 = \dot{z}, \quad x_7 = \phi, \quad x_8 = \dot{\phi}, \quad (4.35)$$

$$x_9 = \theta, \quad x_{10} = \dot{\theta}, \quad x_{11} = \psi, \quad x_{12} = \dot{\psi}, \quad (4.36)$$

La representación espacio estado está dada por:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \\ \dot{x}_{10} \\ \dot{x}_{11} \\ \dot{x}_{12} \end{bmatrix} = \begin{bmatrix} x_2 \\ -\frac{u}{m} \sin(x_9) \\ x_4 \\ \frac{u}{m} \cos(x_9) \sin(x_7) \\ x_6 \\ \frac{u}{m} \cos(x_9) \cos(x_7) - g \\ x_8 \\ \tau_\phi \\ x_{10} \\ \tau_\theta \\ x_{12} \\ \tau_\psi \end{bmatrix}, \quad (4.37)$$

Que puede ser escrito como:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \\ \dot{x}_{10} \\ \dot{x}_{11} \\ \dot{x}_{12} \end{bmatrix} = \begin{bmatrix} x_2 \\ 0 \\ x_4 \\ 0 \\ x_6 \\ -g \\ x_8 \\ 0 \\ x_{10} \\ 0 \\ x_{12} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{\text{sen}(x_9)}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{\cos(x_9)\text{sen}(x_7)}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{\cos(x_9)\cos(x_7)}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix}, \quad (4.38)$$

En este capítulo se describe la estrategia de control aplicada a la plataforma experimental previamente mostrada en el Capítulo 2. Para controlar el UAV se procede como en [4], donde el sistema es dividido en cuatro subsistemas y una linealización por retroalimentación en el control es aplicada en la dinámica z que representa matemáticamente el comportamiento en altura del minihelicóptero. Para lograr los movimientos del UAV, en las diferentes direcciones, se requiere considerar el comportamiento de las velocidades en las propelas mostrado en la Figura B.10.

4.2. Modelo matemático robot todoterreno

La arquitectura Shrimp Rover ha sido estudiada ampliamente y en estado del arte actualmente, se encuentran trabajos donde se ha modelado, en este apartado se muestra un modelo matemático de dicho Robot. El rover móvil es impulsado por seis ruedas que se conectan al cuerpo principal a través de un conjunto de eslabones y articulaciones (ver Figura 4.4). El cuerpo de suspensión principal consta de dos mecanismos de cuatro barras paralelos entre si, ubicados en los laterales del robot, y la horquilla delantera (también un mecanismo de cuatro barras) se emplea para guiar al robot. El ULV es capaz de atravesar terrenos irregulares ajustando sus articulaciones a la topología del lugar, el único contacto con el terreno es en las superficies de las ruedas.



Figura 4.4: Robot con arquitectura shrimp rover.

El movimiento del centro de masa del robot móvil es determinado en términos de los movimientos de las ruedas los cuales son conocidos. Lo anterior para expresar la orientación y posición de puntos deseados como centros de las ruedas, puntos de contacto y articulaciones. Las coordenadas son definidas empleando el método de Denavit-Hartenberg (D-H). Entonces, la posición relativa del movimiento de cada marco de referencia, con respecto al centro de masa del móvil es derivado de la evaluación de las matrices de transformación entre el marco de referencia del robot y cada marco de movimiento de las ruedas.

Para obtener la cinemática de avance del robot, el Jacobiano de las matrices de cada rueda se evalúa de forma simbólica. Los marcos de coordenadas del móvil se ilustran en la Figura 4.5. Todos de los marcos de coordenadas de las ruedas son el sistema de coordenadas de la derecha mostrado a la derecha de dicha imagen. El subíndice para los marcos de coordenadas de locomoción (Figura 4.5 a) es el siguiente: R se refiere al marco de referencia móvil definido en el centro de masa del rover, B es el marco de los mecanismos de cuatro barras paralelos, ubicados lateralmente en el robot, S_i se refiere al marco de las rueda dirección con ($i = 3, 4$), W_i denota el marco ubicado en cada eje de las ruedas con ($i = 1, 2, \dots, 6$). Con el fin de obtener el movimiento de las ruedas, los marcos de referencia de contacto C_i y los marcos de referencia de movimiento instantáneo $M_i(t)$ son definidos para cada rueda (Figura 4.5 b), los C_i definen la los puntos de contacto de las ruedas sobre el terreno, los $M_i(t - \Delta t)$ y los $M_i(t)$ permiten contar los giros de las ruedas ($R_W * \gamma_i$) y los

desplazamientos de rotación δ_i del tiempo $(t - \Delta t)$ al tiempo t , donde Δt , representa un incremento en el tiempo, los A_i representan marcos de referencia auxiliares para las ruedas.

Una matriz de transformación puede ser definida entre dos secuencias seguidas i e $i + 1$, usando los parámetros de D-H a_i , α_i , d_i y θ_i .

$$T_{i,i+1} = Trans_z(d_i)Rot_z(\theta_i)Trans_z(a_i)Rot_x(\alpha_i), \quad (4.39)$$

$$= \begin{bmatrix} c(\theta_i) & -s(\theta_i)c(\alpha_i) & s(\theta_i)s(\alpha_i) & a_i c(\theta_i) \\ s(\theta_i) & c(\theta_i)c(\alpha_i) & -c(\theta_i)s(\alpha_i) & a_i s(\theta_i) \\ 0 & s(\alpha_{i-1}) & c(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (4.40)$$

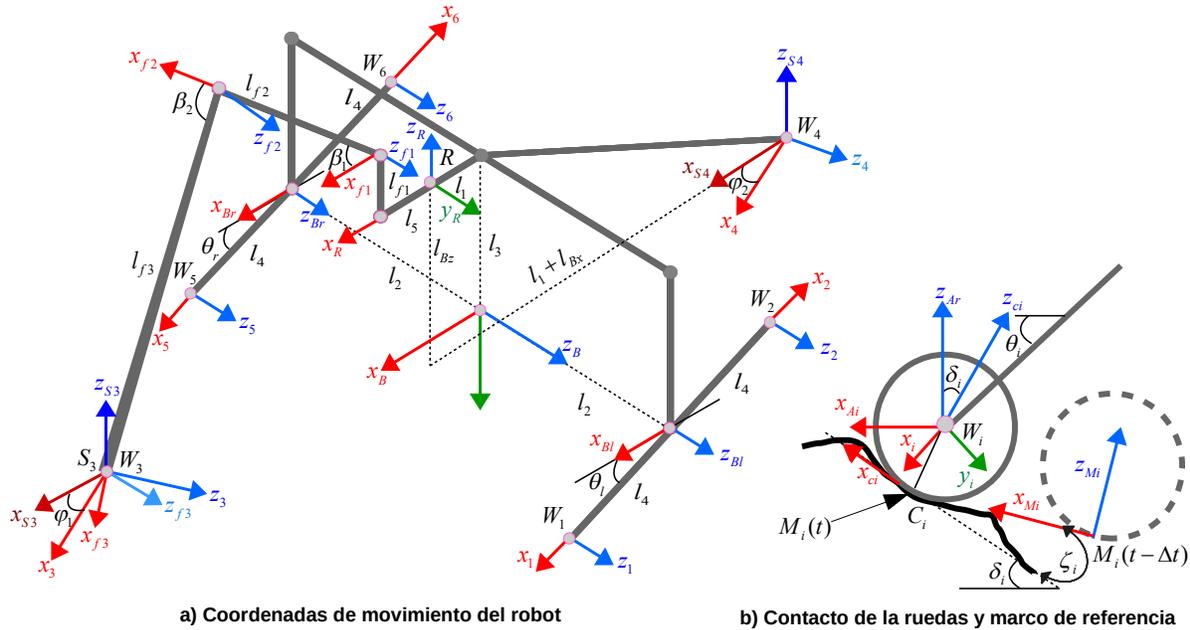


Figura 4.5: Marcos de coordenadas del Shrimp Rover [27].

Los marcos de referencia mostrados la Figura 4.6, corresponden a los del modelo matemático propuesto en [27] cuyo diagrama es mostrado en 4.5, aplicados al robot propuesto en el presente trabajo.

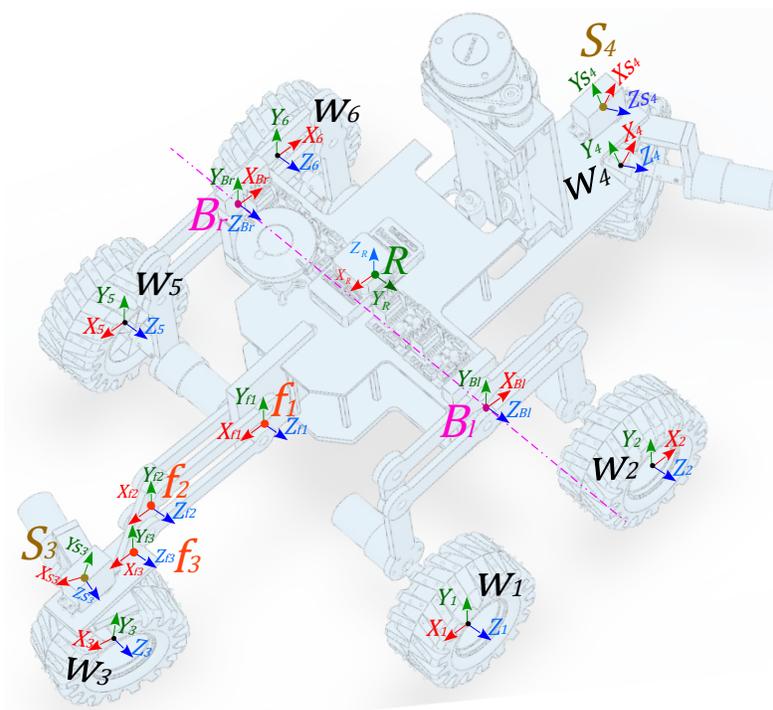


Figura 4.6: Marcos de referencia aplicados al robot todo terreno propuesto.

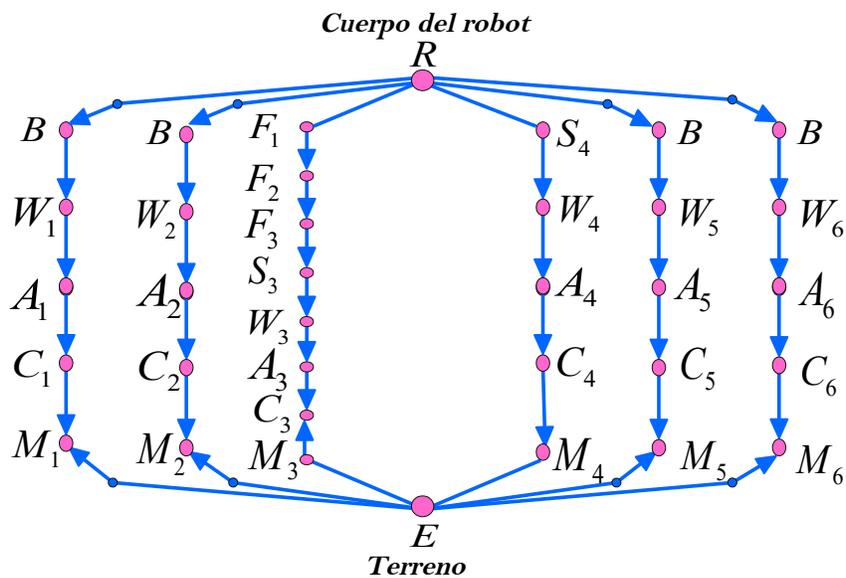


Figura 4.7: Transformaciones para los marcos de referencia del shrimp rover [27].

Los parámetros D-H correspondientes a los marcos de referencia de la Figura 4.5, son mostrados en la Tabla 4.1. Debido a que el Rover es un robot de 6 ruedas, posee 6 transformaciones separadas de coordenadas de cadenas abiertas, que son mostradas en la Figura 4.7. Aplicando la secuencia de transformaciones en las cadenas, las matrices de transformación para cada rueda A_i pueden ser escritas en término del marco de referencia del robot R .

$$T_{R,A_i} = T_{R,B}T_{B,W_i}T_{W_i,A_i}, \quad (i = 1, 2, 5, 6), \quad (4.41)$$

$$T_{R,A_3} = T_{R,F_1}T_{F_1,F_2}T_{F_2,F_3}T_{F_3,S_3}T_{S_3,W_3}T_{W_3,A_i}, \quad (4.42)$$

$$T_{R,A_4} = T_{R,S_4}T_{S_4,W_4}T_{W_4,A_i}, \quad (4.43)$$

Tabla 4.1: Tabla de Parámetros D-H del Rover [27].

Marco de referencia	a_i	α_i	d_i	θ_i
B	$-l_1$	$-\pi/2$	$-l_3$	0
F_1	l_5	$-\pi/2$	l_{f1}	0
F_2	l_{f2}	0	0	$-\beta_1$
F_3	l_{f3}	0	0	β_2
S_3	0	$\pi/2$	0	$\beta_1 - \beta_2$
S_4	$-l_1 - l_{Bx}$	0	$-l_{Bz}$	0
W_1	l_4	0	l_2	θ_1
W_2	l_4	0	l_2	$\pi + \theta_1$
W_3	0	$-\pi/2$	0	ϕ_1
W_4	0	$-\pi/2$	0	ϕ_2
W_5	l_4	0	$-l_2$	θ_r
W_6	l_4	0	$-l_2$	$\pi + \theta_r$
A_i	0	$\pi/2$	0	ρ_i
$\rho_1 = -\theta_1, \rho_2 = \pi - \theta_1, \rho_3 = 0, \rho_4 = 0, \rho_5 = -\theta_r, \rho_6 = \pi - \theta_r$				

Las matrices de transformación para los marcos de referencia de las ruedas A_i hacia los marcos de contacto C_i denotadas como T_{A_i,C_i} pueden ser dadas por:

$$T_{A_i,C_i} = Rot_y(-\delta_i)Trans_z(-R_w), \quad (4.44)$$

$$= \begin{bmatrix} c(\delta_i) & 0 & -s(\delta_i) & R_w s(\delta_i) \\ 0 & 1 & 0 & 0 \\ s(\delta_i) & 0 & c(\delta_i) & -R_w c(\delta_i) \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (4.45)$$

Las transformaciones para el movimiento de las ruedas de $M_i(t - \Delta t) \equiv \hat{M}_i$ a $M(t) \equiv C_i$, están dadas por:

$$T_{\hat{M}_i, C_i} = Trans_x(R_w \gamma_i) Rot_z(\zeta_i), \quad (4.46)$$

$$= \begin{bmatrix} c(\zeta_i) & -s(\zeta_i) & 0 & R_w \zeta_i \\ s(\zeta_i) & c(\zeta_i) & 0 & 0 \\ 0_i & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (4.47)$$

La transformación para el movimiento del marco de referencia de una rueda en el tiempo $(t - \delta t)$ respecto al marco de referencia del Rover R puede ser escrita como:

$$T_{\hat{M}_i, R} = T_{\hat{M}_i, C_i} T_{C_i, W_i} T_{W_i, R}, \quad (4.48)$$

Para describir la transformación del marco de referencia del robot en el tiempo $(t - \delta t)$ al marco de referencia en el tiempo t , se define la matriz $T_{\hat{R}, R} = T_{\hat{R}, \hat{M}_i} T_{\hat{M}_i, R}$. Ya que $T_{\hat{R}, \hat{M}_i}$ es una matriz constante debido a su independencia respecto al tiempo, la derivada de $T_{\hat{R}, R}$ es:

$$\dot{T}_{\hat{R}, R} = T_{\hat{R}, \hat{M}_i} \cdot \dot{T}_{\hat{M}_i, R}, \quad (4.49)$$

La derivada de la transformación dada por $\dot{T}_{\hat{R}, R}$ define el movimiento del marco de referencia del robot R respecto a los marcos de referencia de movimiento de las ruedas \hat{M}_i , y puede ser calculada como:

$$\dot{T}_{\hat{M}_i, R} = \frac{\partial T_{\hat{M}_i, R}}{\partial \zeta_i} \dot{\zeta}_i + \frac{\partial T_{\hat{M}_i, R}}{\partial \gamma_i} \dot{\gamma}_i + \frac{\partial T_{\hat{M}_i, R}}{\partial \delta_i} \dot{\delta}_i + \frac{\partial T_{\hat{M}_i, R}}{\partial \psi_i} \dot{\psi}_i + \frac{\partial T_{\hat{M}_i, R}}{\partial \theta_i} \dot{\theta}_i, \quad (4.50)$$

Empleando las velocidades de posición y orientación, la derivada de las coordenadas del marco de referencia del robot $\dot{T}_{\hat{R}, R}$, puede ser hallada de forma tal que se tenga una forma general del

movimiento:

$$\dot{T}_{\hat{R},R} = \begin{bmatrix} 0 & -\dot{\phi} & \dot{p} & \dot{x} \\ \dot{\phi} & 0 & -\dot{r} & \dot{y} \\ -\dot{p} & \dot{r} & 0 & \dot{z} \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad (4.51)$$

Los ángulos de rotación respecto al marco de referencia del robot son: ϕ respecto al eje z , p respecto al eje y y r respecto al eje x . Sustituyendo las ecuaciones (4.50) y (4.51) en la ecuación (4.49), y factorizando los componentes de velocidad de los ángulos de unión en el lado derecho de la ecuación (4.49), e igualando los elementos de ambos lados, la matriz Jacobiana para cada rueda está dada por:

$$\begin{bmatrix} \dot{x} & \dot{y} & \dot{z} & \dot{\phi} & \dot{p} & \dot{r} \end{bmatrix}^T = J_i \dot{q}_i, \quad i = 1, \dots, 6, \quad (4.52)$$

Por lo anterior, para la rueda 1 se tiene:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{p} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} l_4 s(\theta_1) & R_W c(\delta_1) & l_2 c(\delta_1) & l_3 + l_4 s(\theta_1) \\ 0 & 0 & l_1 c(\delta_1) + l_3 s(\delta_1) - l_4 c(\theta_1 + \delta_1) & 0 \\ l_4 c(\theta_1) & R_W s(\delta_1) & l_2 s(\delta_1) & -l_1 + l_4 c(\theta_1) \\ 0 & 0 & c(\delta_1) & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -s(\delta_1) & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\gamma}_1 \\ \dot{\zeta}_1 \\ \dot{\delta}_1 \end{bmatrix}, \quad (4.53)$$

Combinando cada matriz Jacobiana de las ruedas, se puede deducir una ecuación que permite calcular el movimiento del robot [27]:

$$J = \begin{bmatrix} J_1 \\ J_2 \\ J_3 \\ J_4 \\ J_5 \\ J_6 \end{bmatrix}, \quad (4.54)$$

Donde J_i representa la matriz Jacobiana de la rueda i . El vector de velocidades de las ruedas se denota como $\dot{q} = [\dot{q}_1, \dot{q}_2, \dot{q}_3, \dot{q}_4, \dot{q}_5, \dot{q}_6]^T$. La matriz Jacobiana total se utiliza para relacionar las velocidades de las ruedas con las velocidades del marco de referencia del robot:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{p} \\ \dot{r} \end{bmatrix} = J \cdot \dot{q}, \quad (4.55)$$

Los ángulos de rotación respecto al marco de referencia del robot son: ϕ respecto al eje z , p respecto al eje y y r respecto al eje x . Sustituyendo las ecuaciones (4.50) y (4.51) en la ecuación (4.49), y factorizando los componentes de velocidad de los ángulos de unión en el lado derecho de la ecuación (4.49), e igualando los elementos de ambos lados, la matriz Jacobiana para cada rueda está dada por:

$$\begin{bmatrix} \dot{x} & \dot{y} & \dot{z} & \dot{\phi} & \dot{p} & \dot{r} \end{bmatrix}^T = J_i \dot{q}_i, \quad i = 1, \dots, 6, \quad (4.56)$$

Por lo anterior, para la rueda 1 se tiene la siguiente matriz jacobiana:

$$J_1 = \begin{bmatrix} l_4 s(\theta_1) & R_W c(\delta_1) & l_2 c(\delta_1) & l_3 + l_4 s(\theta_1) \\ 0 & 0 & l_1 c(\delta_1) + l_3 s(\delta_1) - l_4 c(\theta_1 + \delta_1) & 0 \\ l_4 c(\theta_1) & R_W s(\delta_1) & l_2 s(\delta_1) & -l_1 + l_4 c(\theta_1) \\ 0 & 0 & c(\delta_1) & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -s(\delta_1) & 0 \end{bmatrix}, \quad (4.57)$$

donde: θ_i : El ángulo de rotación de la rueda i en relación con su marco de referencia local. Es el ángulo de giro del motor asociado con la rueda i .

δ_i : El ángulo de dirección de la rueda i con respecto a su marco de referencia local. Este ángulo define la orientación de la rueda en el plano horizontal.

γ_i : El ángulo de desplazamiento de la rueda i con respecto a su marco de referencia local. Este ángulo se refiere a la inclinación de la rueda en el plano vertical.

ζ_i : El ángulo de rotación del marco de contacto C_i con respecto al marco de referencia de la rueda A_i . R_W : El radio de la rueda.

l_1, l_2, l_3, l_4 : Son parámetros geométricos del sistema de ruedas y su conexión al cuerpo del robot.

$$J_2 = \begin{bmatrix} l_4 s(\theta_2) & -R_W c(\delta_2) & l_2 c(\delta_2) & l_3 + l_4 s(\theta_2) \\ 0 & 0 & l_1 c(\delta_2) + l_3 s(\delta_2) + l_4 c(\theta_2 + \delta_2) & 0 \\ l_4 c(\theta_2) & -R_W s(\delta_2) & l_2 s(\delta_2) & -l_1 + l_4 c(\theta_2) \\ 0 & 0 & c(\delta_2) & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -s(\delta_2) & 0 \end{bmatrix}, \quad (4.58)$$

$$J_3 = \begin{bmatrix} l_4 s(\theta_3) & R_W c(\delta_3) & -l_2 c(\delta_3) & l_3 + l_4 s(\theta_3) \\ 0 & 0 & l_1 c(\delta_3) - l_3 s(\delta_3) - l_4 c(\theta_3 + \delta_3) & 0 \\ l_4 c(\theta_3) & R_W s(\delta_3) & l_2 s(\delta_3) & -l_1 + l_4 c(\theta_3) \\ 0 & 0 & c(\delta_3) & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -s(\delta_3) & 0 \end{bmatrix}, \quad (4.59)$$

$$J_4 = \begin{bmatrix} l_4 s(\theta_4) & -R_W c(\delta_4) & -l_2 c(\delta_4) & l_3 + l_4 s(\theta_4) \\ 0 & 0 & l_1 c(\delta_4) - l_3 s(\delta_4) + l_4 c(\theta_4 + \delta_4) & 0 \\ l_4 c(\theta_4) & R_W s(\delta_4) & l_2 s(\delta_4) & -l_1 + l_4 c(\theta_4) \\ 0 & 0 & c(\delta_4) & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -s(\delta_4) & 0 \end{bmatrix}, \quad (4.60)$$

$$J_5 = \begin{bmatrix} l_4 s(\theta_5) & R_W c(\delta_5) & -l_2 c(\delta_5) & l_3 + l_4 s(\theta_5) \\ 0 & 0 & -l_1 c(\delta_5) - l_3 s(\delta_5) - l_4 c(\theta_5 + \delta_5) & 0 \\ l_4 c(\theta_5) & R_W s(\delta_5) & l_2 s(\delta_5) & -l_1 + l_4 c(\theta_5) \\ 0 & 0 & c(\delta_5) & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -s(\delta_5) & 0 \end{bmatrix}, \quad (4.61)$$

$$J_6 = \begin{bmatrix} l_4 s(\theta_6) & -R_W c(\delta_6) & l_2 c(\delta_6) & l_3 + l_4 s(\theta_6) \\ 0 & 0 & l_1 c(\delta_6) - l_3 s(\delta_6) + l_4 c(\theta_6 + \delta_6) & 0 \\ l_4 c(\theta_6) & R_W s(\delta_6) & l_2 s(\delta_6) & -l_1 + l_4 c(\theta_6) \\ 0 & 0 & c(\delta_6) & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -s(\delta_6) & 0 \end{bmatrix}, \quad (4.62)$$

La combinación de las matrices Jacobianas para cada rueda se puede hacer sumando todas las matrices juntas. Esto se puede escribir de la siguiente manera:

$$J = \begin{bmatrix} J_1 \\ J_2 \\ J_3 \\ J_4 \\ J_5 \\ J_6 \end{bmatrix}, \quad (4.63)$$

donde J es la matriz Jacobiana total para el robot, y J_1 a J_6 son las matrices Jacobianas para las ruedas individuales.

Cada fila de la matriz J representa una de las velocidades del robot (\dot{x} , \dot{y} , \dot{z} , $\dot{\phi}$, \dot{p} y \dot{r}), y cada columna representa una de las velocidades de los motores ($\dot{\theta}_1$, $\dot{\gamma}_1$, $\dot{\zeta}_1$, $\dot{\delta}_1$, $\dot{\theta}_2$, $\dot{\gamma}_2$, $\dot{\zeta}_2$, $\dot{\delta}_2$, y así sucesivamente).

Este problema requiere utilizar la matriz Jacobiana correspondiente para cada rueda, y luego calcular la pseudoinversa de Moore-Penrose.

Dado que no se puede calcular directamente la inversa de una matriz no cuadrada, se debe utilizar la pseudoinversa de Moore-Penrose (J^+) en lugar de la inversa (J^{-1}). La ecuación sería:

$$\begin{bmatrix} \dot{\theta}_i \\ \dot{\gamma}_i \\ \dot{\zeta}_i \\ \dot{\delta}_i \end{bmatrix} = J_i^+ \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{p} \\ \dot{r} \end{bmatrix}, \quad (4.64)$$

Aquí, J_i^+ es la pseudoinversa de la matriz Jacobiana J_i para la rueda i -ésima e i varía de 1 a 6 para cada rueda del robot. De esta manera, se puede obtener las velocidades de los motores necesarias para cada rueda para lograr la velocidad deseada del robot.

La pseudoinversa de Moore-Penrose de la matriz Jacobiana J se denota como J^+ y se puede calcular utilizando la descomposición en valores singulares (SVD) de la matriz J .

Si $J = U\Sigma V^T$ es la descomposición en valores singulares de J , entonces la pseudoinversa de Moore-Penrose se calcula como:

$$J^+ = V\Sigma^+U^T, \quad (4.65)$$

Donde Σ^+ es la matriz diagonal de los inversos de los valores singulares no nulos de Σ . J^+ : Es

la pseudoinversa de la matriz Jacobiana. Se utiliza para calcular las velocidades de los motores necesarias para lograr ciertas velocidades del robot.

V : Es una matriz ortogonal que proviene de la descomposición en valores singulares de la matriz Jacobiana J . Las columnas de V son los vectores singulares derechos de J .

Σ^+ : Es la pseudoinversa de la matriz diagonal Σ que contiene los valores singulares de J . La matriz Σ^+ se obtiene tomando la matriz Σ , invirtiendo todos los valores singulares no nulos y transponiendo el resultado.

U^T : Es la matriz transpuesta de U , que es una matriz ortogonal que proviene de la descomposición en valores singulares de la matriz Jacobiana J . Las columnas de U son los vectores singulares izquierdos de J .

4.3. Conclusión capítulo 4

En este capítulo, se aborda el Modelado Matemático de los Robots Móviles, una sección fundamental que subraya la importancia de los fundamentos teóricos en el diseño y operación de estos sistemas. Los modelos matemáticos presentados aquí no solo están respaldados y bien documentados en la bibliografía especializada, sino que también han sido ampliamente utilizados y probados en la práctica para cada tipo de robot, asegurando su validez y eficacia. El Modelo Matemático del Vehículo Aéreo se detalla en la primera parte del capítulo, proporcionando una comprensión profunda de la dinámica y el control de estos vehículos. Este modelo es esencial para la programación y operación efectiva de vehículos aéreos en diversas aplicaciones, especialmente en tareas que requieren precisión y estabilidad, como la monitorización ambiental y la recolección de datos.

A continuación, el Modelo Matemático del Robot Todoterreno se presenta, destacando las peculiaridades y desafíos específicos asociados con el movimiento y la navegación en terrenos irregulares o desafiantes. Este modelo es crucial para garantizar que el robot todoterreno pueda operar eficientemente en diferentes tipos de entornos, adaptándose a variaciones en el terreno mientras mantiene su funcionalidad y eficacia.

Estos modelos matemáticos no solo forman la base para el desarrollo y la implementación de las plataformas robóticas descritas en capítulos anteriores, sino que también son fundamentales para la innovación en el control y la automatización de estos sistemas. Al aplicar estos modelos, se pueden optimizar las operaciones de los robots, mejorando su precisión, eficiencia y adaptabilidad, lo que resulta esencial para el éxito de cualquier aplicación práctica en el campo de la robótica móvil. En conjunto, este capítulo resalta la importancia de un sólido fundamento

matemático y teórico en el desarrollo de soluciones robóticas avanzadas, asegurando que los diseños propuestos no solo sean innovadores, sino también prácticos y aplicables en entornos reales.

Capítulo 5

Simulaciones y resultados experimentales

En el presente capítulo son mostrados las simulaciones y los resultados experimentales realizados para cada uno de los robots pertenecientes al sistema de monitoreo ambiental, estos resultados han sido obtenidos al realizar pruebas con la interfaz gráfica creada para el sistema SCADA previamente descrito en capítulos anteriores.

Primero son mostradas dos simulaciones correspondientes al seguimiento de trayectoria de cada robot, seguida de una simulación del control PID subóptimo autosintonizable, aplicado a un actuador del rover. A continuación son mostrados los resultados experimentales, iniciando con el PID autosintonizable aplicado a un motor del ULV, después resultados del sistema de visión y de monitoreo de calidad de aire, del sistema de recolección de muestras de suelos, instrumentación con sensores ópticos para mejora de ubicación del UAV, y por último los resultados experimentales en el seguimiento de trayectoria del robot aéreo y del robot terrestre respectivamente.

5.1. Simulación seguimiento de trayectoria UAV

Para validar el modelo del UAV, se llevaron a cabo simulaciones del mismo. A continuación, se detalla cómo se validó el modelo matemático presentado en [4]. Para simularlo, se utilizó el programa Simulink® de MATLAB® 2016, operando bajo la filosofía de la cadena de integradores. Los bloques de simulación para el control y el modelo del robot se discuten a continuación:

En la Figura 5.1, se muestra el bloque del modelo matemático simulado del cuadricóptero en Simulink®. La integración de cada variable se realiza en los bloques denominados integradores, como se puede observar en el cuadro azul. Este modelo incluye las ecuaciones mostradas en el apartado correspondiente al modelo del robot aéreo.

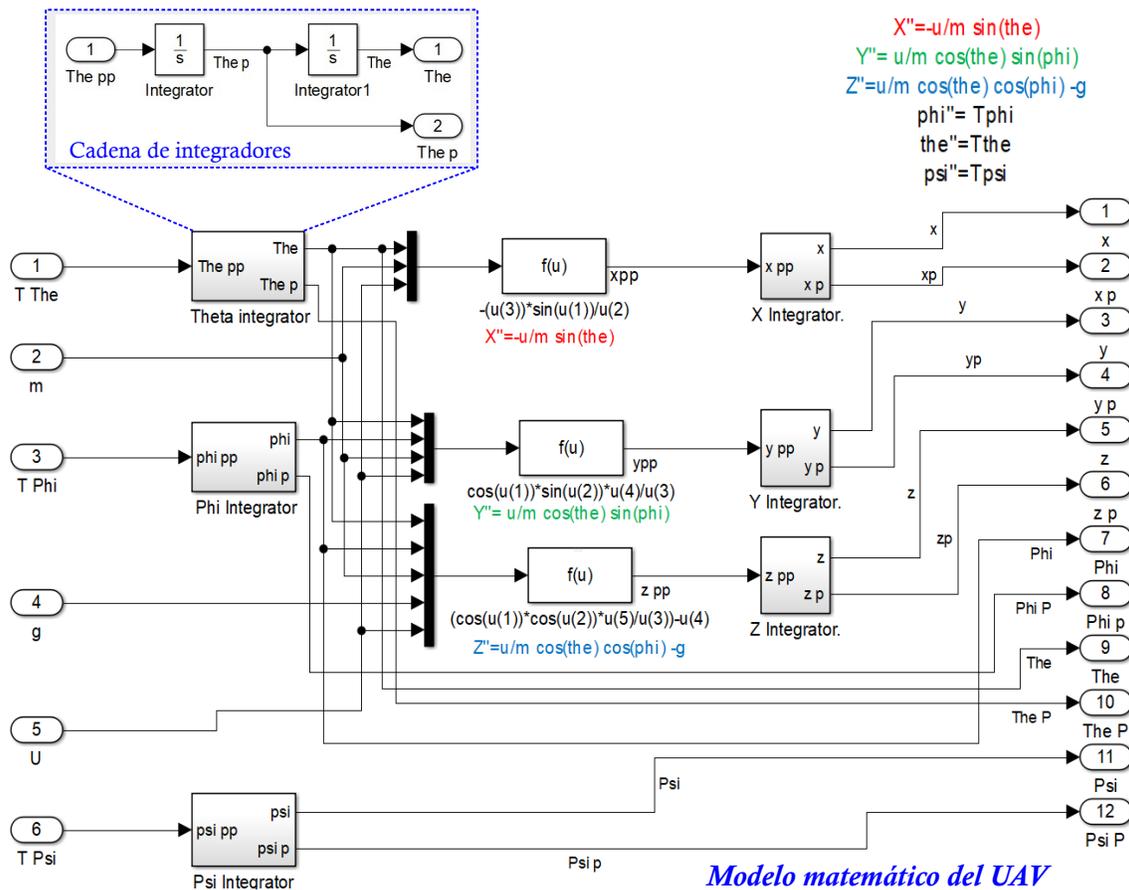


Figura 5.1: Cuadro de bloques del modelo matemático de UAV simulado en Simulink®.

En la Figura 5.2, se exhibe la simulación completa del modelo compactado dentro de un bloque etiquetado como "Modelo de Cuadricóptero". Dicho bloque contiene los bloques de subsistemas representados en la Figura 5.1. El modelo está resaltado dentro de un cuadro azul punteado. El bloque que comprende el controlador está marcado en verde, y dentro de él, se implementa el algoritmo del controlador integral óptimo. Las ganancias del controlador, obtenidas de fuera de línea, están almacenadas en los vectores resaltados en rosa. Por último, las referencias para cada subsistema ($x - \theta$, $y - \phi$, ψ y z) se introducen en los vectores marcados dentro del cuadro rojo. Esta simulación, creada en Simulink®, sirve como la validación del modelo matemático del UAV.

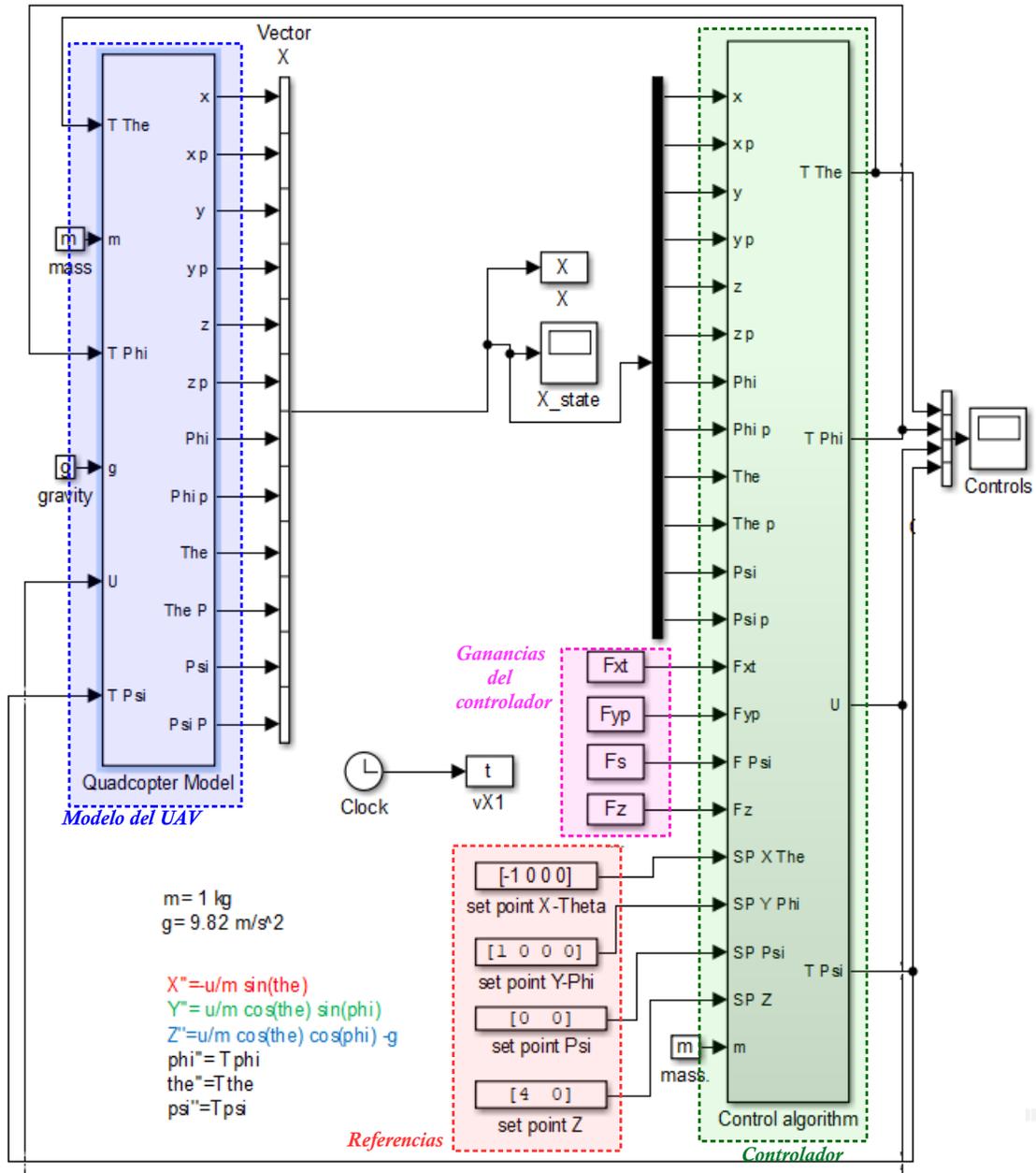


Figura 5.2: Bloque de simulación en Simulink® del modelo matemático del cuadricóptero más el bloque del controlador.

Al configurar una referencia de $x_{\text{ref}} = -1$, $y_{\text{ref}} = 1$, y $z_{\text{ref}} = 4$, y al referenciar el resto de las variables a cero, se simuló el modelo de UAV, resultando en los gráficos que se presentan

en la Figura 5.3. Se puede apreciar que la dirección de los ángulos es coherente con el modelo matemático. Cuando el ángulo de inclinación θ es positivo, x disminuye, y cuando el ángulo ϕ es positivo, y aumenta. Esto demuestra que la definición de los ángulos de rotación es apropiada.

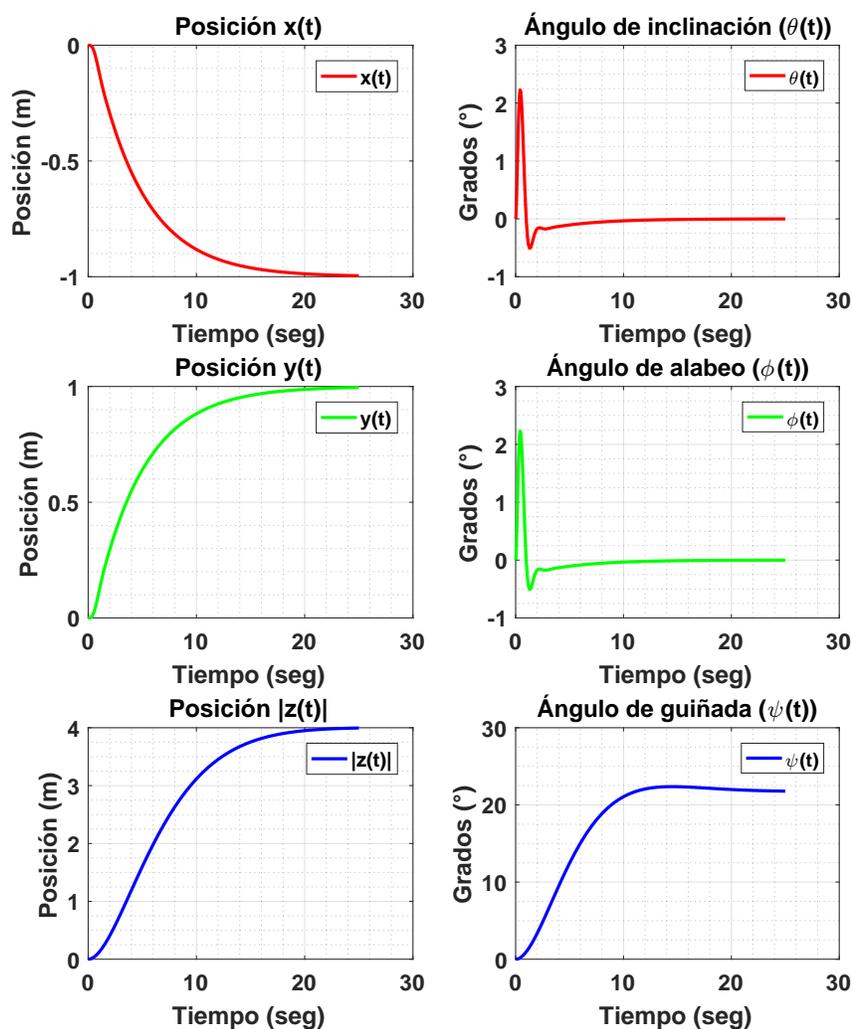


Figura 5.3: Posiciones y ángulos calculados cuando el modelo matemático es simulado en Simulink®.

En la Figura 5.4, se pueden observar las vistas superior (Fig. 5.4-a) y laterales de los planos XZ (Fig. 5.4-b), YZ (Fig. 5.4-c), junto con una vista isométrica (Fig. 5.4-d) de la trayectoria seguida por el cuadricóptero. El robo alcanza con éxito las referencias asignadas, de acuerdo con los ejes de referencia definidos mostrados en la sub figura Fig. 5.4-e.

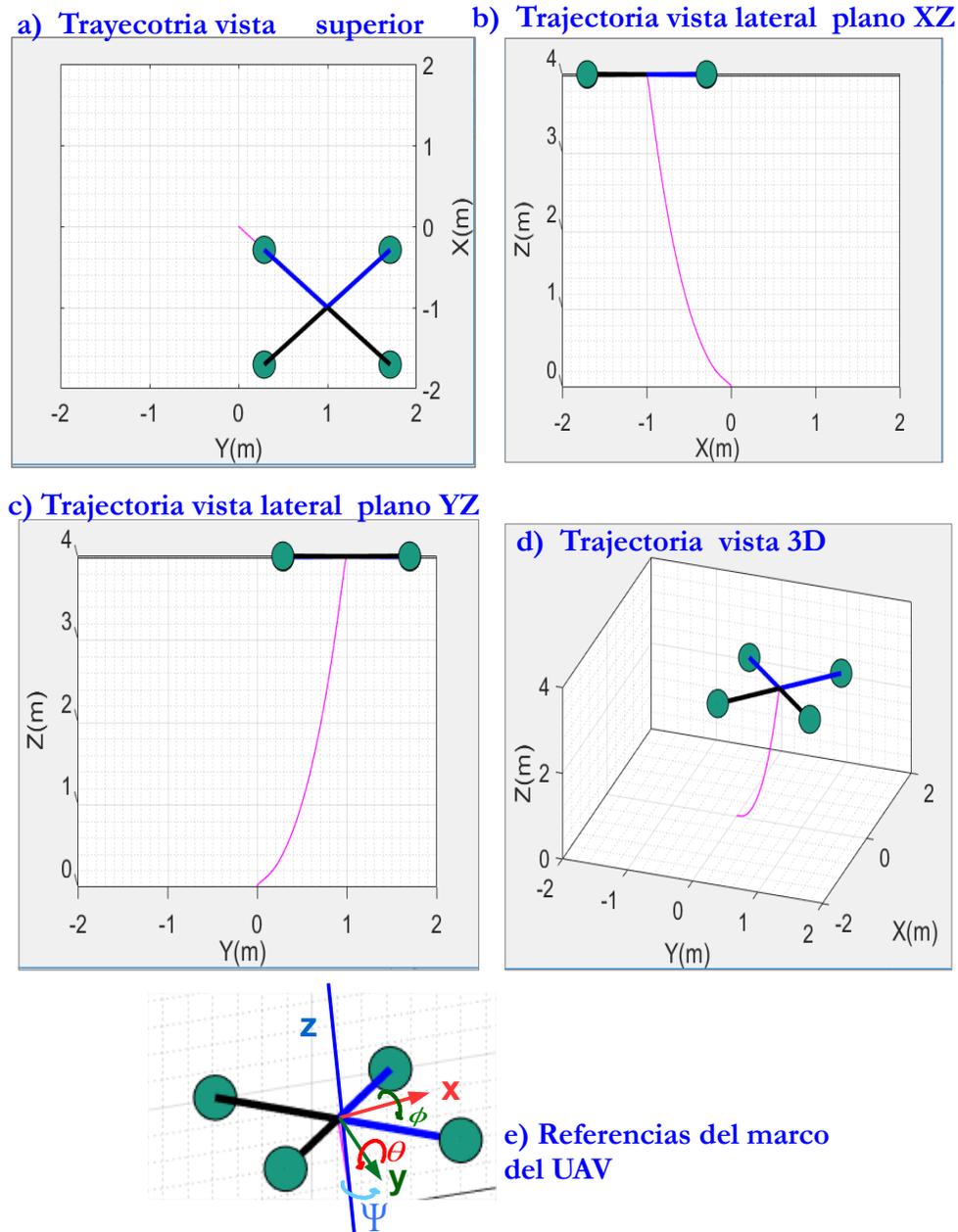


Figura 5.4: Vistas de la trayectoria 3D de la simulación del UAV.

5.2. Simulación seguimiento de trayectoria ULV

El seguimiento de trayectoria del robot todoterreno, es desarrollado en el programa MATLAB® , usando el modelo planteado en este trabajo para este fin, suponiendo un ambiente de trabajo plano y con inclinación nula, se procedió a desarrollar el seguimiento de trayectoria del ULV, buscando que el robot alcance los siguientes puntos: $p_0 = [0, 0]$; $p_1 = [0.25, 0.25]$; $p_2 = [0.25, 0.50]$; $p_3 = [0.75, 0.75]$; un ángulo inicial (guiñada) $\phi(t) = 0$ rad y velocidades lineal y angular nulas.

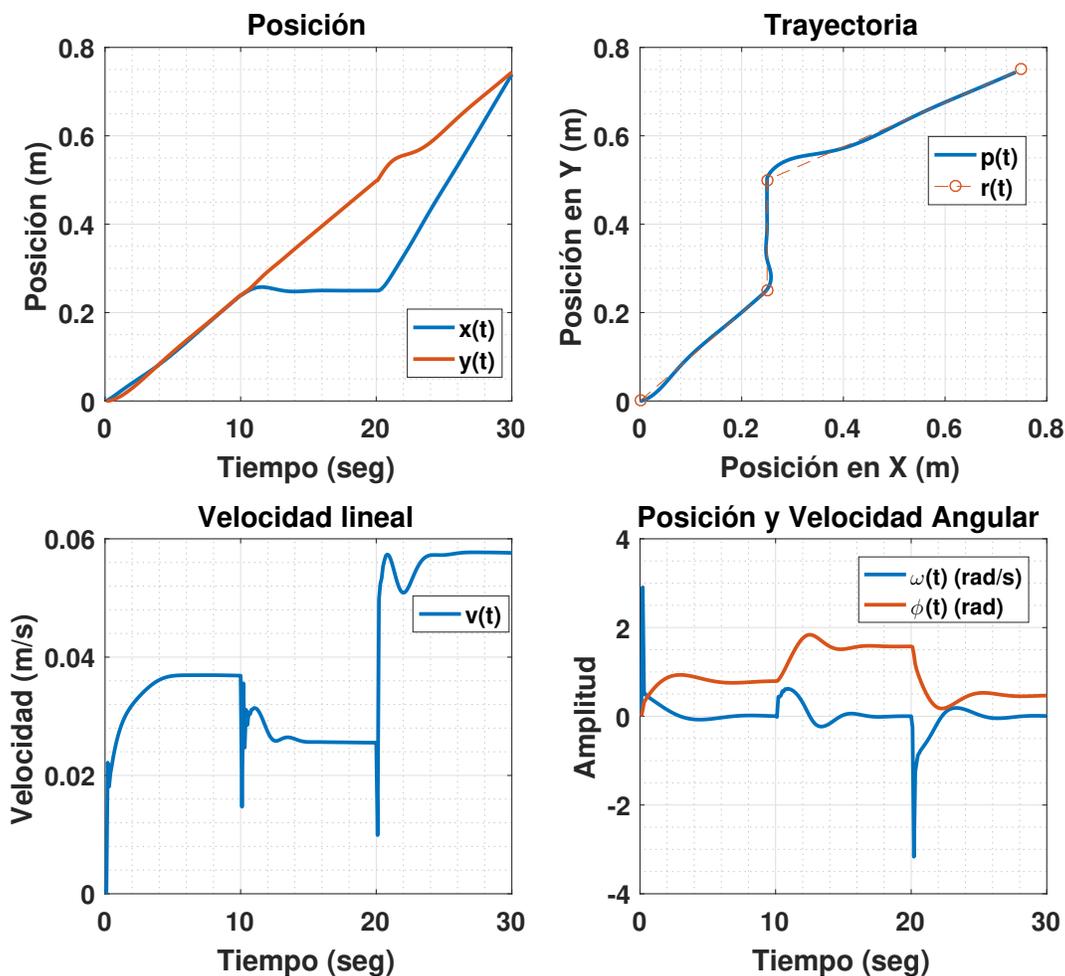


Figura 5.5: Resultados de simulación seguimiento de trayectoria rover.

En las gráficas de la Figura 5.5, se observa el comportamiento de la posición, la orientación y las velocidades de éstas variables, así como, la trayectoria seguida por el rover, en la trayectoria se pueden notar los puntos de referencia en círculos naranjas.

Para hacer el seguimiento de la trayectoria se ha usado un controlador PID con las ganancias: $K_p = [1.2, 0.7]$; $K_i = [0.07, 0.03]$; $K_d = [0.5, 0.3]$; donde el primer elemento de cada vector es usado en cada ganancia para el control en posición y el segundo para el control en orientación respectivamente.

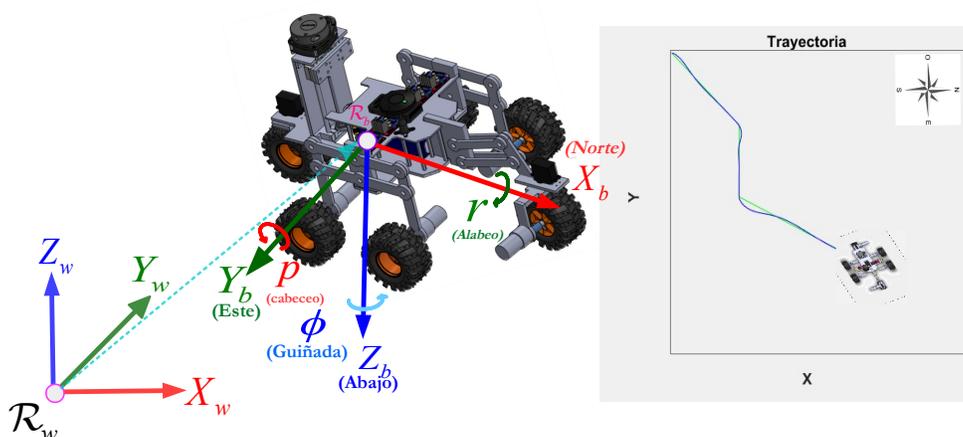


Figura 5.6: Marcos de referencia para simulación de seguimiento de trayectoria del ULV.

Los marcos de referencia del robot R_b y del ambiente de trabajo R_w pueden ser vistos en la Figura 5.6, además de la orientación que se usó para hacer la simulación del seguimiento de la trayectoria del robot. Como puede notarse se agregó una figura .png que muestra el robot desde una vista aérea para mejorar la simulación, permitiendo al usuario observar de forma directa la orientación y posición que va tomando el robot mientras consigue llegar a cada uno de los puntos fijados en la referencia de la trayectoria.

5.3. Simulación controlador subóptimo autosintonizable en motor del ULV

Debido a la naturaleza de las tareas que desempeña cada robot móvil de la plataforma propuesta en este trabajo, es requerido que la ley de control a aplicar a éstos, contemple las siguientes

características:

- 1) Subóptimo.
- 2) Autoajustable.
- 3) Robusto.

Simulación Filtro de Kalman

Para buscar cubrir lo anterior se propone el marco de trabajo presentado en el Apéndice A.1, llamado estrategia de control subóptimo adaptable. Donde la idea principal es tomar la ventajas de un controlador PID, el filtro de Kalman y el método del gradiente descendente para crear un controlador subóptimo adaptable y robusto. A continuación son presentados resultados de simulación, para este esquema aplicado aun sistema de primer orden en ecuación de diferencias, en el programa MATLAB®.

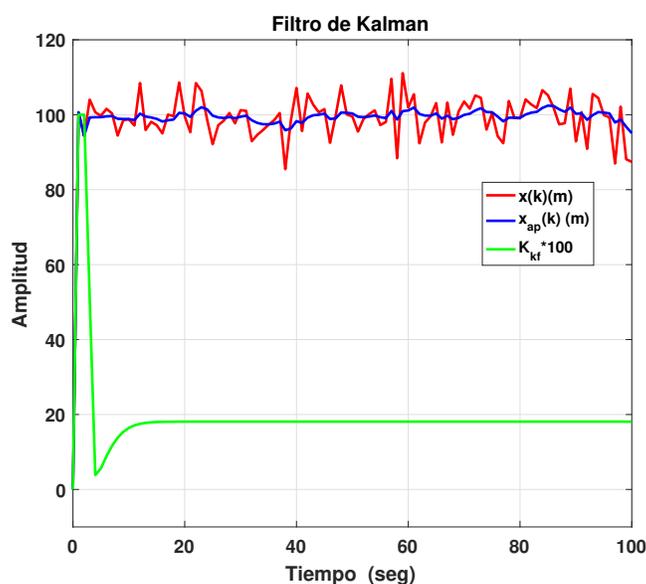


Figura 5.7: Aproximación usando un filtro de Kalman simulado en MATLAB®.

En la gráfica de la Figura 5.7 se puede ver un ejemplo de filtro de Kalman aplicado a una señal original $x(k)$, que contiene un ruido blanco, en color azul puede verse la señal aproximada $x_{ap}(k)$, la cual de acuerdo a la teoría del filtro de Kalman presentada en el Capítulo 2, es una aproximación óptima de $x(k)$, en color verde puede observarse la ganancia de Kalman $K_{kf}(k)$ multiplicada por un factor de 100 (esto con fines de ampliarla para observar su comportamiento).

Como puede verse, la ganancia varía inicialmente (de forma similar a un periodo transitorio) y una vez que el algoritmo logra aproximar la respuesta ésta converge a un valor fijo, lo que implica que en variaciones grandes de la señal a aproximar, puede reiniciarse el filtro y la ganancia se comportará de forma tal que el sistema sea aproximado óptimamente. Esta cualidad es la que garantiza que se puede tener una ganancia con comportamiento óptimo que puede usarse para aproximar las de un controlado PID, sujeto a un índice de desempeño J , de forma recursiva y en línea.

Simulación Gradiente Descendente

A continuación una simulación donde se sintoniza un controlador PID empleando el método del gradiente descendente es mostrada, en ella se ven los errores que se generan para cada sintonización que el algoritmo optimiza, el enfoque empleado en esta simulación es el mostrado en el Capítulo 2, a un PID discretizado como en la ecuación 2.31. Dicho controlador es aplicado a un sistema de primer orden normalizado, se agrega ruido gaussiano para verificar robustez, y cada uno de los $e_i(k)$, que se gráficán en la Figura 5.8, representa el error para cada ajuste usando el método del gradiente descendente, cabe mencionar que para cada ajuste se calcula una única vez el cambio en las ganancias del controlador, es decir se realiza fuera de línea y para cada experimento se parte del ajuste anterior, como condición inicial en las ganancias a modificar (K_1 , K_2 y K_3), y para detener el algoritmo se realiza en función a un criterio establecido por el usuario, mediante un número finito de repeticiones, o al determinar que las nuevas ganancias desestabilizan al sistema.

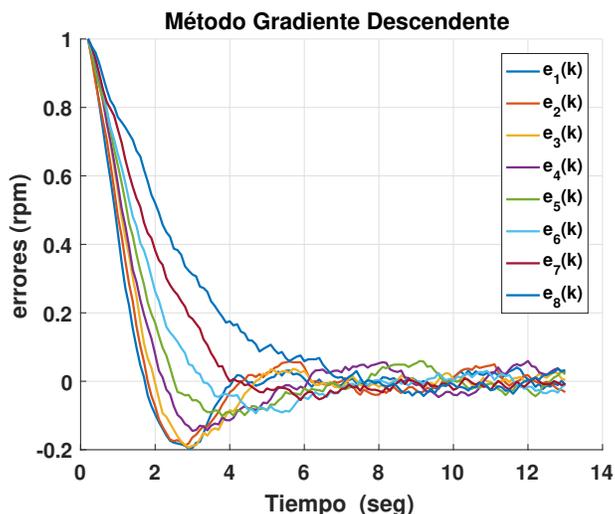


Figura 5.8: Comportamiento del error al emplear el método del gradiente descendente simulado en MATLAB®.

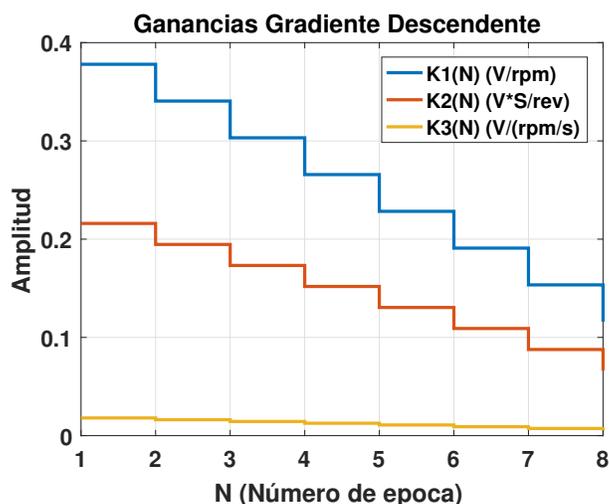


Figura 5.9: Comportamiento de ganancias de PID usando método del gradiente descendente fuera de línea simulado en MATLAB®.

El comportamiento de las ganancias del controlador PID para cada uno de los ajustes realizados, es visto en la gráfica de la Figura 5.9, N representa el número de veces que se ajustó cada ganancia empleando el descenso del gradiente, para este caso se ejecutó 8 veces el algoritmo, hasta encontrar un comportamiento deseado en el error. Es importante mencionar que las ganancias iniciales causan un sobre impulso en el sistema, sin embargo, el algoritmo va disminuyendo dicho sobre impulso al ajustar el valor de las ganancias por cada ciclo.

Simulación del controlador subóptimo adaptable propuesto

Empleando el algoritmo propuesto en el Apéndice A.1, para ajustar un controlador PID, se muestra en la Figura 5.10, la comparación entre un PID sintonizado heurísticamente, con ganancias fijas $K_p = 15$, $K_i = 17$ y $K_d = 0.3$, que genera la trayectoria de color azul, cuando se pide una referencia $r(k) = 30$ con $0s \leq t \leq 20s$, después de los 20 segundos la referencia es fijada en 50 hasta 60 segundos, a partir de este tiempo la referencia se asigna en 20, lo anterior con la finalidad de comprobar la capacidad de adaptación del controlador, se agrega ruido al sistema de con media cero. El controlador subóptimo adaptable genera la tracería mostrada en color rojo, las ganancias iniciales para este controlador son las mismas que se emplean en el PID que genera la trayectoria azul. Como se puede notar, el controlador logra reducir la oscilación y es capaz de trabajar con referencias positivas y negativas.

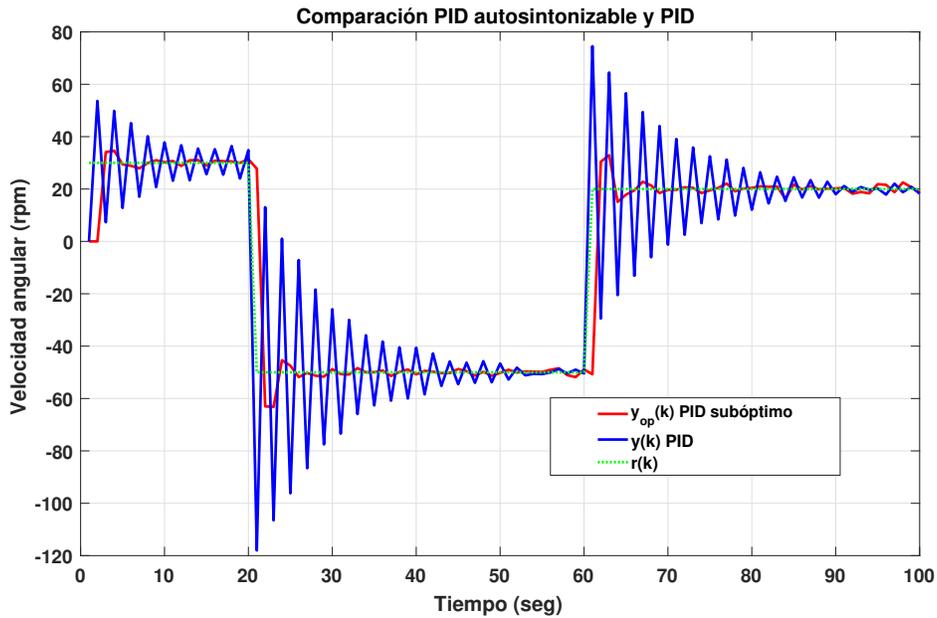


Figura 5.10: Comparación entre controlador PID con controlador PID subóptimo autosintonizable, simulados en MATLAB®.

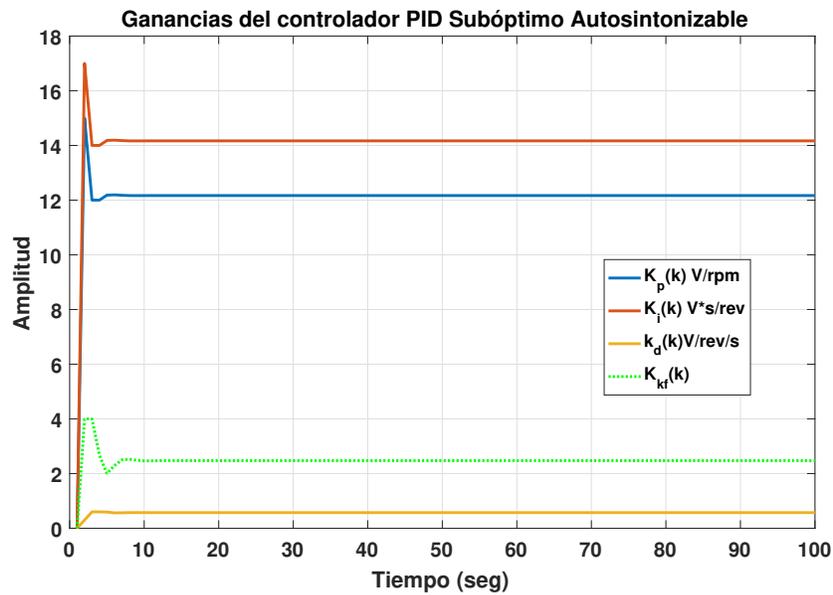


Figura 5.11: Ganancias autoajustables del controlador PID subóptimo adaptable, simulados en MATLAB®.

El comportamiento de las ganancias del controlador adaptable y de las ganancias del filtro de Kalman usada para que el algoritmo funciones es mostrado en la Figura 5.11, como puede verse inician en los valores antes mencionados y a partir de ellos y conforme la ganancia de Kalman se ajusta las otras 3 se adaptan a ella, lo anterior dependiendo de la penalización que el usuario asigne al índice de rendimiento J .

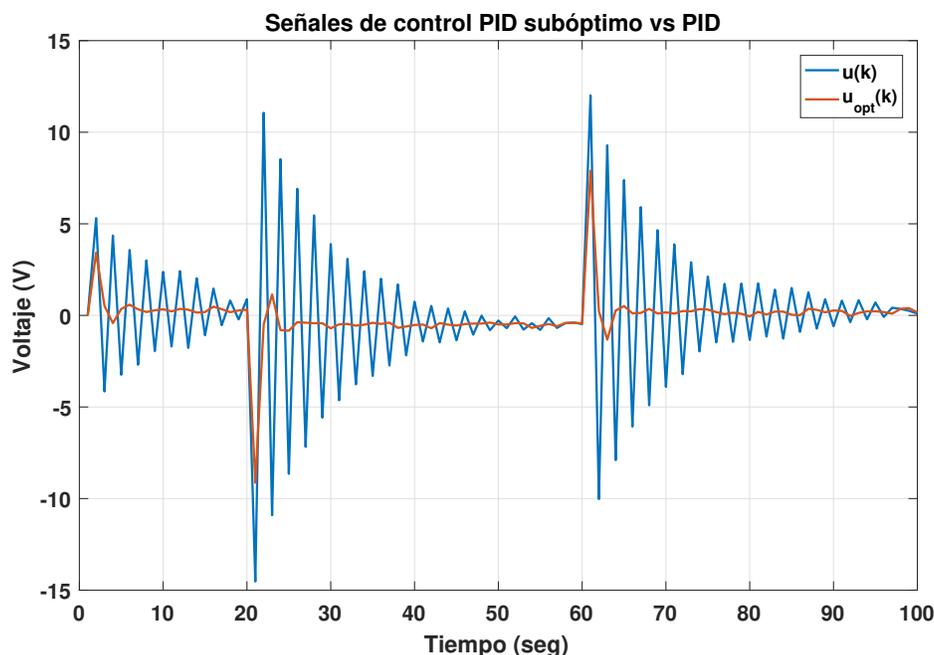


Figura 5.12: Señales de control PID subóptimo autosintonizable contra PID, simulados en MATLAB®.

En la Figura 5.12 se pueden observar las señales de control de los dos PIDs, el subóptimo adaptable y el convencional, calculando la integral de ambas señales, y promediándola para 10 experimentos, en donde en cada uno de ellos se sigue la misma referencia, concluyendo que: la integral promedio de la señal de control del controlador PID no adaptable es:

$$\sum_{i=1}^{10} \left[\sum_{k=1}^N u(k) \Delta T \right] / 10 = 399.8164, \quad (5.1)$$

mientras que la señal de control promedio para $u_{opt}(k)$ es:

$$\sum_{i=1}^{10} \left[\sum_{k=1}^N u_{opt}(k) \Delta T \right] / 10 = 320.2222, \quad (5.2)$$

Con estos datos, la mejora que presenta el controlador PID subóptimo autosintonizable, tiene en cuanto a los niveles de la señal de control con respecto al PID convencional, es de un 19.907%. Lo que demuestra en resultados de simulación que el algoritmo trabaja correctamente con las ventajas esperadas.

5.4. Resultados experimentales controlador subóptimo autosintonizable

Aplicando la estrategia de control mostrada en la Figura(A.5), correspondiente al algoritmo híbrido propuesto en el presente trabajo, a un motor de CD, que cuenta con un encoder incremental instrumentado a un ESP32, como se indica en la Figura(B.6), para medir y controlar su velocidad angular en su armadura ($y(k)$), y comparando el resultado contra un controlador PID discreto, sintonizado por el método de Ziegler-Nichols, buscando un sobre impulso máximo del 10% en la respuesta, se obtienen las curvas en la velocidad de la gráfica mostrada en la Figura(5.13).

La referencia $r(k)$ está definida como 33.5 para $t < 57s$ y 28 para tiempos superiores, permitiendo observar como el PID convencional es capaz de trabajar correctamente en la zona de operación donde fue sintonizado, sin embargo, al modificar dicha referencia puede apreciarse que éste no logra hacer que el sistema converja de la misma manera, causando oscilaciones, esto puede verse en el comportamiento de la velocidad angular controlado con este controlador $y(k)$.

Por otra parte, el controlador PID subóptimo autosintonizable es capaz de observar las oscilaciones y hacer que el sistema converja de mejor forma (trayectoria de la variable $y^*(k)$), las ganancias iniciales de este último controlador son las mismas usadas en el controlador PID: $Kp = 16$, $Ki = 10$ y $Kd = 0.1$.

Esta cualidad de autosintonización es requerida en el robot todoterreno debido a la naturaleza del comportamiento que tendrá cada rueda dependiendo de las irregularidades u obstáculos aplicados a la misma.

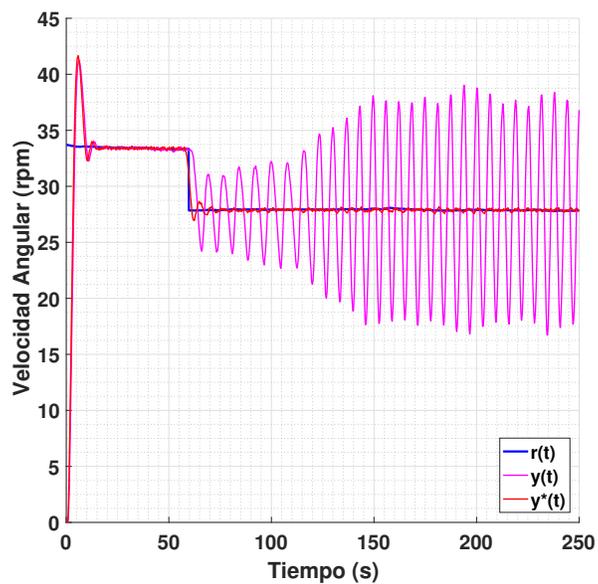


Figura 5.13: Comparación respuesta PID contra control subóptimo autosintonizable.

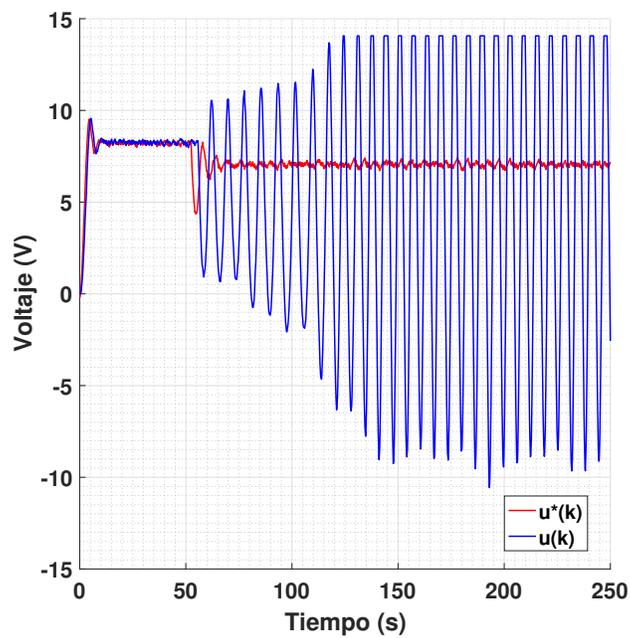


Figura 5.14: Comparación señal de control PID contra control subóptimo autosintonizable.

El comportamiento de las ganancias del controlador PID Subóptimo Autosintonizable son mostradas en la Figura (5.15), además, puede notarse la ganancia del filtro de Kalman que permite que las otras se ajusten, de forma iterativa y óptima.

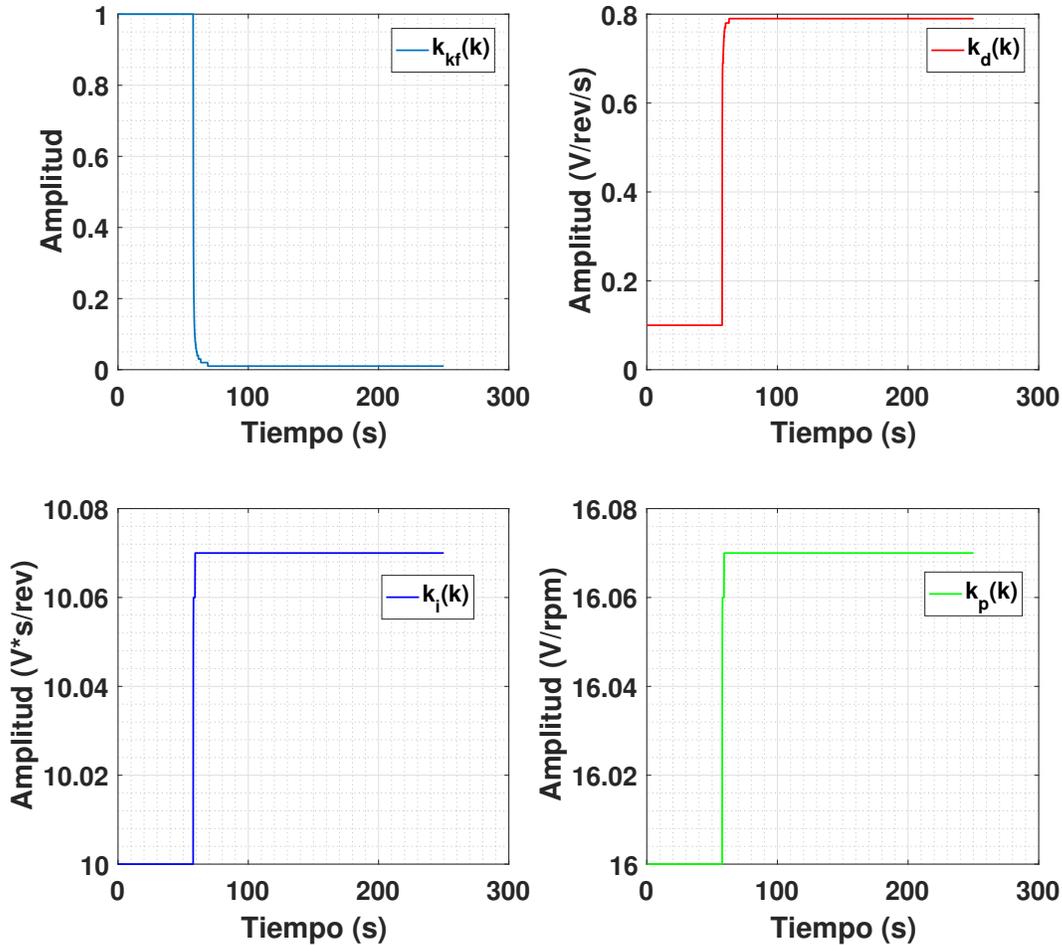


Figura 5.15: Comportamiento de ganancias de controlador subóptimo autosintonizable.

Puede verse una aproximación de las ganancias en su etapa transitoria, en la Figura 5.16, esto con la finalidad de observar mejor el comportamiento que estas ganancias tienen cuando el algoritmo las ajusta en función de la aproximación que realiza el filtro de Kalman.

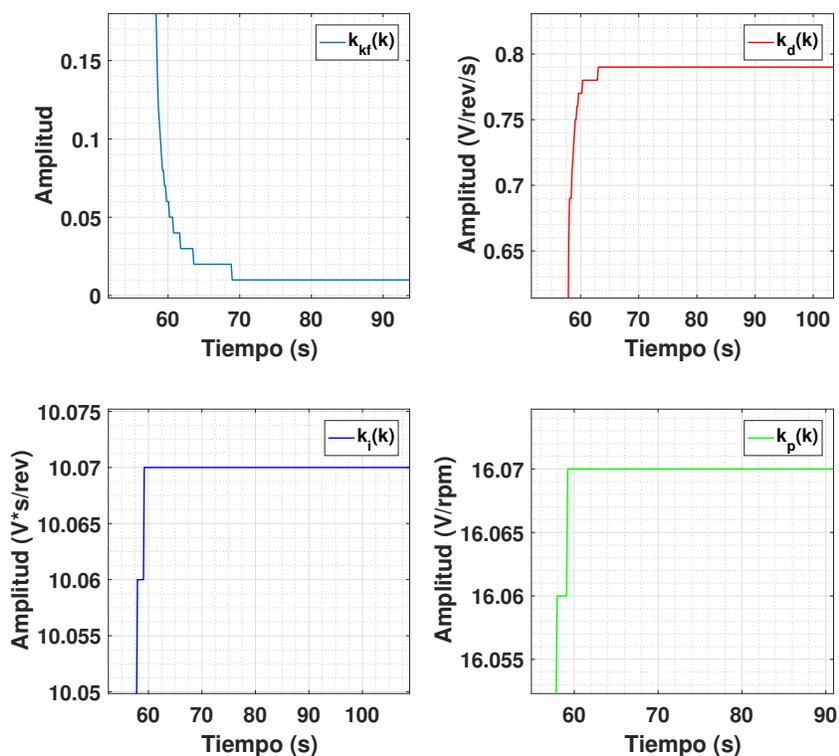


Figura 5.16: Transitorios del comportamiento de las ganancias de controlador subóptimo auto-sintonizable.

De forma análoga la simulación se realizaron 10 pruebas y se obtuvo que para el controlador PID ($u(k)$) el consumo fue:

$$\sum_{i=1}^{10} \left[\sum_{k=1}^N u(k) \Delta T \right] / 10 = 261.6367, \quad (5.3)$$

mientras que la señal de control promedio para $u^*(k)$ es:

$$\sum_{i=1}^{10} \left[\sum_{k=1}^N u^*(k) \Delta T \right] / 10 = 208.3489, \quad (5.4)$$

En la gráfica de la Figura(5.14) se muestra una comparativa entre las señales de los dos controladores, la mejora respecto del controlador PID Subóptimo Autosintonizable con el convencional es del 20.36 %, que es muy cercana a los resultados de simulación, con ello puede verificarse la eficiencia del algoritmo propuesto experimentalmente.

5.5. Resultados del sistema de visión y sensor calidad de aire del UAV

El sistema de visión es empelado para, mostrar en la interfaz gráfica la vista de la cámara embarcada en el robot, además, se usa para correr el algoritmo usado para calcular el desplazamiento, éste está desarrollado en el sistema operativo Raspbian de la Raspberry Pi 3B®, empleando lenguaje Python y la librería OpenCV V4.0®, mediante la configuración de los parámetros de Lucas Kanade [23] procede a calcular 3 descriptores, lo anterior para evitar retardos que afecten la lectura, a partir de dichos descriptores se calcula un "Descriptor Promedio", mostrado en la Figura 5.17, en la parte inferior derecha con 3 colores, es este punto el que genera las coordenadas x , y , relativas al desplazamiento que el robot presenta en el plano de trabajo paralelo al suelo.

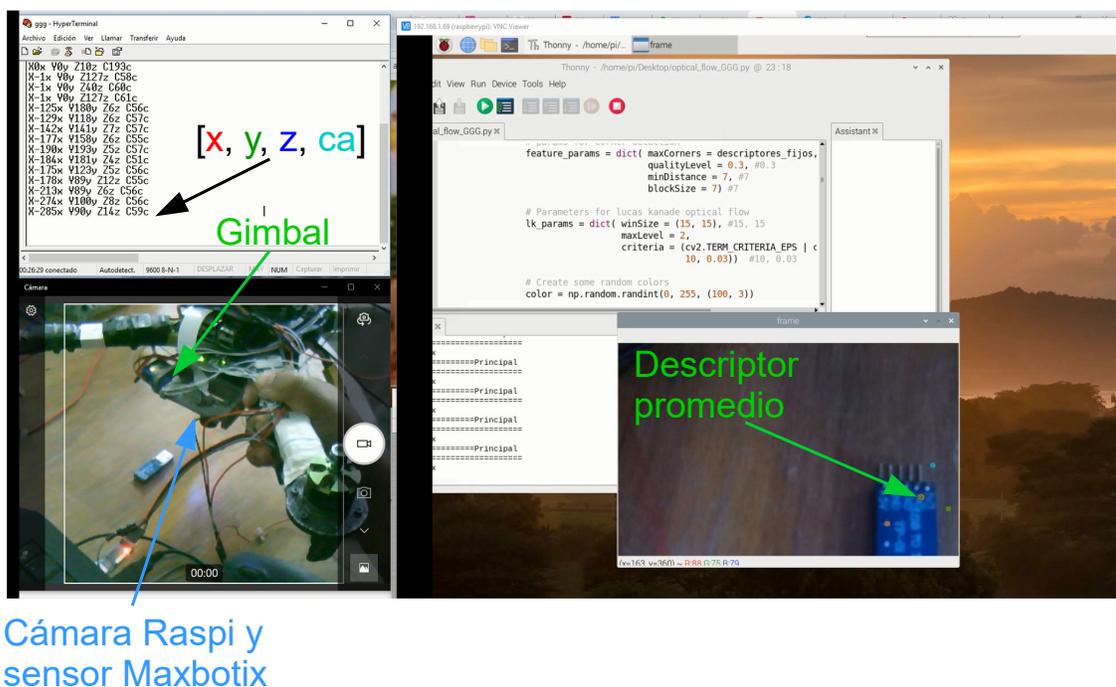


Figura 5.17: Resultados al aplicar Lucas Kanade en Raspberry® y enviar posición en x , y , z y calidad del aire (ca) mediante protocolo RS232.

Cuando el robot se mueve y se pierden los descriptores hallados previamente, se hace un nue-

vo cálculo y se envía el desplazamiento correspondiente a los nuevos descriptores, los datos son enviados por medio de protocolo RS232 a la tarjeta RASPIX, en dicha tarjeta se lee la información entregada por el sensor ultrasónico, que permite modificar la resolución del cálculo de movimiento presentado por el robot, en función a la altura a la que éste se encuentre con respecto del suelo, además, de leer la señal de voltaje correspondiente al sensor MQ137 de calidad del aire.

Las lecturas de los sensores y el cálculo de los desplazamientos en el plano de trabajo, estimados por el sistema de visión, son concatenados de la siguiente forma:

Figura 5.18: Cadena de caracteres concatenada para ser enviada por protocolo RS232.

Las letras agregadas, por ejemplo, para el caso del desplazamiento en x : $X\dots x$, permiten que se pueda decodificar a qué variable corresponde el valor que se encuentra entre dichos caracteres, esto es usado en el algoritmo del PIX-HAWK[®] para evitar pérdida de información, o lectura de datos errónea. La trama antes mencionada es enviada cuando la tarjeta RASPIX recibe un carácter especial, en este caso "#", con ello se evita saturar el buffer de llegada del PIX-HAWK[®], ya que sólo cuando este último envíe dicho carácter especial, la tarjeta mandará la información de lo contrario, no, lo anterior se logra por medio de interrupciones.

El Gimbal (ver Figura B.20) permite que el sensor ultrasónico y la cámara tengan siempre la misma orientación, con ello se compensan las perturbaciones causadas por el movimiento del UAV, inherente a traslación y rotación del robot aéreo. La velocidad de respuesta del Gimbal permite que los fotogramas sean tomados correctamente, mientras los movimientos sean con una relativa baja velocidad, en la siguiente figura es mostrado el Gimbal armado para el UAV y los componentes que lo integran.

La tarjeta RASPIX desarrollada es mostrada en la Figura 5.20, con ella se facilita la integración de la información generada por los diferentes sensores y del sistema de visión, además de que da versatilidad al controlador de vuelo, ya que evita que éste, tenga que estar gestionando datos en tiempo real con el sistema de visión, puesto que esta tarea es ejecutada por la tarjeta RASPIX.

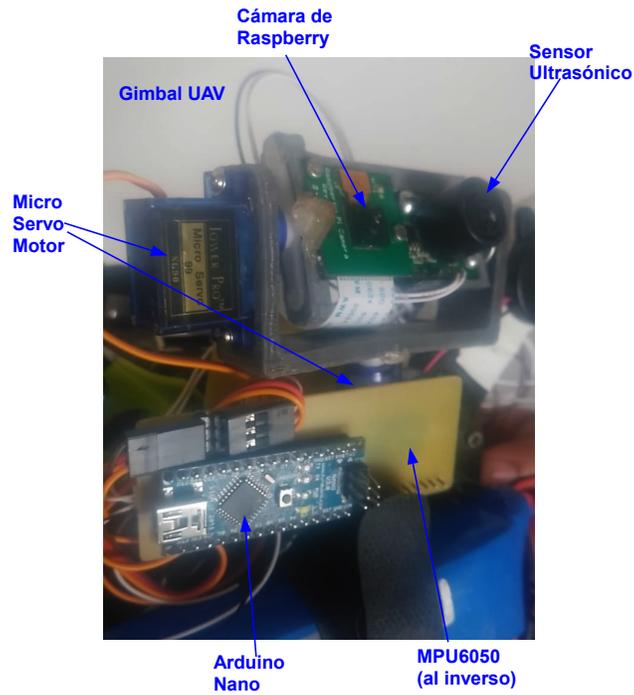


Figura 5.19: Gimbal del UAV armado.

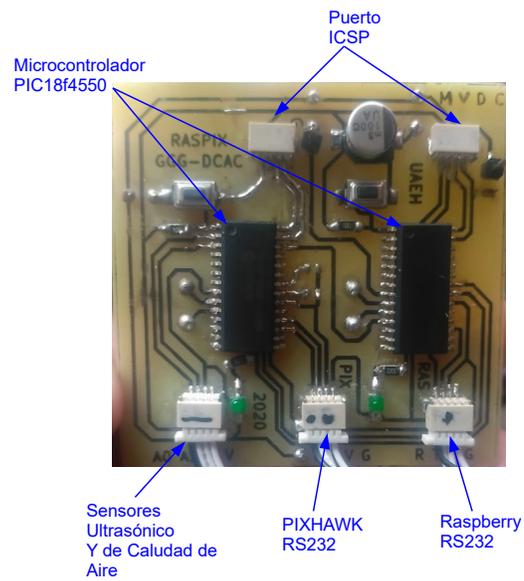


Figura 5.20: Tarjeta electrónica RASPIX.

En el contexto del dióxido de carbono (CO₂), no existe un índice de calidad del aire específico que se utilice de manera generalizada como el AQI (Air Quality Index) para otros contaminantes como el ozono (O₃), el dióxido de azufre (SO₂), el dióxido de nitrógeno (NO₂), las partículas suspendidas (PM₁₀ y PM_{2.5}), y el monóxido de carbono (CO). Sin embargo, la concentración de CO₂ se monitorea habitualmente en interiores para evaluar la calidad del aire, ya que niveles elevados pueden indicar una ventilación insuficiente y potencialmente aumentar el riesgo de transmisión de enfermedades respiratorias. Para CO₂, generalmente se consideran seguros los niveles que están por debajo de 1,000 ppm (partes por millón) según la mayoría de las pautas internacionales, aunque diferentes organizaciones y países pueden tener sus propias normativas y estándares de referencia. Es importante mencionar que el CO₂ a nivel ambiental también se monitorea como parte de las mediciones de gases de efecto invernadero, pero estas mediciones se utilizan más en el contexto del cambio climático que en la calidad del aire relacionada con la salud humana inmediata. Con los datos obtenidos de la cadena de caracteres antes descrita, se alimenta la gráfica mostrada en la Figura 5.21, la cual pertenece al sistema SCADA de este proyecto, en esta etapa de la interfaz, el usuario puede observar los niveles de CO₂ registrados por el sensor embarcado en el robot aéreo, y con base en la norma comparar si la calidad del aire monitoreado es excelente, bueno, regular o malo según la escala mostrada en dicha Figura. Es importante mencionar que, además, se registra en tiempo real esta medida dentro de la memoria micro SD del Pixhawk[®], y en un archivo de texto que genera el sistema SCADA, esta lectura, para poder analizar los datos posteriormente.

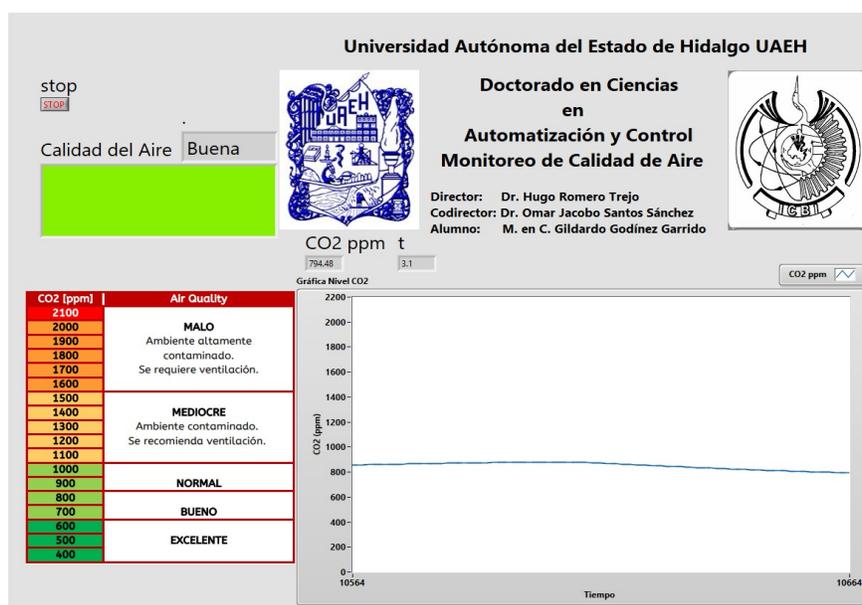


Figura 5.21: Pantalla de monitoreo de calidad del aire.

5.6. Resultados sistema de recolección de muestras de suelo del ULV

En la Figura 5.22 se muestra el sistema de recolección de muestras de suelo, el cual ha sido descrito en el Capítulo 3.1.2, como se puede notar en la imagen se cuenta con un perforador y un sistema de succión que permite que las partículas de tierra que se generan por la perforación puedan ser recolectadas mediante aspiración, a través, de una manguera flexible. Se diseñó una protección basada en un tubo transparente, denotado en la imagen como protector, el cual tiene la característica de ser retráctil, y recobrar su posición original por medio de un resorte, y se encuentra dispuesto de forma concéntrica al perforador, como se puede notar en dicha figura, con este mecanismo se puede proteger la zona de perforación, ayudando a evitar que el viento o algún factor externo altere la muestra, además de ayudar al aspirador a tomar la muestra, ya que las partículas que se desprenden, por abrasión son succionadas instantáneamente.



Figura 5.22: Sistema de toma de muestras de suelo.

La muestra recolectada es almacenada en el contenedor del aspirador que está dispuesto en la parte trasera del robot, y posteriormente puede ser extraída por el usuario, se realizaron pruebas que demostraron que el sistema es eficiente, y con ello permite que el robot extraiga muestras de suelo en lugares de difícil acceso, que pone en riesgo la integridad del usuario, o donde no se requiere que éste intervenga en el proceso. Se eligió que el desprendimiento de partículas fuera por un método abrasivo para evitar que la pérdida de afilado de la herramienta afecte el proceso, es importante mencionar que al usarse este proceso la muestra contiene partículas de la herramienta, que deben ser consideradas al momento de hacer el análisis de la muestra.

5.7. Resultados experimentales mejora en la aproximación de posición del UAV

Debido a que error acumulado que causa el barómetro, no es fiable aplicar leyes de control a bajas alturas en el UAV, esto es un problema, ya que afecta en función del tiempo a una variable del estado del sistema que es la altura $Z(t)$, esto implica que si no se modela la dinámica de dicho sensor la ley control no permitirá garantizar estabilidad en un punto de equilibrio, por ello, se implementa un sensor YPLidar, que permite mejorar el “bias” causado por barómetro, desde 0.11 m hasta una altura de 12 m, ya que para alturas superiores se supera el rango de trabajo de dicho sensor, sin embargo, para los fines del presente proyecto es un rango suficiente. En Figura 5.23 se muestra un ejemplo de un off-set causado por la lectura del barómetro a lo largo del tiempo de $-1m$, cuando en realidad la señal debería ser de $0.2m$, gracias al sensor Lidar se evita hacer medidas erróneas en la altura del UAV, sin que sea necesario fusionar la señal del barómetro con otros sensores.

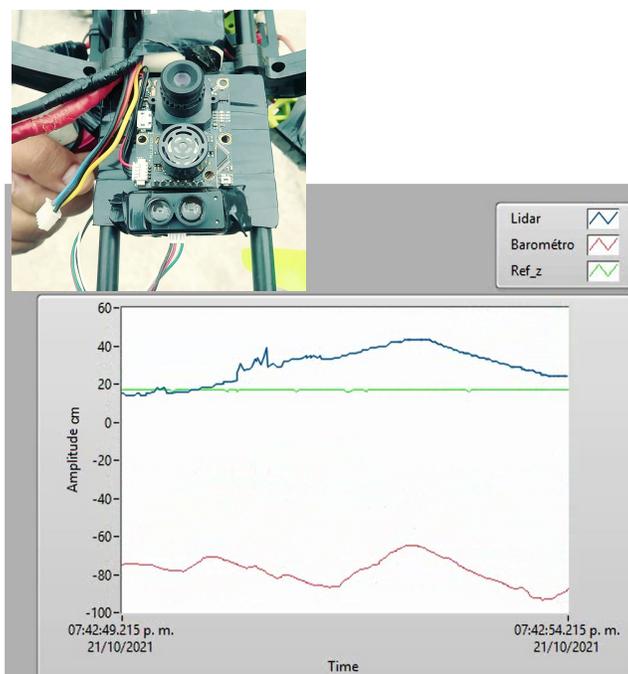


Figura 5.23: Mejora en la aproximación de la posición $Z(t)$, mediante sensor YPLidar de 12m.

Respecto a la posición en el plano de trabajo correspondiente a las coordenadas $X(t)$ e $Y(t)$ se implementa un sensor PX4-FLOW® que permite mejorar la aproximación de dichas posiciones

espaciales, lo anterior debido a que como es bien sabido la aproximación de posición obtenida del GPS puede tener un error de hasta 20m, en la Figura 5.24, se muestra una comparativa en la posición aproximada por el sistema de navegación GPS contra la posición calculada por el sensor de flujo óptico antes mencionado, esto al seguir la referencia de un cuadro de 6x6 m.

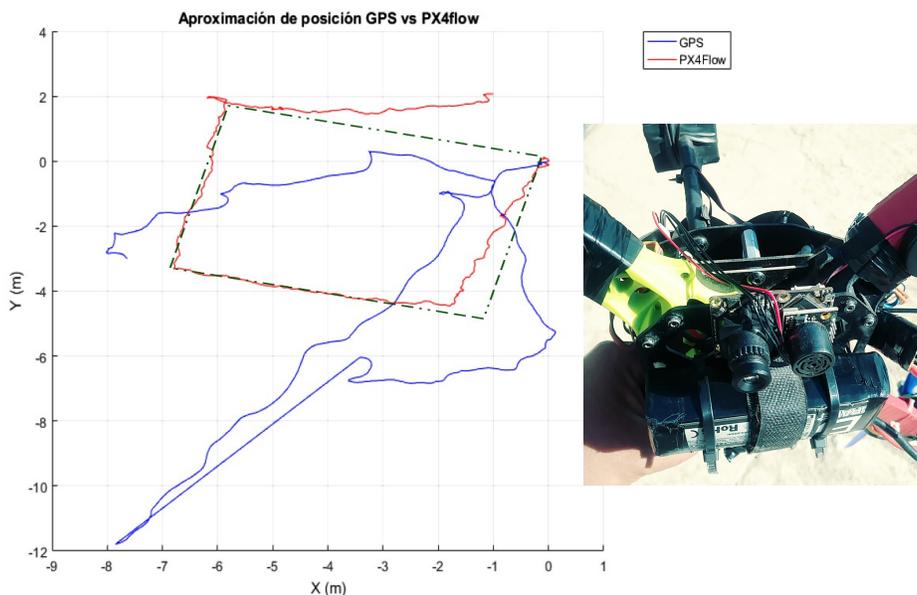


Figura 5.24: Mejora en las aproximaciones de las posiciones $X(t)$ e $Y(t)$, mediante sensor PX4-FLOW®.

La instrumentación e implementación de los dos sensores anteriores YPLidar y PX4-FLOW®, permite mejorar la aproximación de la posición del robot aéreo, en las coordenadas $X(t)$, $Y(t)$ y $Z(t)$, esta mejora ayuda a tener todo el estado en posición y orientación, con una reducción considerable en el error de medida.

5.8. Resultados experimentales, control de posición y orientación del UAV

Contemplando la aplicación de los sensores PX4FLOW® y el sensor Mini Lydar, se mejora la aproximación de la posición del UAV ($x(t)$, $y(t)$ y $z(t)$), lo que ha permitido aplicar un controlador PID en la posición del UAV sin depender exclusivamente de la señal del GPS, las ganancias del controlador, para cada variable de posición, orientación y sus respectivas velocidades, así como las saturaciones en la señal de control ($Sat - Cont_{pos}$, $Sat - Integral$) para

las posiciones y la frecuencia de corte ($FC - Derivada$) para un filtro pasa bajas usado para la acción derivativa, son mostradas en la Figura 5.25.

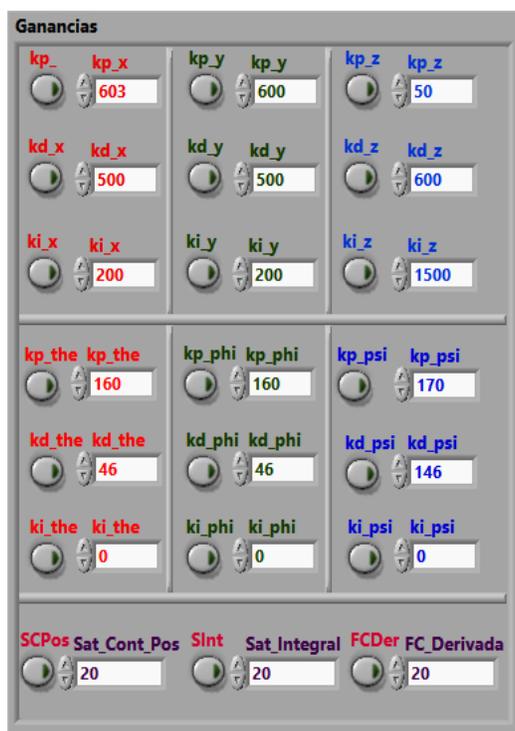


Figura 5.25: Ganancias del controlador PID asignadas mediante LABview®.

La ventana mostrada en la Figura 5.25, corresponde el objeto virtual, del sistema SCADA, que permite al usuario modificar las ganancias del robot aéreo, para ello primero se debe insertar él, el valor de la ganancia a modificar y después mediante el botón que está a la izquierda de ésta, se válida el envío del valor ajustado, esto permite a usuario modificar ganancias sin modificar el firmware del PIX-HAWK®.

En la Figura 5.26 se muestra la trayectoria, seguida por el dron con los siguientes puntos x y y de consigna: $P_1 = (0, 0)$, $P_2 = (0, -4)$, $P_3 = (4, -4)$, $P_4 = (4, 0)$, $P_5 = (8, 0)$, $P_6 = (-4, 8)$, $P_7 = (12, -4)$ y $P_8 = (12, 0)$, con una velocidad de 0.3 m/s, a una altura de 2 m, usando las ganancias de la Figura 5.25, para el controlador. En la figura 5.27 se puede apreciar una vista superior del recorrido realizado por el robot aéreo en seguimiento de la trayectoria de referencia, que se muestra en color gris, el error promedio en $x(t)$ es de 0.5 m y en $y(t)$ de 0.6 m, para $z(t)$ de 0.3 m.

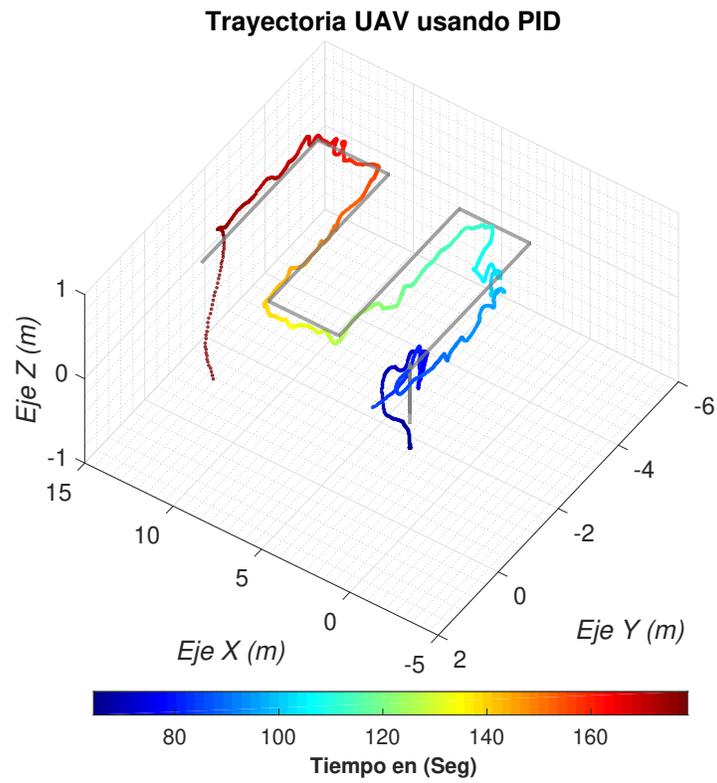


Figura 5.26: Trayectoria de UAV para rutina de vuelo.

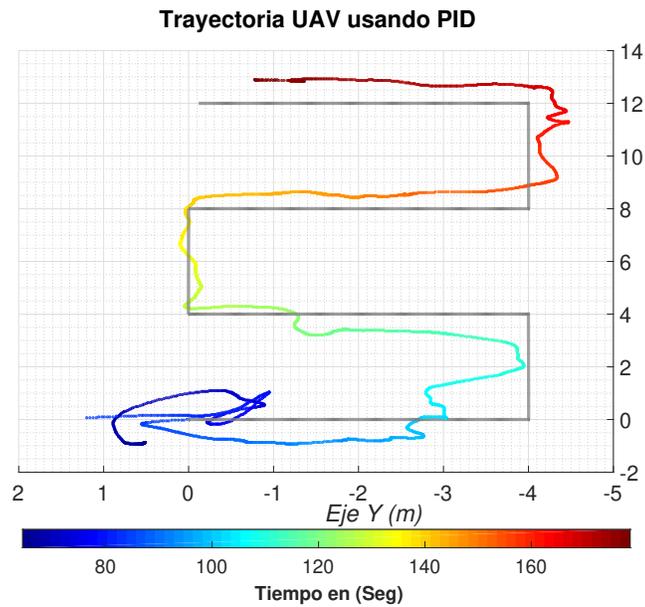


Figura 5.27: Trayectoria de UAV para rutina de vuelo vista aérea.

Las posiciones $x(t)$, $y(t)$ y $z(t)$ y sus correspondientes velocidades ($\dot{x}(t)$, $\dot{y}(t)$ y $\dot{z}(t)$) se observan en las gráficas de la Figura 5.28. Estas señales se obtuvieron del sistema de navegación del dron, combinando información de la unidad de medida inercial y el sistema de posicionamiento global.

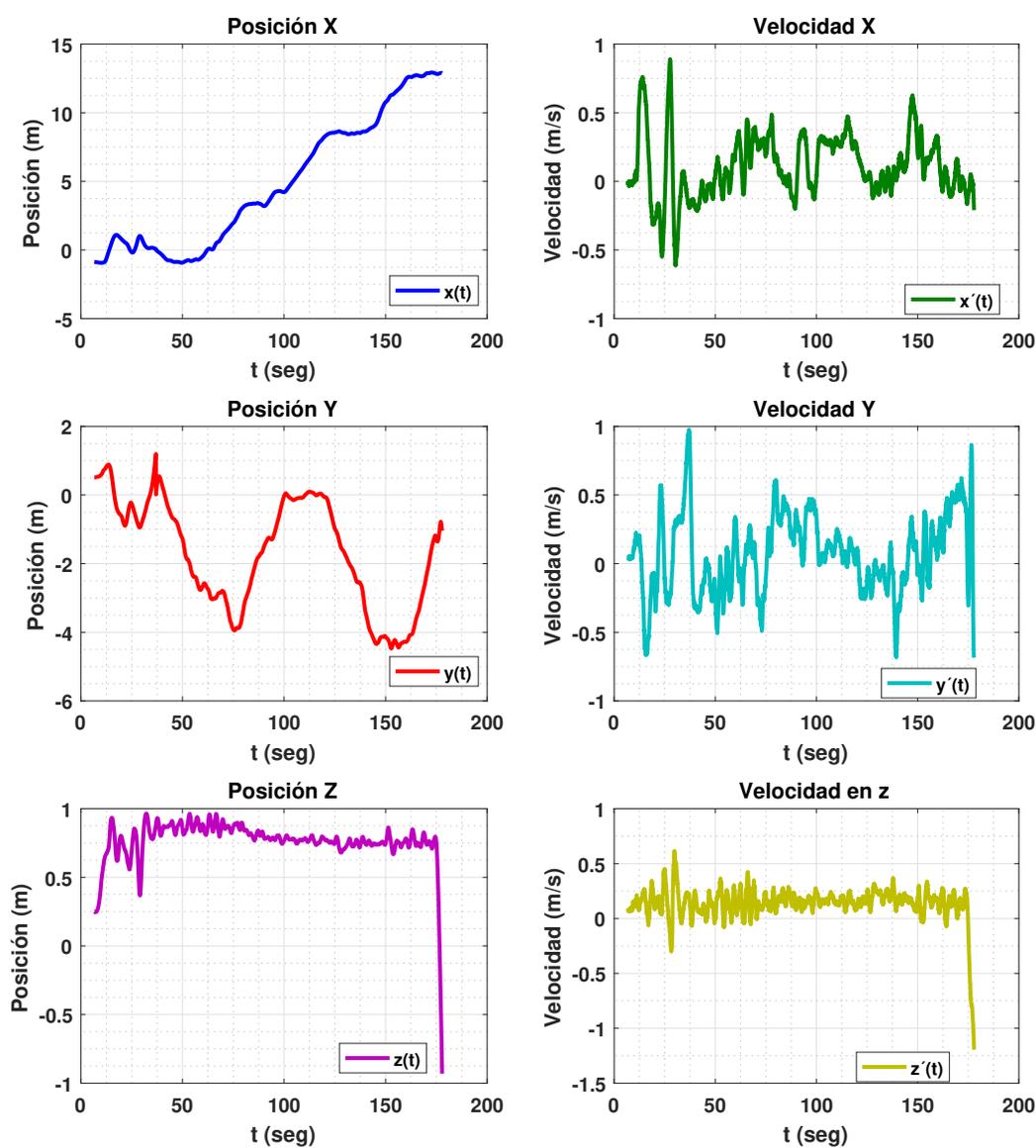


Figura 5.28: Posiciones y velocidades del UAV al aplicar controlador PID en posición y orientación.

Los ángulos: de cabeceo $\theta(t)$, alabeo $\phi(t)$ y guiñada $\psi(t)$, además, de la velocidad angular relativa a cada uno de ellos, son ubicados en las gráficas de la Figura 5.29,

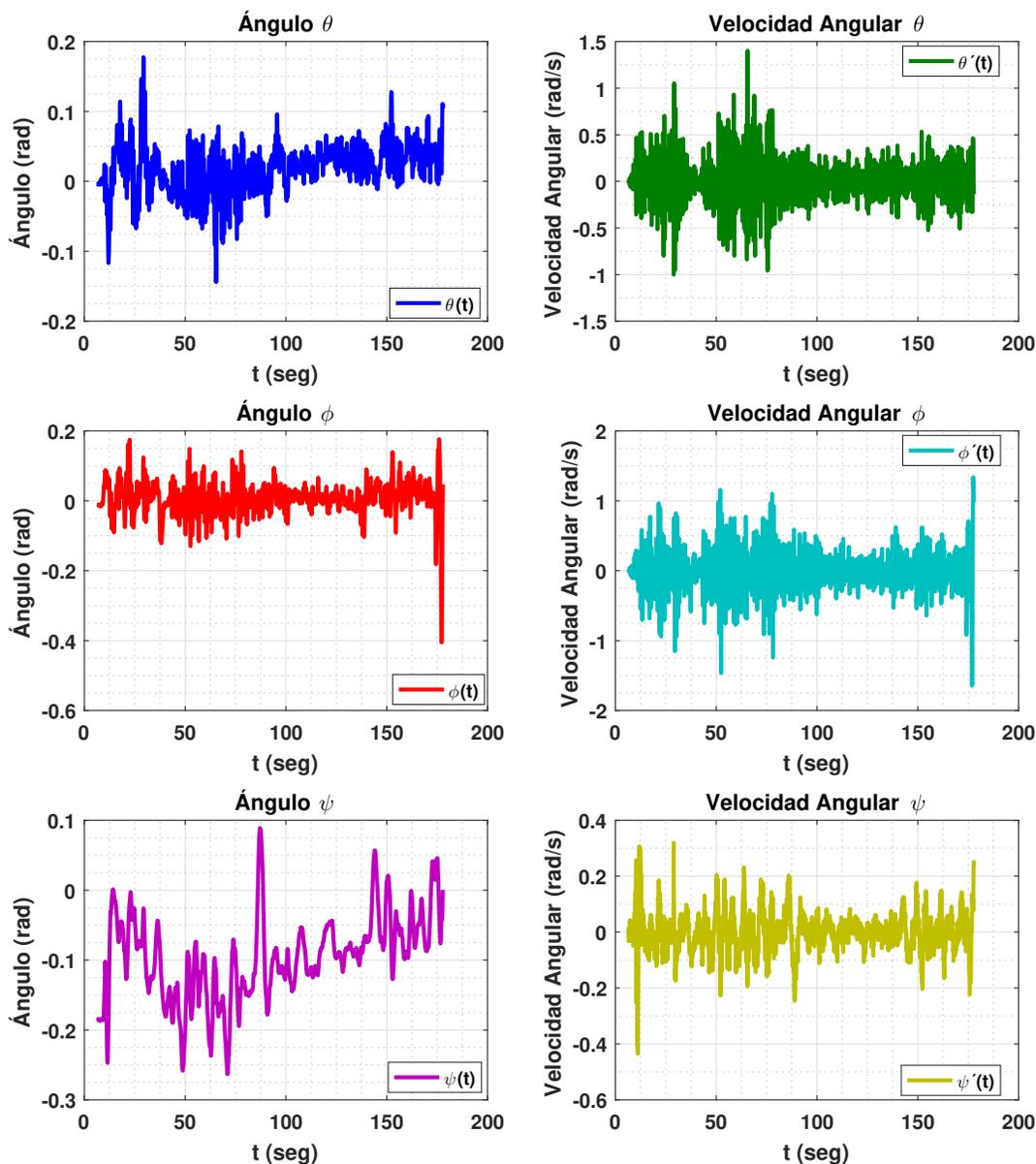


Figura 5.29: Orientación y velocidades angulares del UAV al aplicar controlador PID en posición y orientación.

Las señales de control producidas durante el seguimiento de la trayectoria, correspondientes a los 3 torques ($\tau_\theta(t)$, $\tau_\phi(t)$ y $\tau_\psi(t)$), para la orientación y la fuerza ($u(t)$) para la elevación, se pueden ver en las gráficas de la Figura 5.30.

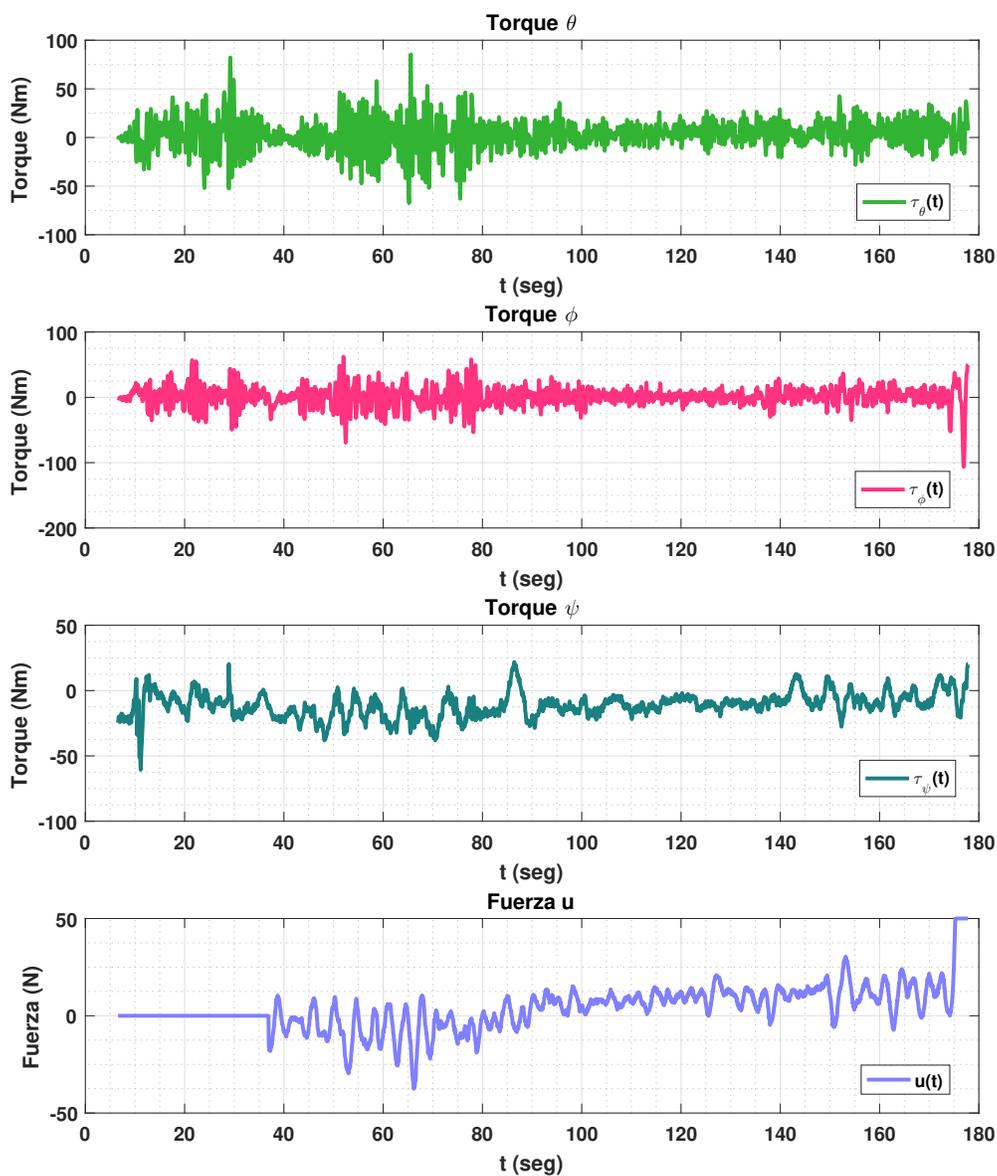


Figura 5.30: Torques y fuerza aplicados al UAV durante el seguimiento de trayectoria.

5.9. Resultados experimentales, control de posición y orientación del ULV

Los resultados experimentales obtenidos al aplicar un controlador PID en el robot todoterreno son mostrados a continuación, la trayectoria es definida por los puntos $P_1(0,0)$, $P_2(1,0)$, $P_3(1.5,1)$, $P_4(1.5,1.5)$ y $P_5(2.5,3)$, estos valores están en metros, y representan los waypoints a alcanzar a lo largo de la ruta. Considerando un tipo de navegación polar, es decir donde se manipula el avance frontal y el ángulo de guiñada del robot, las ganancias del controlador PID usadas son: $Kp_\phi = 80$, $Kp_x = 120$, $Ki_\phi = 10$, $Ki_x = 0.1$, $Kd_\phi = 1$ y $Kd_x = 0.1$, donde las ganancias del ϕ , son para la orientación del robot en guiñada y las ganancias para x corresponden al avance en ese eje de referencia del robot.

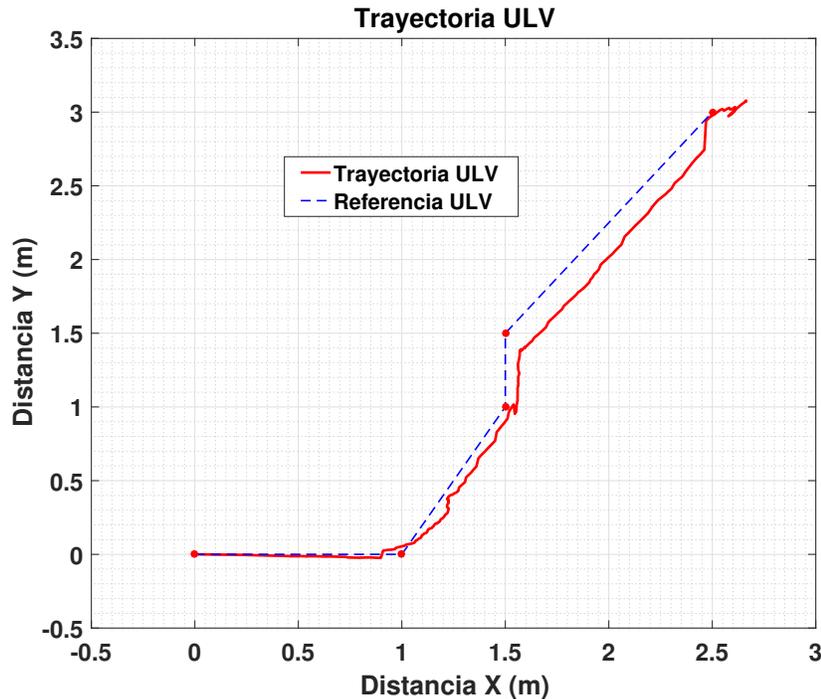


Figura 5.31: Trayectoria del ULV al aplicar controlador PID en posición y orientación.

Las posiciones angulares y sus velocidades son apreciadas en la Figura 5.32, donde $\phi(t)$ representa el ángulo de guiñada y $\dot{\phi}(t)$ su velocidad angular, de forma respectiva $r(t)$ y $\dot{r}(t)$ se usan para indicar la posición y velocidad angular del ángulo de alabeo del robot; y $\rho(t)$ y $\dot{\rho}(t)$ de forma análoga para la orientación en cabeceo del ULV.

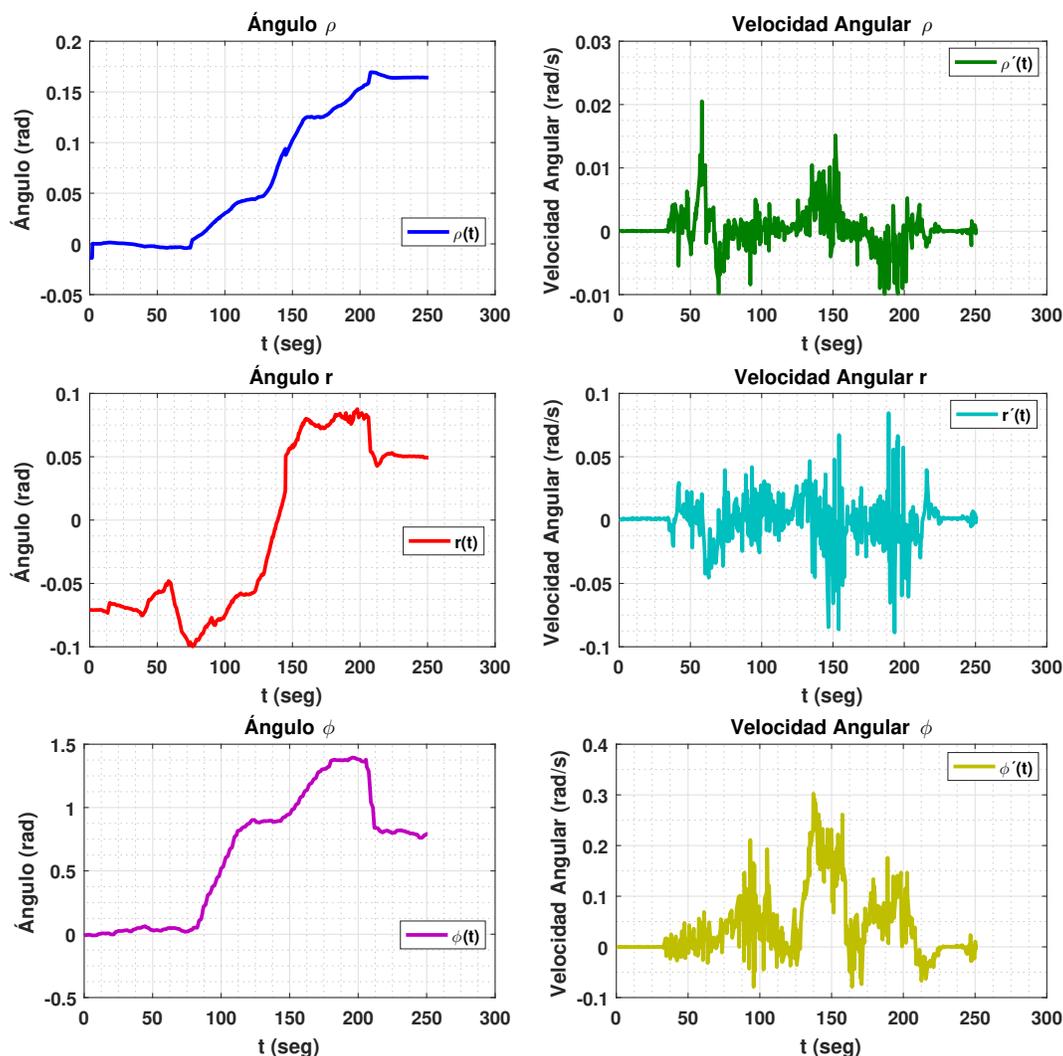


Figura 5.32: Ángulos y velocidades angulares del robot ULV.

Respecto a las posiciones del robot y sus velocidades, éstas son mostradas en la Figura 5.33, puede notarse que la posición en $z(t)$ y su velocidad permanecen constante, la primera que es la lectura del sensor de altura del carro 0.18 m, y la segunda en cero, ya que el robot realizó esta tarea en un ambiente plano sin elevaciones considerables. La combinación de las dos posiciones en el plano de trabajo XY , además, de la referencia en el seguimiento de la trayectoria programada para este experimento, son mostradas en la Figura 5.31.

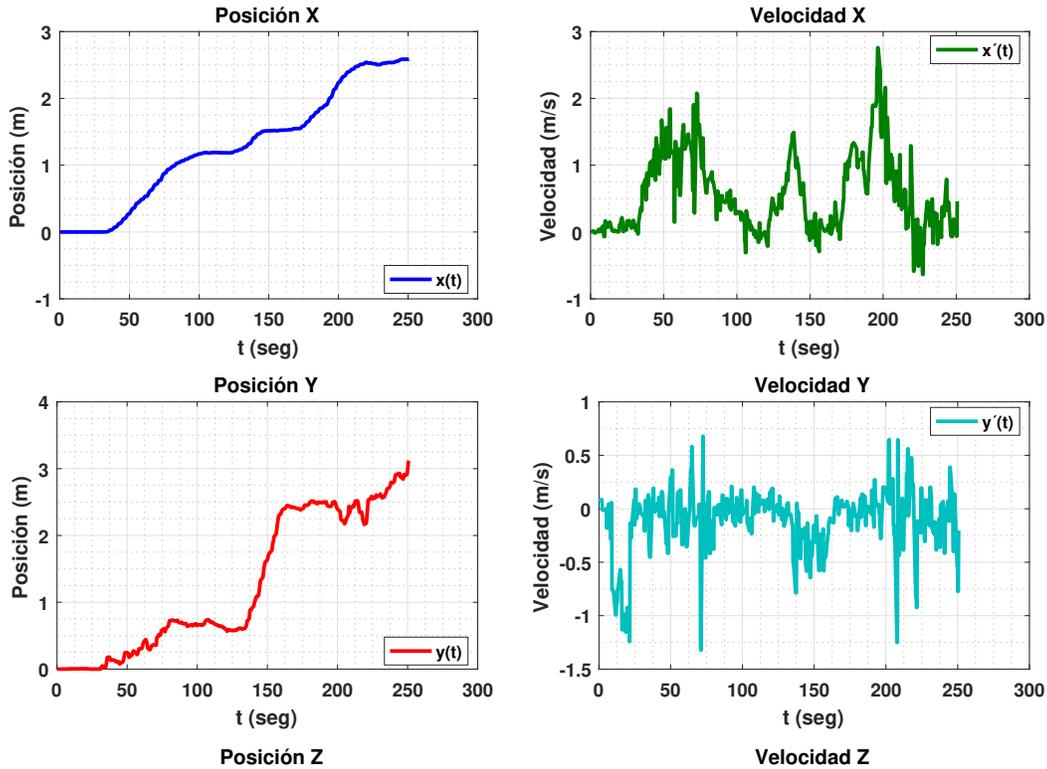


Figura 5.33: Posiciones y velocidades del robot ULV.

Los ángulos correspondientes a las articulaciones laterales del robot, $\theta_l(t)$ para el lado izquierdo; y $\theta_r(t)$ para el lado derecho, además, del ángulo de la articulación delantera $\beta(t)$, perteneciente al mecanismo del seguidor del robot, así como, la posición de carro Z_{car} , que corresponde a la posición a la que se encuentra el carro del mecanismo de perforación y aspiración, se muestran en la Figura 5.34. Estas variables se obtienen de un ESP32® por medio de sus canales analógicos y son enviadas vía RS232 al PIX-HAWK®, esto genera un retardo el cual se puede notar en cada paso de muestreo de las señales, para este experimento el carro en z permaneció en su posición de home que es a 0.18 m del suelo.

Cada una de las ruedas de tracción del robot, generaron en su velocidad angular el comportamiento mostrado en la Figura 5.35. Estas señales se producen a partir del algoritmo de control de posición y orientación calculado en el controlador del robot, este algoritmo permite generar 6 señales PWM u_{R1}, \dots, u_{R6} , que son las señales de control para cada rueda respectivamente, a partir de las dos señales de control calculadas, para la posición $x(t)$ y para el ángulo $\phi(t)$, de estas 6 señales de control se producen las velocidades w_1, \dots, w_6 , que corresponden a las velocidades angulares para cada rueda.

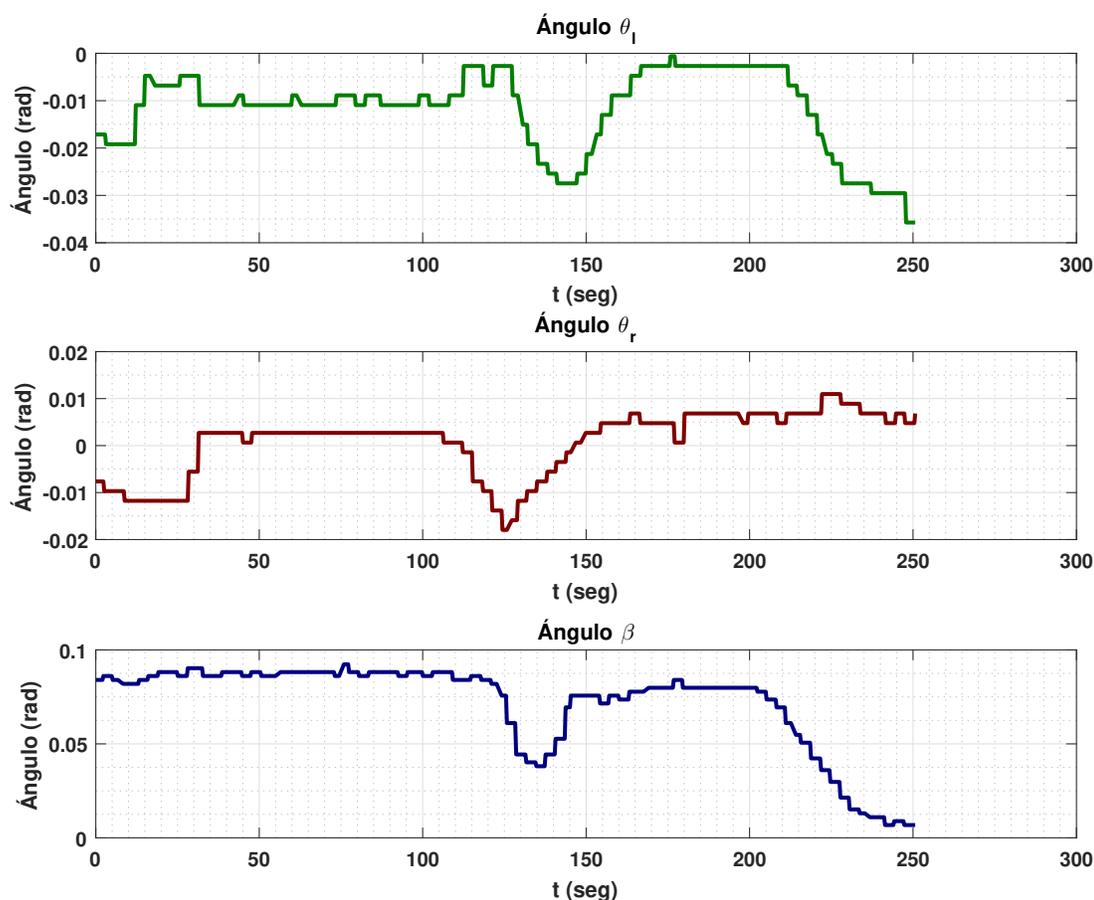


Figura 5.34: Ángulos de articulaciones y posición del carro del ULV.

Las señales de control $u(t)$ que representa la fuerza de empuje en el eje x del marco de referencia del robot, y τ_ϕ que representa el torque requerido para hacer rotar el robot en sí eje de guiñada, son graficadas en la Figura 5.36.

Algunos fenómenos, como el retardo de envío de datos entre el controlador general y los microcontroladores usados para medir las variables en las articulaciones, causan que las señales de éstas, presenten un mayor tiempo de muestreo, como puede notarse en la Figura 5.34, además, los problemas presentados por los sensores como el magnetómetro como el bias, causan afectaciones directas en el seguimiento de trayectoria de los robots, inherentes a estas perturbaciones magnéticas, y deben ser contemplados en la condición inicial de la orientación en cada robot, para tener un desempeño acorde al norte magnético de la tierra.

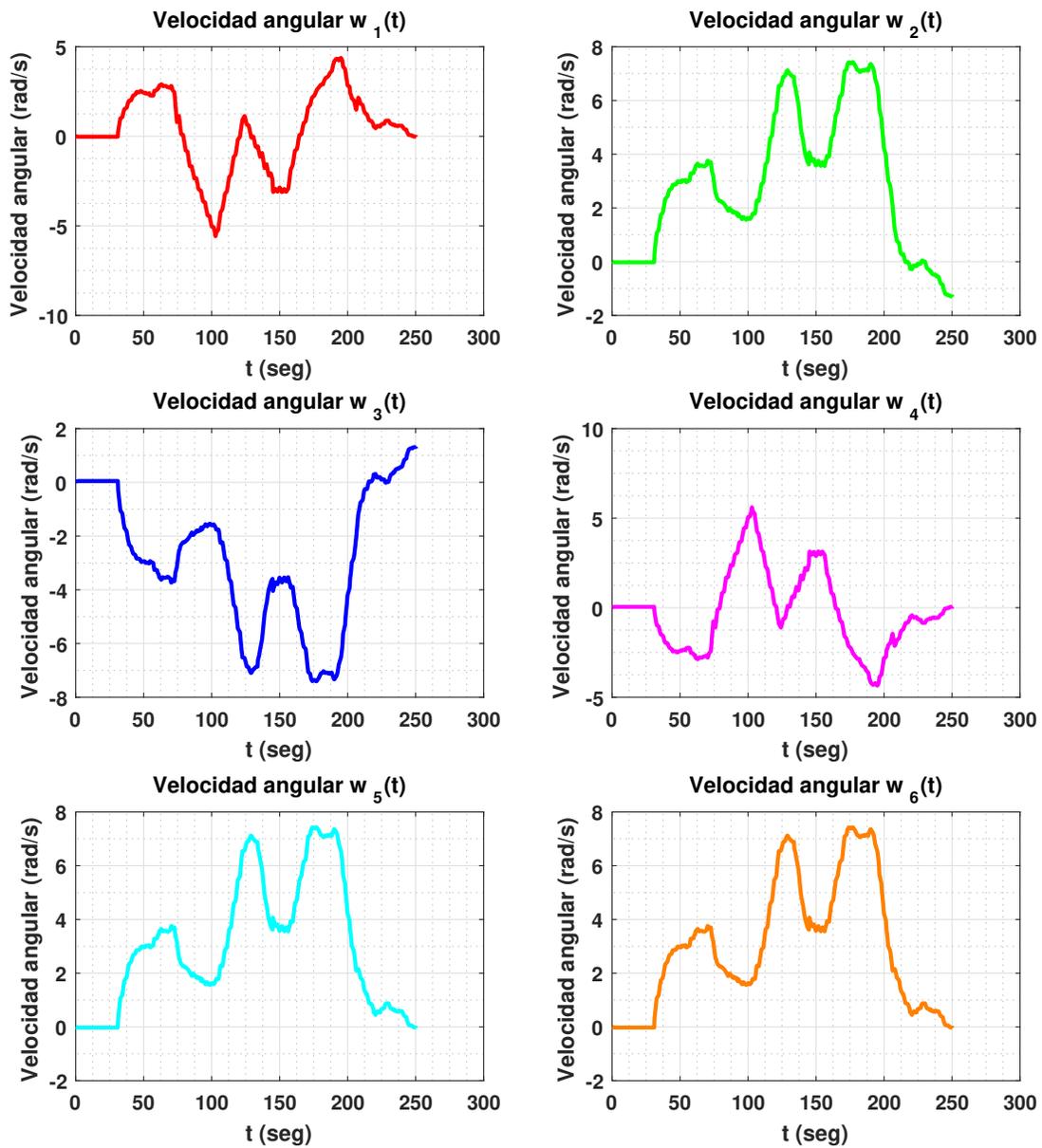


Figura 5.35: Velocidades angulares de los motores de tracción del ULV.

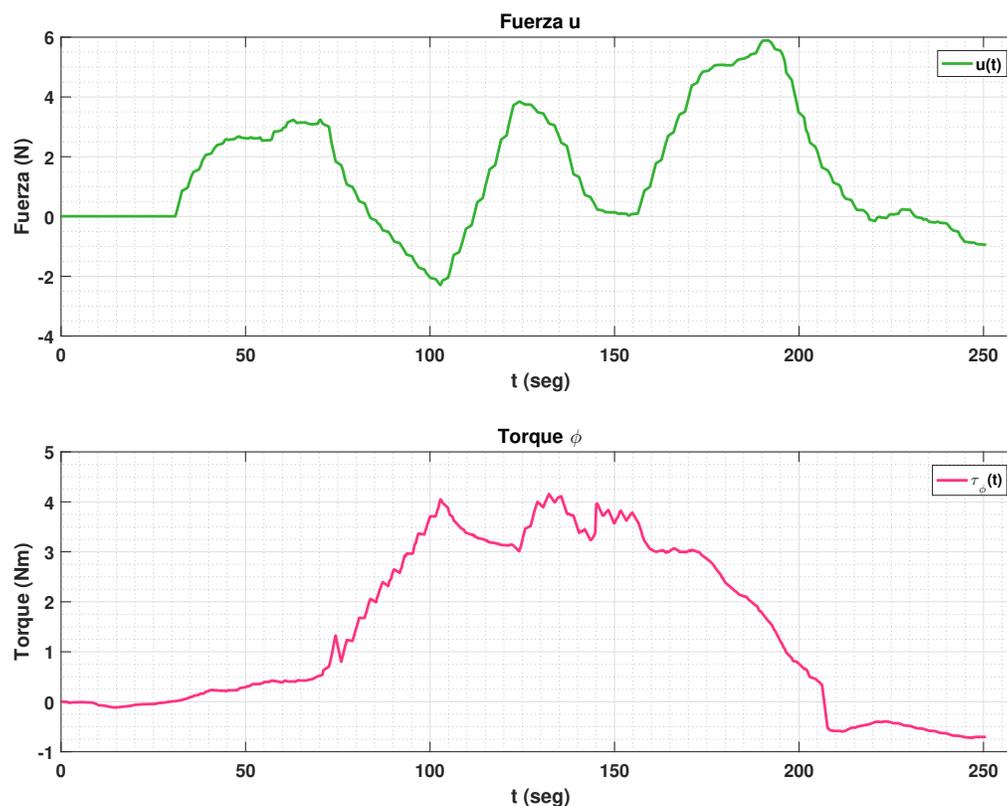


Figura 5.36: Señales de control del ULV.

5.10. Conclusión capítulo 5

En este capítulo, se presentan las simulaciones y resultados experimentales, demostrando la eficacia y aplicabilidad de los algoritmos y sistemas desarrollados en esta investigación. La realización de simulaciones detalladas para ambos robots, UAV y ULV, valida los algoritmos de seguimiento de trayectoria, siendo un paso esencial para garantizar la precisión y la eficiencia de estas plataformas en aplicaciones reales.

La Simulación de Seguimiento de Trayectoria para el UAV y el ULV establece la eficacia de los algoritmos en entornos controlados, preparando el terreno para su implementación en situaciones reales. Estas simulaciones son cruciales para anticipar y ajustar el comportamiento de los robots en diversas condiciones.

La Simulación del Controlador Subóptimo Autosintonizable para un motor actuador del ULV es seguida por los Resultados Experimentales correspondientes. Estos muestran la capacidad del controlador para ajustarse y optimizar el rendimiento del motor en tiempo real, un avance significativo para la automatización y el control en robótica.

Los Resultados Experimentales del Sistema de Visión e Instrumentación de Sensor de Calidad de Aire del UAV demuestran la precisión y fiabilidad de este sistema en la medición de CO₂, confirmando su funcionalidad en el monitoreo ambiental. Además, los Resultados Experimentales del Sistema de Recolección de Muestras de Suelo del ULV evidencian la efectividad de este sistema en la recolección y análisis de muestras, un aspecto clave para la investigación ambiental y agrícola.

Los experimentos adicionales, que incluyen la mejora en la aproximación de posición del UAV y el control de posición y orientación tanto del UAV como del ULV, resaltan la precisión y adaptabilidad de estas plataformas en entornos complejos y dinámicos.

En resumen, los experimentos y simulaciones presentados en este capítulo no solo validan los algoritmos y sistemas desarrollados, sino que también demuestran que cada robot está perfectamente adaptado para su respectivo ambiente de aplicación. Los resultados obtenidos permiten alcanzar los objetivos planteados en esta tesis, mostrando que tanto el sistema de monitoreo de calidad de aire como el de recolección de muestras de suelo funcionan eficazmente, reforzando el potencial de estas plataformas robóticas en aplicaciones prácticas y reales.

Capítulo 6

Conclusiones y trabajo futuro

En este capítulo se abordan las conclusiones generales obtenidas, en función a los objetivos planteados, y se comentan trabajos futuros propuestos, así como, las mejoras y recomendaciones que se consideran pertinentes, para hacer las adecuaciones o validaciones que el autor considera importantes en los aspectos técnicos de este trabajo.

6.1. Conclusión

En función de los objetivos planteados, el desarrollo de la interfaz gráfica, los dos robots; y los resultados de simulación y experimentales obtenidos, se puede concluir lo siguiente:

Desarrollo de la Interfaz Gráfica y Sistema SCADA Se desarrolló una interfaz gráfica, en 100 % de su totalidad. Esta interfaz representa un elemento vital, facilitando una interacción entre el usuario y el sistema SCADA, así como con los robots integrados en la plataforma heterogénea de robótica móvil. Notablemente, la interfaz permite ajustar de manera inalámbrica las ganancias de los controladores de los robots a través del protocolo RS232, sin la necesidad de intervenir en el firmware de los controladores Pix-Hawk incorporados. Adicionalmente, posibilita un acceso directo a variables críticas en cada robot, tales como señales de control, estado actual y lecturas de sensores, consolidando una integración efectiva y eficiente. Está interfaz fue creada en LABview[®], que es un programa industrial, se contempla, además, la interacción con un ambiente virtual basado en Unity[®] donde se incluye cada uno de los robots, permitiendo interactuar en tiempo real con cada uno de ellos y los objetos 3D que los representan en dicho ambiente, esto da la posibilidad de generar simuladores, para validar leyes de control y hacer más dinámica la interacción entre el usuario y el sistema SCADA. El sistema SCADA permite

hacer guardado de datos de todas las variables tanto de monitoreo como de control de cada uno de los robots.

Diseño y Construcción de un Robot Aéreo no Tripulado (UAV) El desarrollo ha contemplado la construcción de un robot aéreo no tripulado (UAV) de alta eficacia, dotado con un sistema de ala rotatoria cuatrirotor, fundamentado en la estructura SK450. Este UAV constituye una parte integral de la plataforma heterogénea de robótica móvil, encontrándose funcional y completado al 100%. Incorpora elementos esenciales tales como un controlador de vuelo PIX-HAWK®[®], GPS, y un sistema de radiocontrol, además de un módem X-bee que facilita la comunicación RF RS232 con el sistema SCADA. A su vez, el robot aéreo está equipado con una serie de sensores que optimizan su rendimiento en la identificación de áreas con potencial contaminación. La implementación de controladores PID ha sido efectuada con éxito, centrando los esfuerzos futuros en la optimización del control para garantizar una operación energéticamente eficiente, conforme a las estrategias delineadas en [4].

Diseño y Construcción de un Robot Terrestre no Tripulado (ULV) Este trabajo también ha dado lugar al diseño y construcción de un robot terrestre no tripulado (ULV), cuya arquitectura shrimp rover le confiere una flexibilidad notable, permitiéndole adaptarse eficientemente a diversas condiciones del terreno. Este vehículo, que alcanza un 100% en términos de su construcción e instrumentación, integra elementos tecnológicos avanzados, que incluyen un sistema de control PIX-HAWK®[®] y múltiples sensores que facilitan su navegación y operación. A fin de alcanzar un rendimiento óptimo, se identifica la necesidad de implementar leyes de control avanzadas para la navegación precisa y la integración efectiva del sensor RPLidar, conforme a las especificaciones presentadas en [27].

Se ha desarrollado una plataforma heterogénea de robótica móvil, consolidando la construcción y diseño exitoso de unidades robóticas aéreas y terrestres no tripuladas. Esto es sustancial, particularmente en lo que respecta a la implementación de una interfaz gráfica avanzada, que posibilita ajustes y configuraciones a vía RF desde dicha interfaz hacia los robots.

Asimismo, se ha formulado un algoritmo subóptimo autoajutable, destacando por su capacidad de minimizar el consumo energético en un 20% en cada motor del robot terrestre. Se anticipa que futuras investigaciones se puede aplicar optimización en el sentido energético en el robot de forma global, aplicando esta propuesta, que está basada en la combinación de 3 algoritmos, el PID, el filtro de Kalman y el descenso del gradiente, aprovechando las ventajas de los 3, para crear un PID autosintonizable en línea optimizado, permitiendo que el controlador ajuste las ganancias en tiempo real sin depender del modelo de la planta ni de la aproximación

de parámetros, esto es una aportación importante por las prestaciones técnicas y prácticas que la integración de los 3 algoritmos presenta.

En conjunto, los resultados preliminares indican una prometedora evolución en el ámbito de la robótica móvil, sugiriendo un potencial significativo para futuras aplicaciones prácticas y académicas, validado por los resultados experimentales y de simulación obtenidos hasta la fecha. Esta investigación subraya la viabilidad y la potencialidad de integrar tecnologías avanzadas en la conceptualización y realización de sistemas robóticos colaborativos y eficientes, proyectándose como una contribución significativa en el campo de la ingeniería robótica y la aplicación hacia el monitoreo ambiental, ya que los resultados, de lectura de calidad del aire y las tareas de recolección de muestras de suelo realizados durante este trabajo, permiten validar que esto es factible.

6.2. Trabajo futuro

Se pueden realizar varias mejoras adicionales al presente proyecto de investigación, como la aplicación de fotometría para recrear el ambiente en el que el robot todoterreno trabajará. Esta técnica puede ser utilizada a través del UAV para obtener información de la topografía del terreno, lo que permitiría optimizar el diseño de trayectorias que seguirá el ULV.

Además, otra posible mejora sería diseñar un PCB para el smart driver propuesto, a fin de reducir espacios y optimizar el consumo energético de los componentes. En este trabajo, el driver se emplea utilizando tarjetas integradas de forma separada, lo que aumenta el número de componentes y, por ende, el consumo de energía del robot.

Se debe trabajar más en la visión artificial de los robots, ya que en este trabajo sólo se usó el sistema de visión con fines de monitoreo, sería bueno hacer que dichos algoritmos cooperen con los de control para mejorar el desempeño de estos últimos, aplicando técnicas como SLAM por visión artificial, etc.

También es recomendable aplicar el algoritmo de control PID Subóptimo Autosintonizable al UAV, ya que para esta tesis, dicho algoritmo propuesto y que es una contribución de este trabajo, únicamente se aplicó al ULV.

Se recomienda hacer una comparación entre un PID Subóptimo Autosintonizable y un PID convencional en el ULV, para analizar el ahorro real aplicar este algoritmo de forma global, ya

que en este trabajo en particular se analizó sobre un sólo actuador del sensor.

Se recomienda hacer seguimiento de trayectorias con programación de evasión de obstáculos y optimización en esta tarea, ya que para la presente tesis, no se contempló ninguna optimización en este sentido.

Con el ambiente virtual creado en Unity® y el sistema SCADA desarrollado en LABview® es posible desarrollar simuladores, que permitan programar leyes de control en cada robot, esto aprovechando las ventajas que da el motor gráfico mediante sus físicas, por ello es importante considerar el desarrollo de un simulador que permita integrar ambos programas con fines didácticos y de investigación, ya que por medio de este tipo de simulador se pueden validar con un alto grado de certidumbre leyes de control.

Es recomendable hacer más pruebas de monitoreo ambiental y profundizar en la instrumentación para esta tarea, esto con ayuda de personal especializado en el área de la contaminación y la corrosión.

Por último, se podrían realizar otras comparativas en el consumo y desempeño entre los algoritmos, como la energía cinética y potencial, la corriente, etc., ya que para esta tesis sólo se abordó una comparativa en energía aplicada a los actuadores

Bibliografía

- [1] Khalil, H. K. *Nonlinear Systems*. Prentice Hall, 1996.
- [2] Slotine, J.J. and Li, W. *Applied Nonlinear Control*. Prentice Hall, 1991.
- [3] Castillo, P., Lozano, R. and Dzul, A. Stabilization of a mini-robotcraft having four rotors, *International Conference on Intelligent Robots and Systems IEE/RSJ* pp.2693-2698, 2004.
- [4] Santos, O., Romero, H., Salazar, S. and Lozano, R. Discrete Optimal Control for a Quadrotor UAV: Experimental Approach, *International Conference on Unmanned Aircraft Systems (ICUAS)* pp.1138-1145, 2014.
- [5] Raffo, G. V., Ortega, M. G. and Rubio, F. R. An integral predictive/nonlinear H_∞ control structure for a quadrotor helicopter, *Automatica*, vol. 46, pp. 29-39, 2010.
- [6] Tahar, M, Meguenni-Zemalache, K. and Omari, A. Control of an Under-Actuated X4-flyer using Integral Backstepping Controller, *Przegla D Elektrotechniczny (Electrical Review)*, R. 87 NR 10/2011, pp. 251-256, 2011.
- [7] Bouabdallah, S. and Siegwart, R. Full Control of a Quadrotor, *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, USA, pp. 153-158, 2007.
- [8] Al-Younes, Y. M., Al-Jarrah, M. A. and Jhemi, A. A. Linear vs. Nonlinear Control Techniques for a Quadrotor Vehicle, *Proceedings of the 7th International Symposium on Mechatronics and its Applications (ISMA10)*, Sharjah, UAE, April 20-22, pp. ISMA10-1:10, 2010.
- [9] Adir, V. G., Stoica, A. M., Integral LQR Control of a Star-Shaped Octorotor, *Incas Bulletin*, vol. 4:2, pp. 3-18, 2012.
- [10] Tang, Y. R. and Li, Y. Design of an Optimal Flight Control System with Integral Augmented Compensator for a Nonlinear UAV Helicopter, *Proceedings of the 10th World Congress on Intelligent Control and Automation*, China, pp. 3927-3932, 2012.

-
- [11] Tommaso Francesco Villa, F. S., *Development and Validation of a UAV Based System for Air Pollution Measurements*. Sensors, 1-15, 2016.
- [12] Óscar Alvear, N. R., *Using UAV-Based Systems to Monitor Air Pollution in UAV-Based Systems to Monitor Air Pollution in*. Journal of Advanced Transportation, .1-14. 2017.
- [13] MOHAN, S., *U A Self-Tuning Proportional-Integral-Derivative Controller for an Autonomous Underwater*. Journal of Computer Science. 2010.
- [14] Myint, C., and Nu Nu Win. *Position and Velocity control for Two-Wheel Differential Drive Mobile Robot*. International Journal of Science, Engineering and Technology Research (IJSETR), 2849-2855. 2016.
- [15] Kumar, A. R. *Robust PID Tuning of Autonomous Underwater Vehicle Using Harmonic Search Algorithm Based on Model Order Reduction*. International Journal of Swarm Intelligence and Evolutionary Computation, 1-6. 2015.
- [16] Asokan, M. S. *A Self-Tuning Proportional-Integral-Derivative Controller for an Autonomous Underwater Vehicle, Based On Taguchi Method*. Journal of Computer Science, 862-871. 2010.
- [17] D.Andrea, M. H. *Performance benchmarking of quadrotor systems using time-optimal control*. Auton Robot, 70-86. 2012.
- [18] J. Sánchez-López, *"Toward visual autonomous ship board landing of a vtol uav"*. International Conference on Unmanned Aircraft Systems (ICUAS), 779-788. 2013.
- [19] J. Soroush Vahid, Kaveh Javanmard, *Modeling and Control of Autonomous Underwater Vehicle (AUV) In Heading and Depth Attitude via PPD Controller with State Feedback*. International Journal Of Coastal And Offshore Engineering (IJCOE),No.4, 11-18. 2016.
- [20] J. Visioli A., "Practical PID Control". Springer,ISBN.9781846285851. 2016
- [21] <https://ardupilot.org/copter/docs/common-telemetry-xbee.html> , mayo 2023
- [22] <https://ardupilot.org/copter/docs/common-rplidar-a2.html> , mayo 2023
- [23] <https://nanonets.com/blog/optical-flow/>
- [24] <https://www.sigmaelectronica.net/comunicacion-serial-raspberry-pi/>
- [25] <https://howtomechatronics.com/tutorials/arduino/arduino-and-mpu6050-accelerometer-and-gyroscope-tutorial/>

- [26] J. Angus P. Andrews Mohinder S. Grewal, -*Kalman Filtering Theory and Practice Using MATLAB*-. WILEY AND SONS, 2008.
- [27] Min Xu, Runhuai Yang, Yong Cheng, Hongcheng Xu "Kinematics Modeling and Step Climbing Study of an All-Terrain Wheeled Mobile Robot on Uneven Terrains". International Conference on Electronic and Mechanical Engineering and Information Technology, IEEE, pag 2725-2728. 2011.
- [28] Shivesh Kumar, Raghavendra S, Mihir Bhagat, K V Gangadharan §Design, simulation and testing of shrimp rover using Recurdyn . Department of Robotics and Automation, Faculty of Manufacturing Engineering, Universiti Teknikal Malaysia Melaka, Journal of Automation and Control Engineering, pág 1-8. 2013.
- [29] Ahmad Y. B. Hashim -On Graph Representation and Ground Surface Profiles while a Shrimp-Designed Robot is Traveling on Unexpected Landscapes- . International Conference on Electronic and Mechanical Engineering and Information Technology, Vol. 2, Num. 3, pag 209-2013. 2014.
- [30] H. Goldstein, *Classical Mechanics*, Adison Wesley Series in Physics, Adison-Wesley, U.S.A., Second Edition, 1980.
- [31] García Pérez O. *Seguimiento óptimo de trayectorias con un cuadricóptero*. Tesis de maestría, Maestría en Ciencias en Sistemas Autónomos de Navegación Aérea y Submarina, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional (CINVESTAV), 2015.
- [32] Peter Dyer et Stephen R. McReynolds. *The Computation and Theory of Optimal Control*. ELSERVIER, 1970.
- [33] Fernando Reyes Cortés. *Matlab: aplicado a robótica y mecatrónica*. Alpha Editorial, 2012.
- [34] Nonami, K., Kendoul, F., Suzuki, S., Wang, W., & Nakazawa, D. *Autonomous Flying Robots: Unmanned Aerial Vehicles and Micro Aerial Vehicles*. Springer Science & Business Media, 2010.
- [35] Rajamani, R. *Vehicle Dynamics and Control*. Springer Science & Business Media, 2011.
- [36] Zaknich, A. *Principles of Adaptive Filters and Self-Learning Systems*. Springer Science & Business Media, 2005.
- [37] Wan, E. A., & Van Der Merwe, R. The unscented Kalman filter. *Kalman filtering and neural networks*, 221-280, 2001.

-
- [38] Bouabdallah, S. and Siegwart, R., *Full Control of a Quadrotor*, Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, USA, pp. 153-158, 2007.
- [39] Castillo, P., Lozano, R., and Dzul, A. E. *Modelling and control of mini-flying machines*. Springer Science and Business Media, 2005.
- [40] Alderete, T. S. *Simulator aero model implementation*. NASA Ames Research Center, Moffett Field, California, 21, 1995.
- [41] Etkin, B., and Reid, L. D. *Dynamics of flight: stability and control*. John Wiley and Sons, 1995.
- [42] Salazar-Cruz, S., Palomino, A., & Lozano, R. *Trajectory tracking for a four rotor mini-aircraft*. In Proceedings of the 44th IEEE Conference on Decision and Control (pp. 2505-2510). IEEE, December 2005.
- [43] Castillo, P., Lozano, R., & Dzul, A. *Stabilization of a mini-rotorcraft having four rotors*. In 2004 IEEE/RSJ international conference on intelligent robots and systems (IROS)(IEEE Cat. No. 04CH37566) (Vol. 3, pp. 2693-2698). IEEE Septembe, 2004.
- [44] Castillo, P., Dzul, A., & Lozano, R. *Real-time stabilization and tracking of a four-rotor mini rotorcraft*. IEEE Transactions on control systems technology, 12(4), 510-516, 2004.
- [45] Santos, O., Romero, H., Salazar, S., Garcia-Perez, O., & Lozano, R. *Optimized discrete control law for quadrotor stabilization: Experimental results*. Journal of Intelligent & Robotic Systems, 84, 67-81, 2016.
- [46] Castillo, P., García, P., Lozano, R., & Albertos, P. *Modelado y estabilización de un helicóptero con cuatro rotores*. Revista Iberoamericana de Automática e Informática Industrial RIAI, 4(1), 41-57, 2007.
- [47] Romero, H., Salazar, S., & Lozano, R. *Visual servoing applied to real-time stabilization of a multi-rotor UAV*. Robotica, 30(7), 1203-1212, 2012.
- [48] Ordaz, J., Salazar, S., Mondié, S., Romero, H., & Lozano, R. *Predictor-based position control of a quad-rotor with delays in GPS and vision measurements*. Journal of intelligent & robotic systems, 70, 13-26, 2013.
- [49] Romero, H., Salazar, S., Lozano, R., & Benosman, R. . *Fusion of optical flow and inertial sensors for four-rotor rotorcraft stabilization*. IFAC Proceedings Volumes, 40(15), 209-214, 2007.

- [50] Fernández-Caballero, A., Maria Belmonte, L., Morales, R., & Andres Somolinos, J. . *Generalized proportional integral control for an unmanned quadrotor system*. International Journal of Advanced Robotic Systems, 12(7), 85, 2015.
- [51] Giernacki, W., Goliski, J., Goliska, J., Espinoza-Fraire, T., & Rao, J. *Mathematical modeling of the coaxial quadrotor dynamics for its attitude and altitude control*. Energies, 14(5), 1232. 2021.
- [52] Islam, S., Liu, P. X., & El Saddik, A. *Robust control of four-rotor unmanned aerial vehicle with disturbance uncertainty*. IEEE Transactions on Industrial Electronics, 62(3), 1563-1571, 2014.
- [53] Rinaldi, M., Primatesta, S., & Guglieri, G. *A comparative study for control of quadrotor uavs*. Applied Sciences, 13(6), 3464, 2023.
- [54] Estier, T., Piguët, R., Eichhorn, R., & Siegwart, R. (2000, December). *Shrimp, a rover architecture for long range martian mission*. In Proceedings of the Sixth ESA Workshop on Advanced Space Technologies for Robotics and Automation (ASTRA2000) (pp. 5-7).
- [55] Yang, J., Dong, M., & Ye, J. (2017, December). *A literature review of the rocker-bogie suspension for the planetary rover*. In 2017 International Seminar on Artificial Intelligence, Networking and Information Technology (ANIT 2017) (pp. 137-142). Atlantis Press.
- [56] Hashim, A. Y. (2014). *On Graph Representation and Ground Surface Profiles while a Shrimp-Designed Robot is Traveling on Unexpected Landscapes*. Journal of Automation and Control Engineering, 2(3).
- [57] Shahnawaz, S., Azim, R. A., uz Zaman, U. K., & Asad, U. (2020, April). *Kinematic Analysis of Pitch Angle for the Platform of a Shrimp Rover in an Unstructured Terrain*. In 2020 6th International Conference on Control, Automation and Robotics (ICCAR) (pp. 530-535). IEEE.
- [58] Lopez-Franco, M., Sanchez, E. N., Alanis, A. Y., & Arana-Daniel, N. (2013, June). *Discrete-time decentralized inverse optimal neural control for a shrimp robot*. In 2013 American Control Conference (pp. 1183-1188). IEEE.
- [59] Dian, S., Liu, T., Liang, Y., Liang, M., & Zhen, W. (2011, August). *A novel shrimp rover-based mobile robot for monitoring tunnel power cables*. In 2011 IEEE International Conference on Mechatronics and Automation (pp. 887-892). IEEE.

- [60] Lamon, P., Krebs, A., Lauria, M., Siegwart, R., & Shooter, S. (2004, April). *Wheel torque control for a rough terrain rover*. In IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 (Vol. 5, pp. 4682-4687). IEEE.
- [61] Michaud, S., Schneider, A., Bertrand, R., Lamon, P., Siegwart, R., Winnendael, M. V., & Schiele, A. (2002). *SOLERO: Solar powered exploration rover*. In Proc. of The 7th ESA Workshop on Advanced Space Technologies for Robotics and Automation (ASTRA). ETH-Zürich.
- [62] Murambikar, R., Omase, V., Nayak, V., Patil, K., & Mahulkar, Y. (2019). *Design and fabrication of rocker bogie mechanism using solar energy*. HISTORY, 6(04).
- [63] Meghdari, A., Pishkenari, H. N., Gaskarimahalle, A. L., Mahboobi, S. H., & Karimi, R. (2005). *A novel approach for optimal design of a rover mechanism*. Journal of Intelligent and Robotic Systems, 44, 291-312.
- [64] Nia, H. T., Alemohammad, S. H., Bagheri, S., Khiabani, R. H., & Meghdari, A. (2006). *Design, dynamic analysis and optimization of a rover for rescue operations*. In RoboCup 2005: Robot Soccer World Cup IX 9 (pp. 290-300). Springer Berlin Heidelberg.
- [65] Lamon, P., & Siegwart, R. (2003). *3D-Odometry for rough terrain-Towards real 3D navigation*. In 2003 IEEE International Conference on Robotics and Automation (No. CONF).
- [66] Hashim, A. Y. B., Nordin, M. S., Muhammad, M. N., Jamaludin, Z., Najib, A. M., Kamely, M. A., ... & Hasib, H. (2013). *Shrimp Robot Locomotion Response Model to the Ground Surface Profile*. In Conference on Computer Science and Computational Mathematics (ICCSCM 2013) (p. 1).
- [67] Tavakoli Nia, H., Alemohammad, S. H., Bagheri, S., Hajiaghvae Khiabani, R., & Meghdari, A. (2005, January). *Rough Terrain Rovers Dynamics Analysis and Optimization*. In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (Vol. 47446, pp. 903-909).
- [68] Wen, G., Ma, C., Cheng, D., Jin, Q., Chen, Z., Yang, X., ... & Zhou, J. (2013). *A four-wheel-rhombus-arranged mobility system for a new lunar robotic rover*. International Journal of Advanced Robotic Systems, 10(10), 370.
- [69] SantosSánchez, O.; García, O.; Romero, H.; Salazar, S.; Lozano, R. *Finite horizon nonlinear optimal control for a quadrotor: Experimental results*. Optim. Control Appl. Methods 2021, 42, 54–80.

- [70] Godinez-Garrido, G., Santos-Sánchez, O. J., Romero-Trejo, H., and García-Pérez, O. *Discrete Integral Optimal Controller for Quadrotor Attitude Stabilization: Experimental Results*. *Applied Sciences*, 13(16), 9293 2023.
- [71] Zamani, A.; Etedali, S. *Robust output feedback-based neuro-fuzzy controller for seismically excited tall buildings with ATMD accounting for variations in the type of supporting soil*. *Soil Dyn. Earthq. Eng.* 2023, 164, 107614.
- [72] Zamani, A.; Etedali, S. *Seismic structural control using magneto-rheological dampers: A decentralized interval type-2 fractional-order fuzzy PID controller optimized based on energy concepts*. *ISA Trans.* 2023, 137, 288–302.
- [73] Sadegh, E.; Abbas-Ali, Z. *Semi-active control of nonlinear smart base-isolated structures using MR damper: sensitivity and reliability analyses*. *Smart Mater. Struct.* 2022, 31, 065021
- [74] Antonio-Toledo, M.E.; Sanchez, E.N.; Alanis, A.Y.; Flórez, J.A.; Perez-Cisneros, M.A. *Real-time integral backstepping with sliding mode control for a quadrotor UAV*. *IFAC-PapersOnLine* 2018, 51, 549–554.
- [75] Elkhateem, A.S.; Engin, S.N. *Robust LQR and LQR-PI control strategies based on adaptive weighting matrix selection for a UAV position and attitude tracking control*. *Alex. Eng. J.* 2021, 61, 6275–6292.
- [76] Al-Isawi, M.M.; Attiya, A.J.; Adoghe, J.O. *UAV Control Based on Dual LQR and Fuzzy-PID Controller*. *Al-Khwarizmi Eng. J.*, 2020, 16, 43–53.
- [77] Rao, J.; Li, B.; Zhang, Z.; Chen, D.; Giernacki, W. *Position Control of Quadrotor UAV Based on Cascade Fuzzy Neural Network*. *Energies*, 2022, 15, 1763.
- [78] Dyer, E.; Sirouspour, S.; Jafarinasab, M. *Energy Optimal Control Allocation in a Redundantly Actuated Omnidirectional UAV*. In Proceedings of the International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 5316–5322.
- [79] Ding, Y.; Cao, R.; Liang, S.; Qi, F.; Yang, Q.; Yan, W. *Density-based optimal UAV path planning for photovoltaic farm inspection in complex topography*. Chinese Control And Decision Conference. In Proceedings of the International Conference on Robotics and Automation (ICRA), Chengdu, China, 14–16 June 2020; pp. 3931–3936.
- [80] Bucolo, M.; Buscarino, A.; Fortuna, L.; Gagliano, S. Generalizing the Letov formula for the discrete-time case. *Int. J. Dyn. Control*, 2023, 11, 94–100.

Apéndice A

Estrategias, métodos y algoritmos

A.1. Estrategia de control subóptimo autosintonizable

Para el presente trabajo se plantea ajustar las ganancias de un controlador PID automáticamente utilizando el filtro de Kalman, usando la ganancia estimada del filtro de Kalman como una medida de la incertidumbre del estado del sistema. Con la finalidad de que a medida que la incertidumbre del estado aumenta, también incremente la ganancia del PID para corregir cualquier desviación en la variable de salida.

Una forma de relacionar matemáticamente la ganancia del filtro de Kalman con las ganancias del PID es utilizar la ganancia estimada del filtro de Kalman como una variable de realimentación para ajustar las ganancias del PID. Para hacer esto, se puede utilizar una función que convierta la ganancia estimada del filtro de Kalman en una ganancia correspondiente para el PID. Se propone que debido a las ventajas del método del gradiente descendente, esta función sea hallada mediante este proceso, a continuación se desglosa el algoritmo propuesto:

1.- Definir la ley de control en tiempo discreto con ganancias variantes:

En tiempo discreto, un sistema no lineal se puede expresar como:

$$x[k + 1] = f_d(x[k], u[k]), \quad (\text{A.1})$$

$$y[k] = h(x[k]), \quad (\text{A.2})$$

donde el estado $x[k] \in \mathbb{R}^n$, y el vector de control $u[k] \in \mathbb{R}^p$, f_d y h son funciones discretas. Considerando una referencia constante $y_R \in \mathbb{R}^p$, el control integral es una retroalimentación de

estado tal que:

$$y[k] \rightarrow y_R, k \rightarrow \infty, \quad (\text{A.3})$$

El error de seguimiento en tiempo discreto es $e[k] = y[k] - y_R$ y la acción integral se puede expresar como:

$$\sigma[k+1] = \sigma[k] + e[k] = \sigma[k] + (h(x[k]) - y_R), \quad (\text{A.4})$$

El sistema aumentado en tiempo discreto se puede expresar como:

$$x[k+1] = f_d(x[k], u[k]), \quad (\text{A.5})$$

$$\sigma[k+1] = \sigma[k] + h(x[k]) - y_R, \quad (\text{A.6})$$

Asuma que el sistema es linealizable alrededor de $x_{ss}, \bar{\sigma}, u_{ss}$ en tiempo discreto, obtenemos el sistema linealizado en tiempo discreto:

$$\xi[k+1] = \mathcal{A}_d \xi[k] + \mathcal{B}_d v[k], \quad (\text{A.7})$$

donde $\xi[k] = \begin{bmatrix} x[k] - x_{ss} \\ \sigma[k] - \bar{\sigma} \end{bmatrix}$, $v[k] = u[k] - u_{ss}$ y \mathcal{A}_d y \mathcal{B}_d son las matrices discretizadas equivalentes a las matrices \mathcal{A} y \mathcal{B} en el caso continuo.

Ahora, diseñamos una matriz de ganancia \mathcal{K}_d tal que el sistema en lazo cerrado $\xi[k+1] = (\mathcal{A}_d + \mathcal{B}_d \mathcal{K}_d(k)) \xi[k]$ sea estable. Particionamos $\mathcal{K}_d(k)$ como $[\mathcal{K}_{1d}(k) \mathcal{K}_{2d}(k)]$. Entonces, el control discreto se expresa como:

$$u[k] = \mathcal{K}_{1d}(x[k] - x_{ss}) + \mathcal{K}_{2d}(\sigma[k] - \bar{\sigma}) + u_{ss}, \quad (\text{A.8})$$

En tiempo discreto, el punto de equilibrio (x_{ss}, σ_{ss}) es estable y todas las soluciones que comienzan suficientemente cerca de él se acercan a medida que k tiende a infinito. Entonces, $y[k] - y_R \rightarrow 0$ a medida que $k \rightarrow \infty$. Siguiendo estas ideas, en este trabajo, la matriz $\mathcal{K}_d(k)$ se diseña mediante el enfoque de control óptimo.

$$y[k] = \mathcal{C}_d \xi[k], \quad (\text{A.9})$$

2.- Definir la función de costo:

La función de costo puede ser definida como:

$$J = \xi(N)^T H \xi(N) + \sum_{i=k+1}^{N-1} \left\{ \xi(i)^T Q \xi(i) + v(i)^T R v(i) \right\} + \xi(k)^T Q \xi(k) + v(k)^T R v(k), \quad (\text{A.10})$$

donde $\xi(k)$ es el estado del sistema aumentado, $v(k)$ es la señal de control, k es el tiempo actual, N es el tiempo final, Q es la matriz de peso del estado, R es la matriz de peso de la señal de control y H es la matriz de peso del estado final.

3.- Inicializar las ganancias del controlador y la ganancia del filtro de Kalman:

Iniciar los valores de las ganancias del controlador y la ganancia del filtro de Kalman.

\mathcal{K}_{1d} , \mathcal{K}_{2d} y K_{kf0} ,

4.- Aplicar el Filtro de Kalman al sistema en lazo cerrado:

Predicción-extrapolación-estimación: a. Se calcula una predicción del estado $\xi(k+1)$, denotada como $\xi'(k+1)$, usando la ecuación:

$$\xi'(k+1) = \mathcal{A}_d \hat{\xi}(k) + \mathcal{B}_d v(k), \quad (\text{A.11})$$

b. Se predice el valor de la matriz de covarianza del error previo a la medida $P'(k+1)$, utilizando:

$$P'(k+1) = \mathcal{A}_d P(k) \mathcal{A}_d^T + Q_{kf}, \quad (\text{A.12})$$

Innovación-actualización-corrección: a. El valor del estado se actualiza con un factor de corrección que está en función del error, mediante:

$$\hat{\xi}(k+1) = \xi'(k+1) + K_{kf}(k+1)[y(k+1) - \mathcal{C}_d \xi'(k+1)], \quad (\text{A.13})$$

b. Se busca el valor $K_{kf}(k+1)$ para hallar un valor óptimo de $\hat{\xi}(k+1)$ tal que la matriz de covarianza del error sea mínima, utilizando:

$$K_{kf}(k+1) = \mathcal{C}_d P'(k+1) \cdot [\mathcal{C}_d P'(k+1) \mathcal{C}_d^T + R_{kf}(k+1)]^{-1}, \quad (\text{A.14})$$

c. Se busca el valor $P(k+1)$:

$$P(k+1) = [I - K_{fk}(k) \mathcal{C}_d] P'(k+1), \quad (\text{A.15})$$

donde: $\xi(k+1)$: El estado aumentado real del sistema en el instante de tiempo $k+1$.
 $\hat{\xi}(k)$: La estimación del estado aumentado del sistema en el instante de tiempo k .

$\xi'(k+1)$: La predicción del estado aumentado del sistema en el instante de tiempo $k+1$.

$v(k)$: La entrada de control aplicada al sistema en el instante de tiempo k .

$y(k+1)$: La medición del estado aumentado en el instante de tiempo $k+1$.

\mathcal{A}_d : La matriz de transición del estado que describe la relación entre el estado actual y el siguiente en función del modelo del sistema.

\mathcal{B}_d : La matriz de entrada que relaciona la entrada de control $u(k)$ con el estado del sistema.

\mathcal{C}_d : La matriz de observación que relaciona el estado aumentado del sistema con las mediciones.

$P(k)$: La matriz de covarianza del error de la estimación del estado en el instante de tiempo k .

$P'(k+1)$: La matriz de covarianza del error previo a la medida en el instante de tiempo $k+1$.

Q_{fk} : La matriz de covarianza del ruido del proceso, que representa la incertidumbre en el modelo del sistema.

$R_{fk}(k+1)$: La matriz de covarianza del ruido de la medición en el instante de tiempo $k+1$, que representa la incertidumbre en las mediciones.

$K_{kf}(k+1)$: La ganancia de Kalman en el instante de tiempo $k+1$, que determina la combinación óptima entre la predicción del estado y las mediciones para minimizar el error de estimación.

5.- Calcular el gradiente de la función de costo con respecto a las ganancias del controlador y la ganancia del filtro de Kalman:

$$\frac{\partial J}{\partial \mathcal{K}_{1d}} = 2 \sum_{i=k}^{N-1} e(i) \cdot \frac{\partial e(i)}{\partial \mathcal{K}_{1d}} \quad (\text{A.16})$$

$$\frac{\partial J}{\partial \mathcal{K}_{2d}} = 2 \sum_{i=k}^{N-1} e(i) \cdot \frac{\partial e(i)}{\partial \mathcal{K}_{2d}} \quad (\text{A.17})$$

$$\frac{\partial J}{\partial K_{kf}} = 2 \sum_{i=k}^{N-1} e(i) \cdot \frac{\partial e(i)}{\partial K_{kf}} \quad (\text{A.18})$$

donde $e(i)$ es el error en el tiempo i , y las derivadas parciales con respecto a las ganancias del controlador y la ganancia del filtro de Kalman.

6.- Actualizar las ganancias del controlador y la ganancia del filtro de Kalman utilizando el gradiente descendente:

Actualizar las ganancias del controlador y la ganancia del filtro de Kalman utilizando el método del gradiente descendente:

$$\mathcal{K}_{1d}(k+1) = \mathcal{K}_{1d}(k) - \varepsilon \frac{\partial J}{\partial \mathcal{K}_{1d}(k)}, \quad (\text{A.19})$$

$$\mathcal{K}_{2d}(k+1) = \mathcal{K}_{2d}(k) - \varepsilon \frac{\partial J}{\partial \mathcal{K}_{2d}(k)}, \quad (\text{A.20})$$

$$\mathcal{K}_{kf}(k+1) = \mathcal{K}_{kf}(k) - \varepsilon \frac{\partial J}{\partial \mathcal{K}_{kf}(k)}, \quad (\text{A.21})$$

donde k es el tiempo actual y ε es la tasa de aprendizaje, que debe ser fijada mínimo como el cuadrado del tiempo de muestreo, cuando dicho tiempo es menor a 1, para garantizar que el sistema convergerá, con ello se determina qué tan rápido se actualizan las ganancias.

7.- Aplicar la ley de control con ganancias actualizadas:

Finalmente, se aplica la ley de control con las ganancias actualizadas en el paso de tiempo k , y continuar el proceso recursivo de actualización de las ganancias y aplicación de la ley de control hasta alcanzar el tiempo final N .

$$v(k) = -\mathcal{K}_{1d}(k)x(k) - \mathcal{K}_{2d}(k)\hat{x}(k),$$

En este trabajo, esta estrategia de control, es usada para manipular cada uno de los actuadores de tracción del ULV, permitiendo que cada rueda tenga un control autosintonizable, lo que da la ventaja de poder desarrollar trayectorias en terrenos irregulares.

A.2. Estrategia de control aplicado al UAV

Para iniciar el proceso de síntesis de la estrategia de control a aplicar en este trabajo, se realiza la asignación de variables de estado sobre el modelo matemático del minihelicóptero descrito en la ecuación (4.32) dentro del Capítulo 2. Esta asignación de variables está descrita de la siguiente manera:

$$\begin{array}{l} x_1 = x \\ x_2 = \dot{x} \\ x_3 = y \\ x_4 = \dot{y} \\ x_5 = z \\ x_6 = \dot{z} \\ x_7 = \phi \\ x_8 = \dot{\phi} \\ x_9 = \theta \\ x_{10} = \dot{\theta} \\ x_{11} = \psi \\ x_{12} = \dot{\psi} \end{array} \Rightarrow \left[\begin{array}{l} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\frac{u}{m} \text{sen}(x_9) \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = \frac{u}{m} \cos(x_9) \text{sen}(x_7) \\ \dot{x}_5 = x_6 \\ \dot{x}_6 = \frac{u}{m} \cos(x_9) \cos(x_7) - g \\ \dot{x}_7 = x_8 \\ \dot{x}_8 = \tau_\phi \\ \dot{x}_9 = x_{10} \\ \dot{x}_{10} = \tau_\theta \\ \dot{x}_{11} = x_{12} \\ \dot{x}_{12} = \tau_\psi \end{array} \right], \quad (\text{A.22})$$

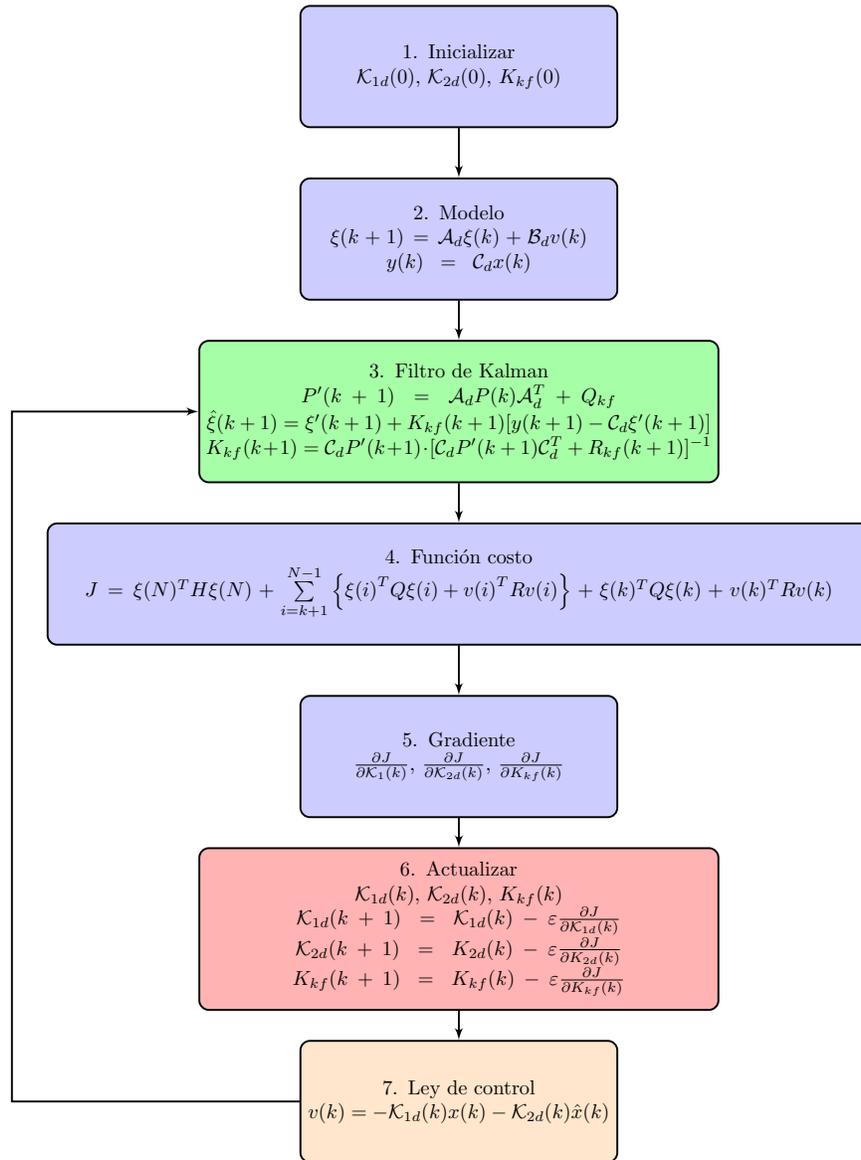


Figura A.1: Algoritmo propuesto para controlador subóptimo adaptable.

Al utilizar esta asignación de variables de estado (A.22), evidentemente se obtiene la ecuación de estado siguiente:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \\ \dot{x}_{10} \\ \dot{x}_{11} \\ \dot{x}_{12} \end{bmatrix} = \begin{bmatrix} x_2 \\ 0 \\ x_4 \\ 0 \\ x_6 \\ -g \\ x_8 \\ 0 \\ x_{10} \\ 0 \\ x_{12} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ -\frac{\text{sen}(x_9)}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{\cos(x_9)\text{sen}(x_7)}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{\cos(x_9)\cos(x_7)}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix}, \quad (\text{A.23})$$

A partir de la ecuación de estado (A.23) se realiza la síntesis de la ley de control óptima con acción integral. Como anteriormente se mencionó, la estrategia consiste en dividir el sistema en 4 subsistemas los cuales están definidos de la siguiente forma:

- Subsistema z .

$$\begin{aligned} \dot{x}_5 &= x_6, \\ \dot{x}_6 &= \frac{u}{m} \cos(x_9) \cos(x_7) - g, \end{aligned} \quad (\text{A.24})$$

- Subsistema ψ .

$$\begin{aligned} \dot{x}_{11} &= x_{12}, \\ \dot{x}_{12} &= \tau_\psi, \end{aligned} \quad (\text{A.25})$$

- Subsistema y - ϕ .

$$\begin{aligned} \dot{x}_3 &= x_4, \\ \dot{x}_4 &= \frac{u}{m} \cos(x_9) \text{sen}(x_7), \\ \dot{x}_7 &= x_8, \\ \dot{x}_8 &= \tau_\phi, \end{aligned} \quad (\text{A.26})$$

- Subsistema $x-\theta$.

$$\begin{aligned}\dot{x}_1 &= x_2, \\ \dot{x}_2 &= -\frac{u}{m} \text{sen}(x_9), \\ \dot{x}_9 &= x_{10}, \\ \dot{x}_{10} &= \tau_\theta,\end{aligned}\tag{A.27}$$

A partir del trabajo presentado en [4], se propone la aplicación de una ley de control $u(t)$ que permite realizar una linealización por retroalimentación de estado en el subsistema (A.24). Este control linealizante está definido como:

$$u = \frac{m(u_1 + g)}{(\cos(\theta) \cos(\phi))},\tag{A.28}$$

Entonces lo que se busca primeramente es sintetizar las leyes de control τ_θ y τ_ϕ , que sean lo suficientemente robustas y de dinámica rápida de tal forma que las condiciones $\tan(\theta) \approx \theta$ y $\tan(\phi) \approx \phi$ se cumplan para un tiempo relativamente pequeños. Suponiendo, que las condiciones anteriores se cumplen, entonces cada uno de los subsistemas descritos en (A.24)-(A.27) pueden ser representados de la siguiente manera:

$$\begin{bmatrix} \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}_{A_z} \begin{bmatrix} x_5 \\ x_6 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{B_z} [u_1],\tag{A.29}$$

$$\begin{bmatrix} \dot{x}_{11} \\ \dot{x}_{12} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}_{A_\psi} \begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{B_\psi} [\tau_\psi],\tag{A.30}$$

$$\begin{bmatrix} \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_7 \\ \dot{x}_8 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & g & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{A_{y\phi}} \begin{bmatrix} x_3 \\ x_4 \\ x_7 \\ x_8 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}}_{B_{y\phi}} [\tau_\phi],\tag{A.31}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_9 \\ \dot{x}_{10} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -g & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{A_{x\theta}} \begin{bmatrix} x_1 \\ x_2 \\ x_9 \\ x_{10} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}}_{B_{x\theta}} [\tau_\theta], \quad (\text{A.32})$$

Es claro que los subsistemas presentados anteriormente son lineales y con representación genérica indicada a continuación respectivamente:

$$\begin{aligned} \dot{x}_z &= A_z x_z + B_z u_z, \\ \dot{x}_\psi &= A_\psi x_\psi + B_\psi \tau_\psi, \\ \dot{x}_{y\phi} &= A_{y\phi} x_{y\phi} + B_{y\phi} \tau_\phi, \\ \dot{x}_{x\theta} &= A_{x\theta} x_{x\theta} + B_{x\theta} \tau_\theta, \end{aligned} \quad (\text{A.33})$$

Ahora bien, retomando la ecuación (A.28) se define a la señal u_1 como la ley de control óptimo. Para obtener la ley de control óptima se debe resolver de forma numérica una Ecuación Algebraica de tipo Riccati, este control está definido como:

$$u_1 = -R_z^{-1} B_z^T \mathbb{P}_z x_z(t), \quad (\text{A.34})$$

donde u_1 está sujeto al índice de desempeño cuadrático J_z dado por:

$$J_z = \int_0^\infty (x_z^T Q_z x_z + u_1^2 R_z) dt, \quad (\text{A.35})$$

El control óptimo τ_ψ para la regulación del ángulo de guiñada ψ del cuadricóptero, está definido de la manera siguiente:

$$\tau_\psi = -R_\psi^{-1} B_\psi^T \mathbb{P}_\psi x_\psi(t), \quad (\text{A.36})$$

el cual está sujeto al índice de desempeño J_ψ definido por:

$$J_\psi = \int_0^\infty (x_\psi^T Q_\psi x_\psi + \tau_\psi^2 R_\psi) dt, \quad (\text{A.37})$$

Para el caso del control óptimo τ_ϕ dedicado a regular el ángulo de alabeo ϕ del UAV, se encuentra definido como:

$$\tau_{y\phi} = -R_{y\phi}^{-1} B_{y\phi}^T \mathbb{P}_{y\phi} x_{y\phi}(t), \quad (\text{A.38})$$

con índice de desempeño $J_{y\phi}$ asociado definido por:

$$J_{y\phi} = \int_0^{\infty} (x_{y\phi}^T Q_{y\phi} x_{y\phi} + \tau_{y\phi}^2 R_{y\phi}) dt, \quad (\text{A.39})$$

Finalmente la señal de control óptimo relacionado con el ángulo de cabeceo θ que está denotado por τ_θ se encuentra definido de la forma siguiente:

$$\tau_{x\theta} = -R_{x\theta}^{-1} B_{x\theta}^T \mathbb{P}_{x\theta} x_{x\theta}(t), \quad (\text{A.40})$$

y su respectivo índice de desempeño $J_{x\theta}$ es:

$$J_{x\theta} = \int_0^{\infty} (x_{x\theta}^T Q_{x\theta} x_{x\theta} + \tau_{x\theta}^2 R_{x\theta}) dt, \quad (\text{A.41})$$

A.2.1. Modelo dinámico discretizado UAV

En razón de que en la presente tesis se desarrollan validaciones experimentales de la estrategia de control propuesta para la estabilización del cuadricóptero, es importante considerar que estas validaciones involucran la utilización y aplicación de dispositivos de cómputo (microcontrolador). En función de esta reciente condición, se hace necesario encontrar la representación discreta tanto del modelo como de las leyes de control hasta ahora sintetizadas. Para tal fin se emplea la aproximación de Euler sobre los subsistemas definidos en la ecuación (A.33) con tiempo de muestreo T , obteniéndose lo siguiente:

- Subsistema $z(k)$.

$$\begin{aligned} x_5(k+1) &= Tx_6(k) + x_5(k), \\ x_6(k+1) &= T \left[\frac{u(k)}{m} \cos(x_9(k)) \cos(x_7(k)) - g \right] + x_6(k), \end{aligned} \quad (\text{A.42})$$

- Subsistema $\psi(k)$.

$$\begin{aligned} x_{11}(k+1) &= Tx_{12}(k) + x_{11}(k), \\ x_{12}(k+1) &= T\tau_\psi(k) + x_{12}(k), \end{aligned} \quad (\text{A.43})$$

- Subsistema $y\text{-}\phi(k)$.

$$\begin{aligned} x_3(k+1) &= Tx_4(k) + x_3(k), \\ x_4(k+1) &= T \left[\frac{u(k)}{m} \cos(x_9(k)) \text{sen}(x_7(k)) \right] + x_4(k), \\ x_7(k+1) &= Tx_8(k) + x_7(k), \\ x_8(k+1) &= T\tau_\phi(k) + x_8(k), \end{aligned} \quad (\text{A.44})$$

- Subsistema $x\text{-}\theta(k)$.

$$\begin{aligned} x_1(k+1) &= Tx_2(k) + x_1(k), \\ x_2(k+1) &= T \left[-\frac{u(k)}{m} \text{sen}(x_9(k)) \right] + x_2(k), \\ x_9(k+1) &= Tx_{10}(k) + x_9(k), \\ x_{10}(k+1) &= T\tau_\theta(k) + x_{10}(k), \end{aligned} \quad (\text{A.45})$$

Para eliminar las no linealidades se aplica un proceso análogo al aplicado para el caso continuo. Entonces, realizando una linealización exacta con la ley de control $u(k)$ propuesta, queda de la siguiente forma:

$$u(k) = m(u_1(k) + g)(\cos(x_9(k))\cos(x_7(k)))^{-1}, \quad (\text{A.46})$$

Lo cual causa que los subsistemas discretos queden definidos de la forma que a continuación se presenta:

$$\begin{bmatrix} x_5(k+1) \\ x_6(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}}_{A_z(k)} \begin{bmatrix} x_5(k) \\ x_6(k) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ T \end{bmatrix}}_{B_z(k)} [u_1(k)], \quad (\text{A.47})$$

$$\begin{bmatrix} x_{11}(k+1) \\ x_{12}(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}}_{A_{\psi(k)}} \begin{bmatrix} x_{11}(k) \\ x_{12}(k) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ T \end{bmatrix}}_{B_{\psi(k)}} [\tau_{\psi}(k)], \quad (\text{A.48})$$

$$\begin{bmatrix} x_3(k+1) \\ x_4(k+1) \\ x_7(k+1) \\ x_8(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & gT & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{A_{y\phi(k)}} \begin{bmatrix} x_3(k) \\ x_4(k) \\ x_7(k) \\ x_8(k) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ T \end{bmatrix}}_{B_{y\phi(k)}} [\tau_{\phi}(k)], \quad (\text{A.49})$$

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_9(k+1) \\ x_{10}(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & -gT & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{A_{x\theta(k)}} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_9(k) \\ x_{10}(k) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ T \end{bmatrix}}_{B_{x\theta(k)}} [\tau_{\theta}(k)], \quad (\text{A.50})$$

Para que la linealización anterior de los subsistemas sea válida debe garantizarse que $\tau_{\theta}(k)$ es tal que $\tan(\theta(k)) \approx \theta(k)$ y que $\tau_{\phi}(k)$ es lo suficientemente robusta para que $\tan(\phi(k)) \approx \phi(k)$. Entonces si estas condiciones son satisfechas, los cuatro subsistemas linealizados están dados en la ecuación como a continuación:

$$\begin{aligned} x_z(k+1) &= A_z x_z(k) + B_z u_1(k), \\ x_{\psi}(k+1) &= A_{\psi} x_{\psi}(k) + B_{\psi} \tau_{\psi}(k), \\ x_{y\phi}(k+1) &= A_{y\phi} x_{y\phi}(k) + B_{y\phi} \tau_{\phi}(k), \\ x_{x\theta}(k+1) &= A_{x\theta} x_{x\theta}(k) + B_{x\theta} \tau_{\theta}(k). \end{aligned} \quad (\text{A.51})$$

A.2.2. Control integral aplicado al modelo dinámico discreto del UAV

De lo abordado en el Apéndice A, se sabe que para considerar la acción integral, se define una nueva variable de estado $\dot{\sigma}$, la cual define el error de estado de la siguiente forma:

$$\dot{\sigma}(t)e(t) = x(t) - x_R(t), \quad (\text{A.52})$$

Donde $e(t)$ representa el error, $x(t)$ la variable de proceso y $x_R(t)$ la referencia fijada para dicha variable. Retomando T como el tiempo de muestreo y aplicando la aproximación de Euler se puede decir que:

$$\dot{\sigma}(t) \approx \frac{\sigma(k+1) - \sigma(k)}{T}, \quad (\text{A.53})$$

Sustituyendo la ecuación (A.53) en (A.52), se tiene:

$$\frac{\sigma(k+1) - \sigma(k)}{T} = e(k), \quad (\text{A.54})$$

$$\sigma(k+1) = \sigma(k) + Te(k), \quad (\text{A.55})$$

Sustituyendo el error (A.52) en la ecuación anterior, ésta puede ser reescrita de la siguiente manera:

$$\sigma(k+1) = \sigma(k) + Tx(k) - Tx_R(t), \quad (\text{A.56})$$

A continuación se procede a aplicar el algoritmo del control integral a cada uno de los subsistemas discretos (A.47)-(A.50) que representan al UAV. Para el subsistema relacionado con la altura ($z(k)$) del vehículo aéreo, se consideran las siguientes definiciones: $x_5(k)x_{1z}(k)$ y $x_6(k)x_{2z}(k)$, a partir de las definiciones anteriores dicho subsistema puede reescribirse como:

$$\begin{bmatrix} x_{1z}(k+1) \\ x_{2z}(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}}_{A_{z(k)}} \begin{bmatrix} x_{1z}(k) \\ x_{2z}(k) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ T \end{bmatrix}}_{B_{z(k)}} [u_1(k)], \quad (\text{A.57})$$

Continuando con el proceso de síntesis de la acción integral se define $e_{1z}(k)x_{1z}(k) - x_{1zR}(k)$ y $e_{2z}(k)x_{2z}(k) - x_{2zR}(k)$ como los errores en los estados $x_{1z}(k)$ y $x_{2z}(k)$, donde $x_{1zR}(k)$ y $x_{2zR}(k)$ son las referencia dadas para $x_{1z}(k)$, $x_{2z}(k)$, respectivamente. Considerando $\sigma_{1z}(k+1) = e_{1z}(k)$ y $\sigma_{2z}(k+1) = e_{2z}(k)$ el vector del sistema aumentado para el subsistema $z(k)$ queda entonces definido por:

$$\xi_{z(k)}(k) = \begin{bmatrix} x_{1z}(k) \\ x_{2z}(k) \\ \sigma_{1z}(k) \\ \sigma_{2z}(k) \end{bmatrix}, \quad (\text{A.58})$$

Consecuentemente, la representación espacio estado para el sistema aumentado en $z(k)$ es:

$$\xi_{z(k)}(k+1) = \begin{bmatrix} x_{1z}(k+1) \\ x_{2z}(k+1) \\ \sigma_{1z}(k+1) \\ \sigma_{2z}(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ T & 0 & 1 & 0 \\ 0 & T & 0 & 1 \end{bmatrix}}_{A_z} \begin{bmatrix} x_{1z}(k) \\ x_{2z}(k) \\ \sigma_{1z}(k) \\ \sigma_{2z}(k) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ T \\ 0 \\ 0 \end{bmatrix}}_{B_z} u_1(k) - \underbrace{\begin{bmatrix} 0 \\ 0 \\ x_{1zR}(k) \\ x_{2zR}(k) \end{bmatrix}}_{\mathcal{X}_{Zref}}, \quad (\text{A.59})$$

$$\xi_{z(k)}(k+1) = \begin{bmatrix} A_{z(k)} & 0 \\ TC_{z(k)} & I_2 \end{bmatrix} \xi_{z(k)}(k) + \begin{bmatrix} B_{z(k)} \\ 0 \end{bmatrix} u_1(k), \quad (\text{A.60})$$

Con respecto al subsistema relacionado con el ángulo de guiñada ψ , se tiene $x_{11}(k)x_{1\psi}(k)$ y $x_{12}(k)x_{2\psi}(k)$, por tanto el subsistema $\psi(k)$ está definido como:

$$\begin{bmatrix} x_{1\psi}(k+1) \\ x_{2\psi}(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}}_{A_{\psi(k)}} \begin{bmatrix} x_{1\psi}(k) \\ x_{2\psi}(k) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ T \end{bmatrix}}_{B_{\psi(k)}} [\tau_{\psi}(k)], \quad (\text{A.61})$$

donde de manera similar al estado $z(k)$ se tiene que $e_{1\psi}(k)x_{1\psi}(k) - x_{1\psi R}(k)$ y $e_{2\psi}(k)x_{2\psi}(k) - x_{2\psi R}(k)$ son los errores en los estados $x_{1\psi}(k)$ y $x_{2\psi}(k)$, con $x_{1\psi R}(k)$ y $x_{2\psi R}(k)$ definiendo sus respectivas referencias. Nuevamente asignando $\sigma_{1\psi}(k+1) = e_{1\psi}(k)$ y $\sigma_{2\psi}(k+1) = e_{2\psi}(k)$, el vector de estado extendido para el subsistema $\psi(k)$ queda definido como:

$$\xi_{\psi(k)}(k) = \begin{bmatrix} x_{1\psi}(k) \\ x_{2\psi}(k) \\ \sigma_{1\psi}(k) \\ \sigma_{2\psi}(k) \end{bmatrix}, \quad (\text{A.62})$$

Siendo su correspondiente representa en espacio de estado la siguiente:

$$\xi_{\psi(k)}(k+1) = \begin{bmatrix} x_{1\psi}(k+1) \\ x_{2\psi}(k+1) \\ \sigma_{1\psi}(k+1) \\ \sigma_{2\psi}(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ T & 0 & 1 & 0 \\ 0 & T & 0 & 1 \end{bmatrix}}_{\mathcal{A}_{\psi}} \begin{bmatrix} x_{1\psi}(k) \\ x_{2\psi}(k) \\ \sigma_{1\psi}(k) \\ \sigma_{2\psi}(k) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ T \\ 0 \\ 0 \end{bmatrix}}_{\mathcal{B}_{\psi}} \tau_{\psi}(k) - \underbrace{\begin{bmatrix} 0 \\ 0 \\ x_{1\psi R}(k) \\ x_{2\psi R}(k) \end{bmatrix}}_{\mathcal{X}_{\psi ref}}, \quad (\text{A.63})$$

$$\xi_{\psi(k)}(k+1) = \begin{bmatrix} A_{\psi(k)} & 0 \\ TC_{\psi(k)} & I_2 \end{bmatrix} \xi_{\psi(k)}(k) + \begin{bmatrix} B_{\psi(k)} \\ 0 \end{bmatrix} \tau_{\psi}(k), \quad (\text{A.64})$$

El procedimiento se aplica y desarrolla de manera análoga para los dos subsistemas restantes $y - \phi(k)$ y $x - \theta(k)$. Para el subsistema $y - \phi(k)$ se realizan las siguientes definiciones:

$$\begin{aligned} x_3(k) &\triangleq x_{1y}(k), \\ x_4(k) &\triangleq x_{2y}(k), \\ x_7(k) &\triangleq x_{3\phi}(k), \\ x_8(k) &\triangleq x_{4\phi}(k), \end{aligned} \quad (\text{A.65})$$

de tal manera que al ser aplicadas se llega a la siguiente representación en espacio de estado para este subsistema.

$$\begin{bmatrix} x_{1y}(k+1) \\ x_{2y}(k+1) \\ x_{3\phi}(k+1) \\ x_{4\phi}(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & gT & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{A_{y\phi(k)}} \begin{bmatrix} x_{1y}(k) \\ x_{2y}(k) \\ x_{3\phi}(k) \\ x_{4\phi}(k) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ T \end{bmatrix}}_{B_{y\phi(k)}} [\tau_{\phi}(k)], \quad (\text{A.66})$$

Definiendo ahora las variables del sistema aumentado y los errores, se tiene que:

$$\begin{aligned} \sigma_{1y}(k+1)e_{1y}(k) &= x_{1y}(k) - x_{1yR}(k), \\ \sigma_{2y}(k+1)e_{2y}(k) &= x_{2y}(k) - x_{2yR}(k), \\ \sigma_{3\phi}(k+1)e_{3\phi}(k) &= x_{3\phi}(k) - x_{3\phi R}(k), \\ \sigma_{4\phi}(k+1)e_{4\phi}(k) &= x_{4\phi}(k) - x_{4\phi R}(k), \end{aligned} \quad (\text{A.67})$$

Donde $x_{1yR}(k)$ es la referencia para $x_{1y}(k)$, $x_{2yR}(k)$ es la referencia para $x_{2y}(k)$, $x_{3\phi R}(k)$ es la referencia para $x_{3\phi}(k)$ y $x_{4\phi R}(k)$ es la referencia para $x_{4\phi}(k)$. El vector del subsistema $y - \phi(k)$ aumentado está dado por:

$$\xi_{y\phi}(k) = \begin{bmatrix} x_{1y}(k) \\ x_{2y}(k) \\ x_{3\phi}(k) \\ x_{4\phi}(k) \\ \sigma_{1y}(k) \\ \sigma_{2y}(k) \\ \sigma_{3\phi}(k) \\ \sigma_{4\phi}(k) \end{bmatrix}, \quad (\text{A.68})$$

lo que lleva a la siguiente ecuación de estado representando en forma discreta la dinámica del subsistema $y - \phi(k)$:

$$\xi_{y\phi}(k+1) = \underbrace{\begin{bmatrix} 1 & T & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & gT & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & T & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ T & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & T & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & T & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & T & 0 & 0 & 0 & 1 \end{bmatrix}}_{A_{y\phi}} \begin{bmatrix} x_{1y}(k) \\ x_{2y}(k) \\ x_{3\phi}(k) \\ x_{4\phi}(k) \\ \sigma_{1y}(k) \\ \sigma_{2y}(k) \\ \sigma_{3\phi}(k) \\ \sigma_{4\phi}(k) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ T \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{B_{y\phi}} \tau_{\phi}(k) - \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ x_{1yR}(k) \\ x_{2yR}(k) \\ x_{3\phi R}(k) \\ x_{4\phi R}(k) \end{bmatrix}}_{X_{y\phi ref}}, \quad (\text{A.69})$$

$$\xi_{y\phi}(k+1) = \begin{bmatrix} A_{y\phi}(k) & 0 \\ TC_{y\phi}(k) & I_4 \end{bmatrix} \xi_{y\phi}(k) + \begin{bmatrix} B_{y\phi}(k) \\ 0 \end{bmatrix} \tau_{\phi}(k), \quad (\text{A.70})$$

Finalmente para el subsistema $x - \theta(k)$ se proponen los siguientes asignaciones de variables:

$$\begin{aligned} x_1(k) &\triangleq x_{1x}(k), \\ x_2(k) &\triangleq x_{2x}(k), \\ x_9(k) &\triangleq x_{3\theta}(k), \\ x_{10}(k) &\triangleq x_{4\theta}(k). \end{aligned} \quad (\text{A.71})$$

Lo cual implica que la representación espacio estado para este subsistema está definida como:

$$\begin{bmatrix} x_{1x}(k+1) \\ x_{2x}(k+1) \\ x_{3\theta}(k+1) \\ x_{4\theta}(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & -gT & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{A_{x\theta(k)}} \begin{bmatrix} x_{1x}(k) \\ x_{2x}(k) \\ x_{3\theta}(k) \\ x_{4\theta}(k) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ T \end{bmatrix}}_{B_{x\theta(k)}} [\tau_\theta(k)], \quad (\text{A.72})$$

Las variables auxiliares y los errores para construir el sistema aumentado son:

$$\begin{aligned} \sigma_{1x}e_{1x}(k) &= x_{1x}(k) - x_{1xR}(k), \\ \sigma_{2x}e_{2x}(k) &= x_{2x}(k) - x_{2xR}(k), \\ \sigma_{3\theta}e_{3\theta}(k) &= x_{3\theta}(k) - x_{3\theta R}(k), \\ \sigma_{4\theta}e_{4\theta}(k) &= x_{4\theta}(k) - x_{4\theta R}(k). \end{aligned} \quad (\text{A.73})$$

Donde $x_{1xR}(k)$ es la referencia para $x_{1x}(k)$, $x_{2xR}(k)$ es la referencia para $x_{2x}(k)$, $x_{3\theta R}(k)$ es la referencia para $x_{3\theta}(k)$ y $x_{4\theta R}(k)$ es la referencia para $x_{4\theta}(k)$. El vector del subsistema $x - \theta(k)$ aumentado está dado por:

$$\xi_{x\theta}(k) = \begin{bmatrix} x_{1x}(k) \\ x_{2x}(k) \\ x_{3\theta}(k) \\ x_{4\theta}(k) \\ \sigma_{1x}(k) \\ \sigma_{2x}(k) \\ \sigma_{3\theta}(k) \\ \sigma_{4\theta}(k) \end{bmatrix}, \quad (\text{A.74})$$

Claramente se puede deducir que la ecuación de estado asociada a esta asignación y aumento en dimensión del vector de estado es la siguiente:

$$\xi_{x\theta(k)}(k+1) = \begin{bmatrix} x_{1x}(k+1) \\ x_{2x}(k+1) \\ x_{3\theta}(k+1) \\ x_{4\theta}(k+1) \\ \sigma_{1x}(k+1) \\ \sigma_{2x}(k+1) \\ \sigma_{3\theta}(k+1) \\ \sigma_{4\theta}(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & T & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -gT & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & T & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ T & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & T & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & T & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & T & 0 & 0 & 0 & 1 \end{bmatrix}}_{A_{x\theta}} \begin{bmatrix} x_{1x}(k) \\ x_{2x}(k) \\ x_{3\theta}(k) \\ x_{4\theta}(k) \\ \sigma_{1x}(k) \\ \sigma_{2x}(k) \\ \sigma_{3\theta}(k) \\ \sigma_{4\theta}(k) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ T \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{B_{x\theta}} \tau_{\theta}(k) - \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ x_{1xR}(k) \\ x_{2xR}(k) \\ x_{3\theta R}(k) \\ x_{4\theta R}(k) \end{bmatrix}}_{x_{x\theta ref}}, \quad (\text{A.75})$$

$$\xi_{x\theta(k)}(k+1) = \begin{bmatrix} A_{x\theta(k)} & 0 \\ TC_{x\theta(k)} & I_4 \end{bmatrix} \xi_{x\theta(k)}(k) + \begin{bmatrix} B_{x\theta(k)} \\ 0 \end{bmatrix} \tau_{\theta}(k), \quad (\text{A.76})$$

Los controles óptimos $u_1(k)$, $\tau_{\psi}(k)$, $\tau_{\phi}(k)$ y $\tau_{\theta}(k)$ han sido hallados utilizando la función e índice de desempeño asociado a cada uno de los subsistemas aumentados. Es claro que para que estos cuatro controles puedan ser aplicados es necesario verificar la controlabilidad de cada subsistema. Por lo tanto, se procede de la siguiente forma: si el par $A_z(k), B_z(k)$ es controlable y $\begin{bmatrix} A_z(k) & B_z(k) \\ C_z(k) & 0 \end{bmatrix}$, es de rango $2 + 1$, donde 2 es el número de variables de estado y 1 es el número de controles, entonces (A_z, B_z) es controlable. Suponiendo que esto se cumple, se puede calcular el control óptimo $u_1^*(k)$ está definido como:

$$u_1^*(k) = -(R_z + B_z^T P_z B_z)^{-1} B_z^T P_z A_z x_z^*(k), \quad (\text{A.77})$$

sujeto al índice de desempeño dado por la siguiente ecuación.

$$J_z = \sum_{k=1}^{\infty} (x_z^T(k) Q_z x_z(k) + u_1^2(k) R_z), \quad (\text{A.78})$$

donde $Q_z \geq 0$, $R_z > 0$ y P_z es la solución a la ecuación algebraica del tipo de Riccati asociada definida como:

$$P_z = A_z^T P_z A_z - A_z^T P_z B_z (R_z + B_z^T P_z B_z)^{-1} B_z^T P_z + Q_z, \quad (\text{A.79})$$

De forma similar si el par $A_{\psi}(k), B_{\psi}(k)$ es controlable y $\begin{bmatrix} A_{\psi}(k) & B_{\psi}(k) \\ C_{\psi}(k) & 0 \end{bmatrix}$, es de rango $2 + 1$, donde 2 es el número de variables de estado y 1 es el número de controles, para este

subsistema, entonces $(\mathcal{A}_\psi, \mathcal{B}_\psi)$ es controlable, si es así el control óptimo para el ángulo de guiñada ψ del UAV es:

$$\tau_\psi^*(k) = -(R_\psi + B_\psi^T P_\psi B_\psi)^{-1} B_\psi^T P_\psi A_\psi x_\psi^*(k), \quad (\text{A.80})$$

con índice de desempeño J_ψ definido por:

$$J_\psi = \sum_{k=1}^{\infty} (x_\psi^T(k) Q_\psi x_\psi(k) + \tau_\psi^2(k) R_\psi), \quad (\text{A.81})$$

Donde $Q_\psi \geq 0$, $R_\psi > 0$ y P_ψ es la solución a la ecuación algebraica del tipo de Riccati para este subsistema y definida como:

$$P_\psi = A_\psi^T P_\psi A_\psi - A_\psi^T P_\psi B_\psi (R_\psi + B_\psi^T P_\psi B_\psi)^{-1} B_\psi^T + Q_\psi, \quad (\text{A.82})$$

Trabajando ahora con el subsistema y - $\phi(k)$ se tiene que si el par $A_{y\phi}(k), B_{y\phi}(k)$ es controlable y $\begin{bmatrix} A_{y\phi}(k) & B_{y\phi}(k) \\ C_{y\phi}(k) & 0 \end{bmatrix}$, es de rango $4 + 1$, donde 4 es el número de variables de estado y 1 es el número de controles, entonces $(\mathcal{A}_{y\phi}, \mathcal{B}_{y\phi})$ es controlable, lo que implica que el control óptimo para el alabeo del UAV es $\tau_{y\phi}^*(k)$ es:

$$\tau_{y\phi}^*(k) = -(R_{y\phi} + B_{y\phi}^T P_{y\phi} B_{y\phi})^{-1} B_{y\phi}^T P_{y\phi} A_{y\phi} x_{y\phi}^*(k), \quad (\text{A.83})$$

Sujeto al índice de desempeño $J_{y\phi}$ mostrado en la siguiente ecuación:

$$J_{y\phi} = \sum_{k=1}^{\infty} (x_{y\phi}^T(k) Q_{y\phi} x_{y\phi}(k) + \tau_{y\phi}^2(k) R_{y\phi}), \quad (\text{A.84})$$

Donde $Q_{y\phi} \geq 0$, $R_{y\phi} > 0$ y $P_{y\phi}$ es la solución a la ecuación algebraica del tipo de Riccati, dada por:

$$P_{y\phi} = A_{y\phi}^T P_{y\phi} A_{y\phi} - A_{y\phi}^T P_{y\phi} B_{y\phi} (R_{y\phi} + B_{y\phi}^T P_{y\phi} B_{y\phi})^{-1} B_{y\phi}^T + Q_{y\phi}, \quad (\text{A.85})$$

Finalmente para el subsistema x - $\theta(k)$ de igual manera, si el par $A_{x\theta}(k), B_{x\theta}(k)$ es controlable y $\begin{bmatrix} A_{x\theta}(k) & B_{x\theta}(k) \\ C_{x\theta}(k) & 0 \end{bmatrix}$, es de rango $4 + 1$, donde 4 es el número de variables de estado y 1 es el número de controles, entonces $(\mathcal{A}_{x\theta}, \mathcal{B}_{x\theta})$ es controlable, lo que implica que el control óptimo para el ángulo de cabeceo θ del UAV es:

$$\tau_{x\theta}^*(k) = -(R_{x\theta} + B_{x\theta}^T P_{x\theta} B_{x\theta})^{-1} B_{x\theta}^T P_{x\theta} A_{x\theta} x_{x\theta}^*(k), \quad (\text{A.86})$$

con índice de desempeño $J_{x\theta}$ definido como:

$$J_{x\theta} = \sum_{k=1}^{\infty} (x_{x\theta}^T(k) Q_{x\theta} x_{x\theta}(k) + \tau_{x\theta}^2(k) R_{x\theta}), \quad (\text{A.87})$$

Donde $Q_{x\theta} \geq 0$, $R_{x\theta} > 0$ y $P_{x\theta}$ es la solución a la ecuación algebraica del tipo de Riccati dada por:

$$P_{x\theta} = A_{x\theta}^T P_{x\theta} A_{x\theta} - A_{x\theta}^T P_{x\theta} B_{x\theta} (R_{x\theta} + B_{x\theta}^T P_{x\theta} B_{x\theta})^{-1} B_{x\theta}^T P_{x\theta} + Q_{x\theta}. \quad (\text{A.88})$$

A.3. Algoritmos.

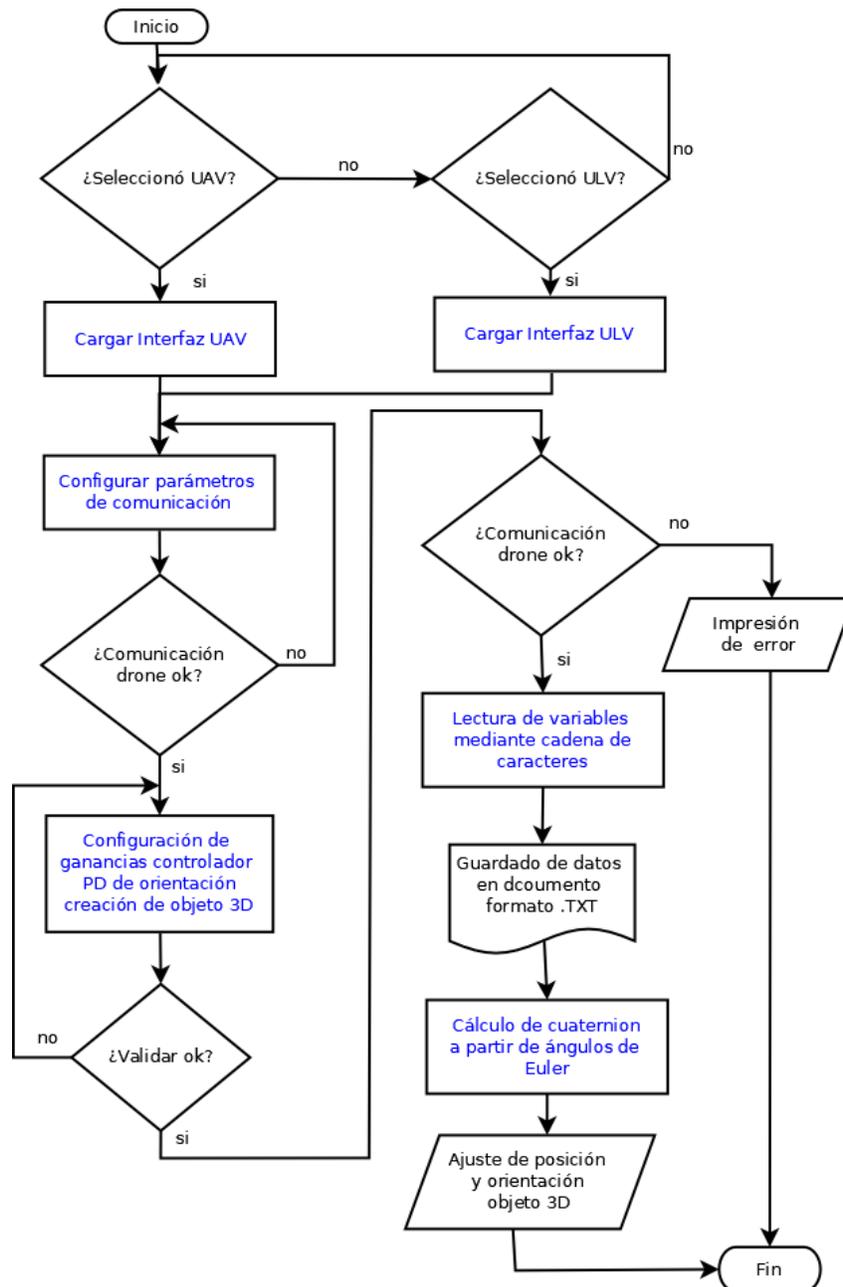


Figura A.2: Diagrama de flujo interfaz gráfica.

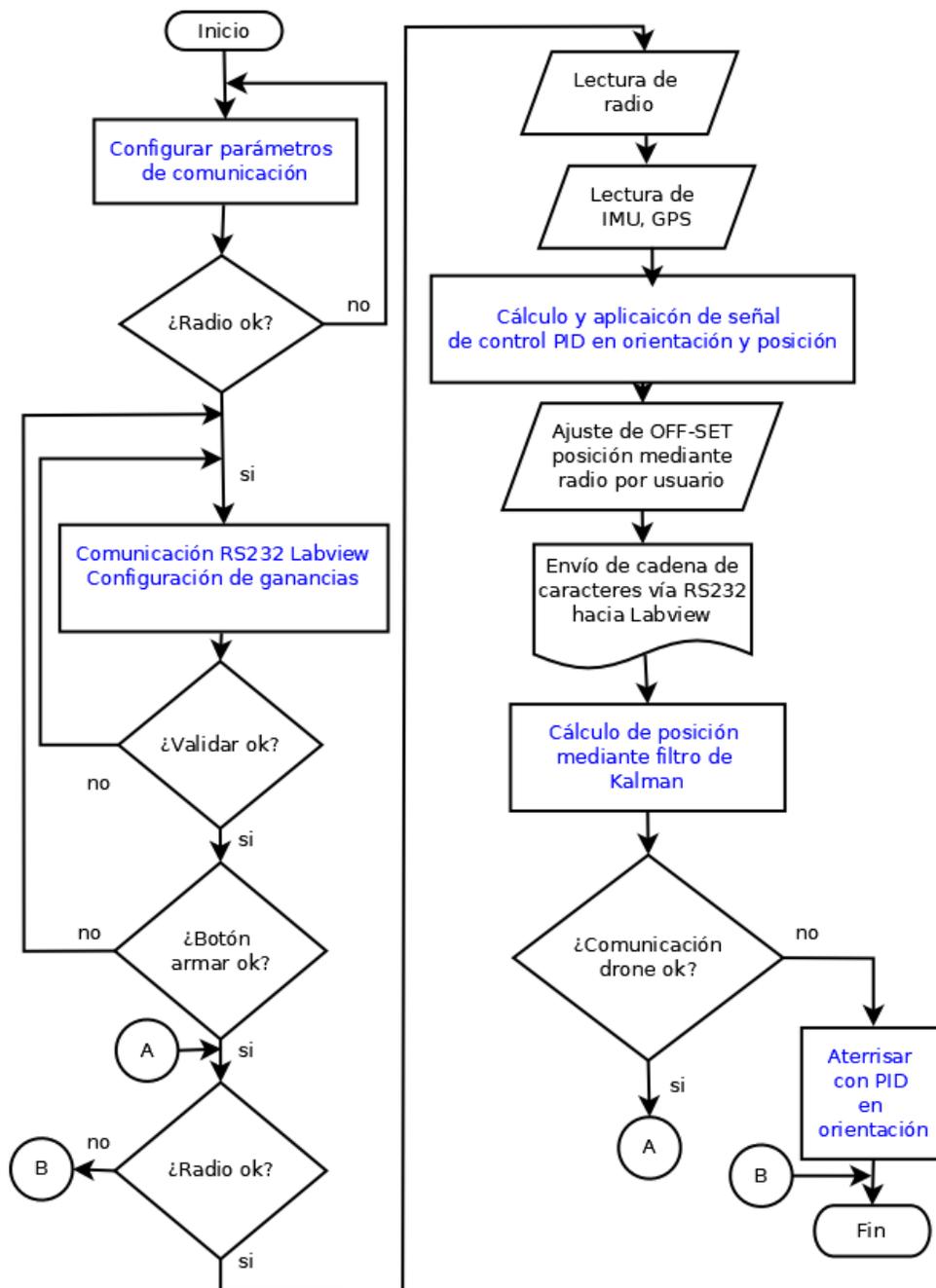


Figura A.3: Diagrama de flujo UAV.

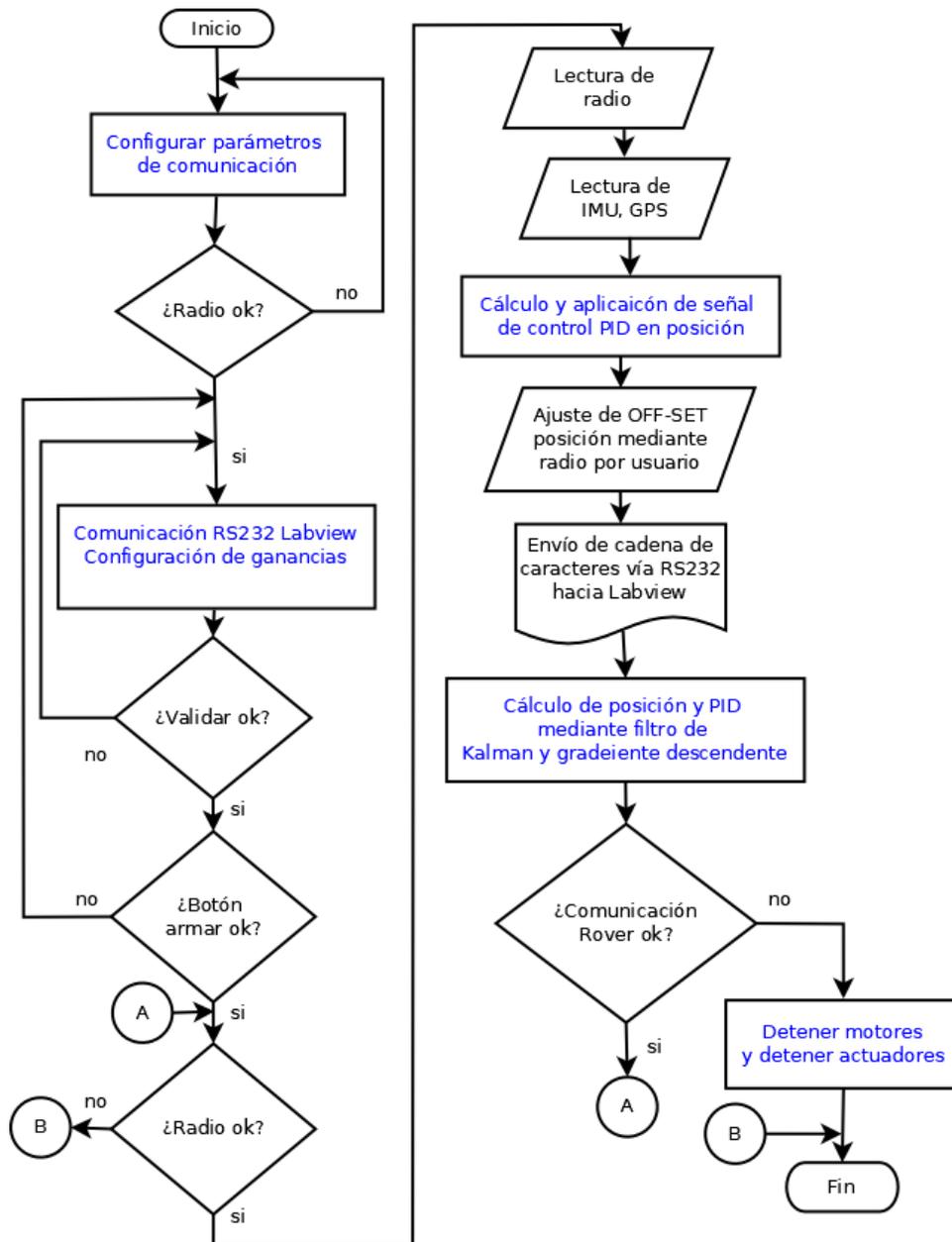


Figura A.4: Diagrama de flujo ULV.

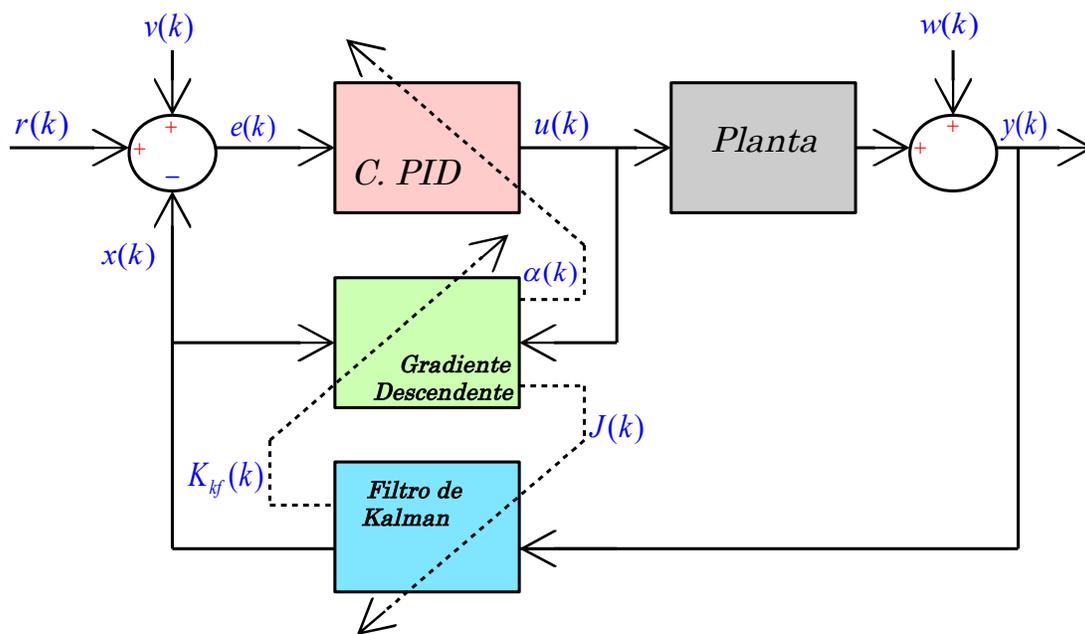


Figura A.5: Diagrama controlador PID híbrido autosintoniable.

Apéndice B

Planos y esquemas

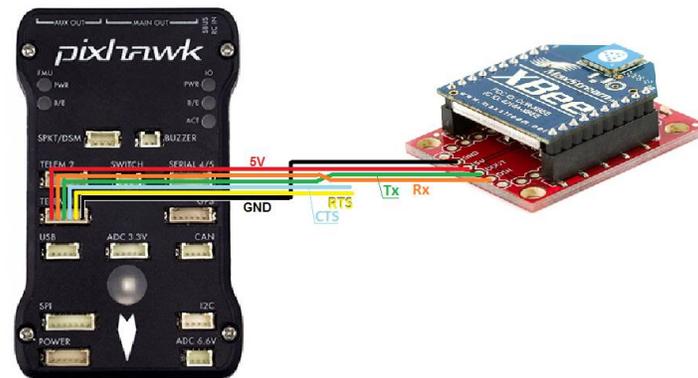


Figura B.1: XBee PIX-HAWK® pinout [21].



Figura B.2: RPlidar PIX-HAWK® pinout [22].

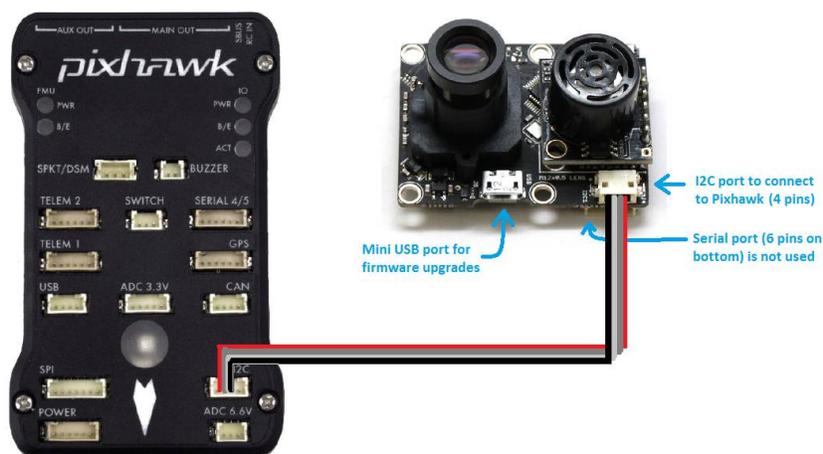


Figura B.3: Conexión sensor PX4-Flow y controlador PIX-HAWK® [22].

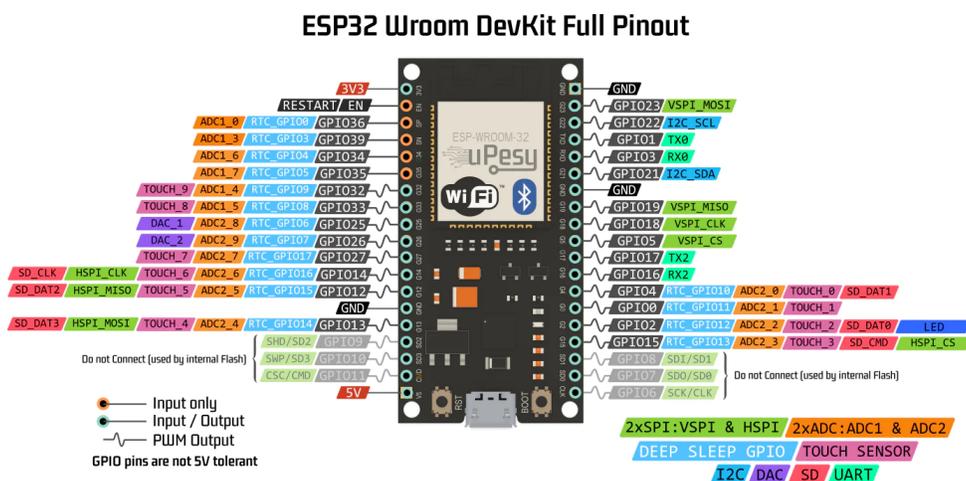


Figura B.4: ESP32 wrover pin-out.

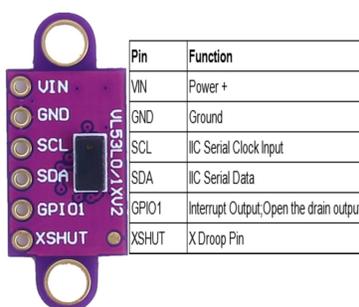


Figura B.5: Sensor de altura del carro transversal del rover.

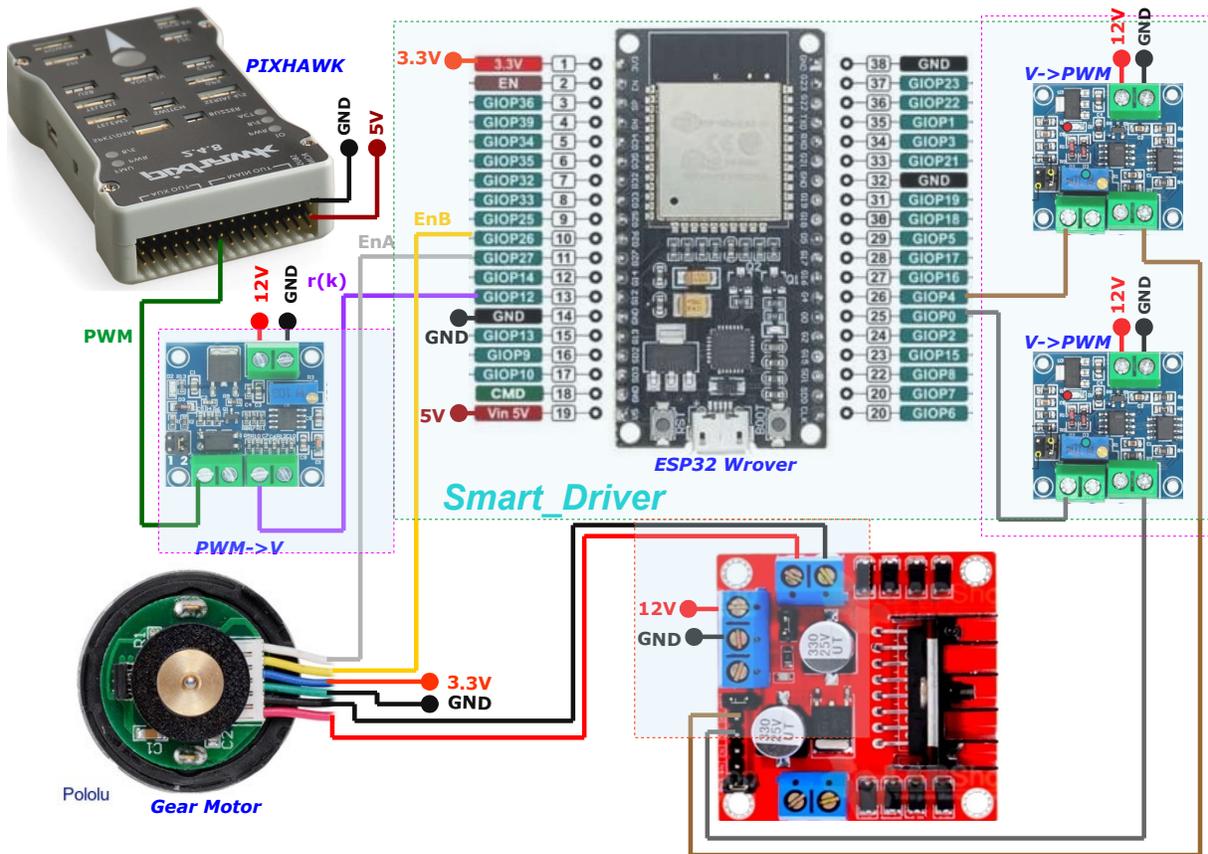


Figura B.6: Driver de los motores del ULV.

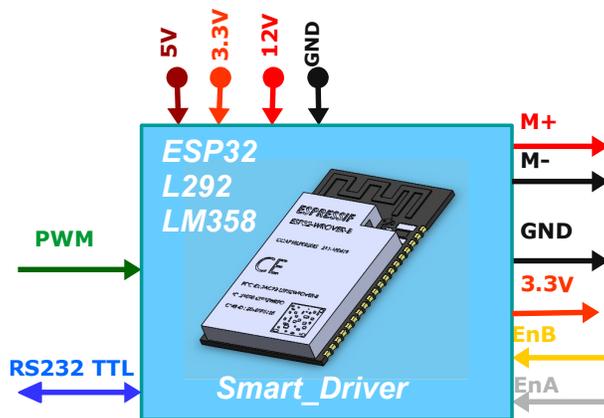


Figura B.7: Smart driver de los motores del ULV.

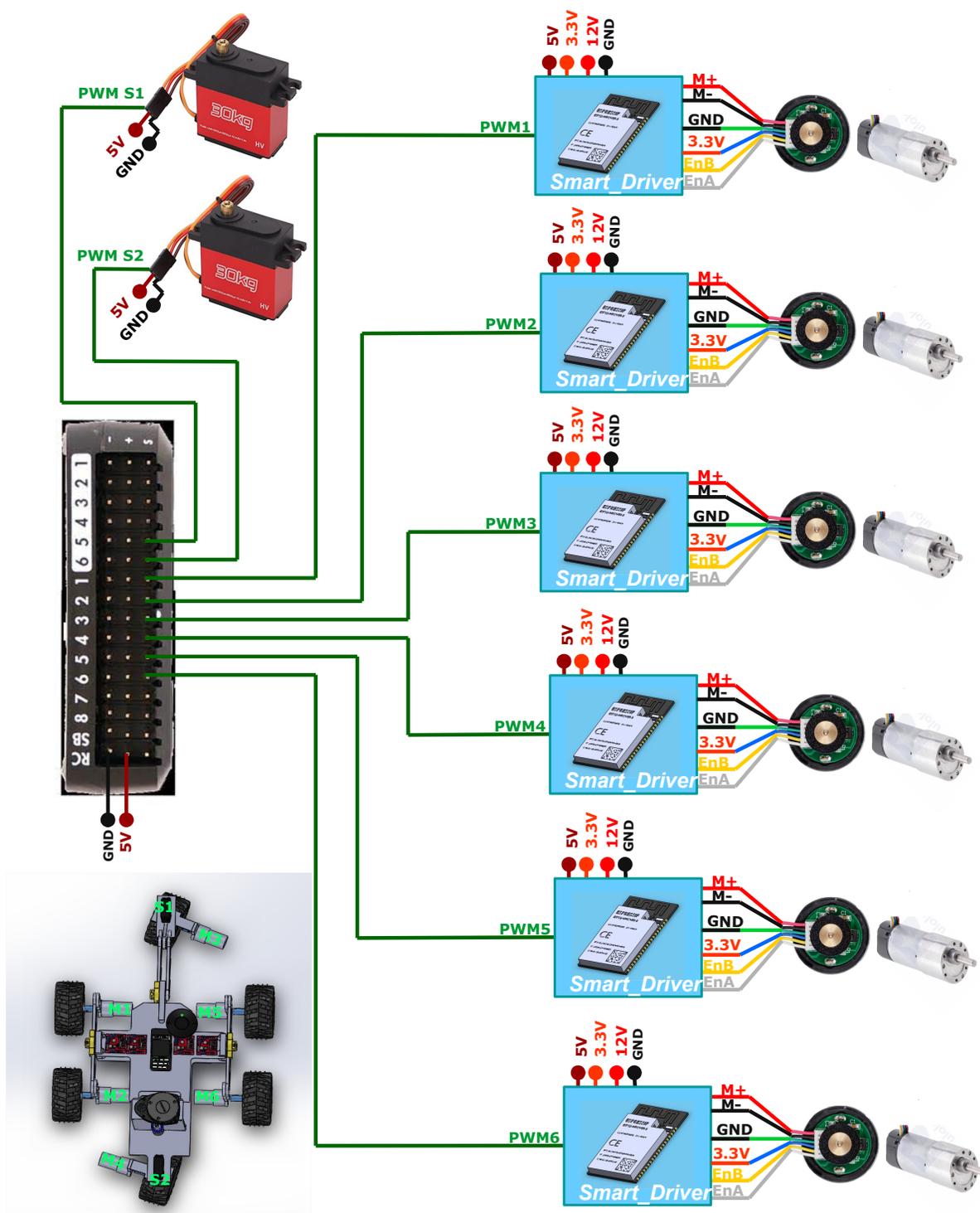
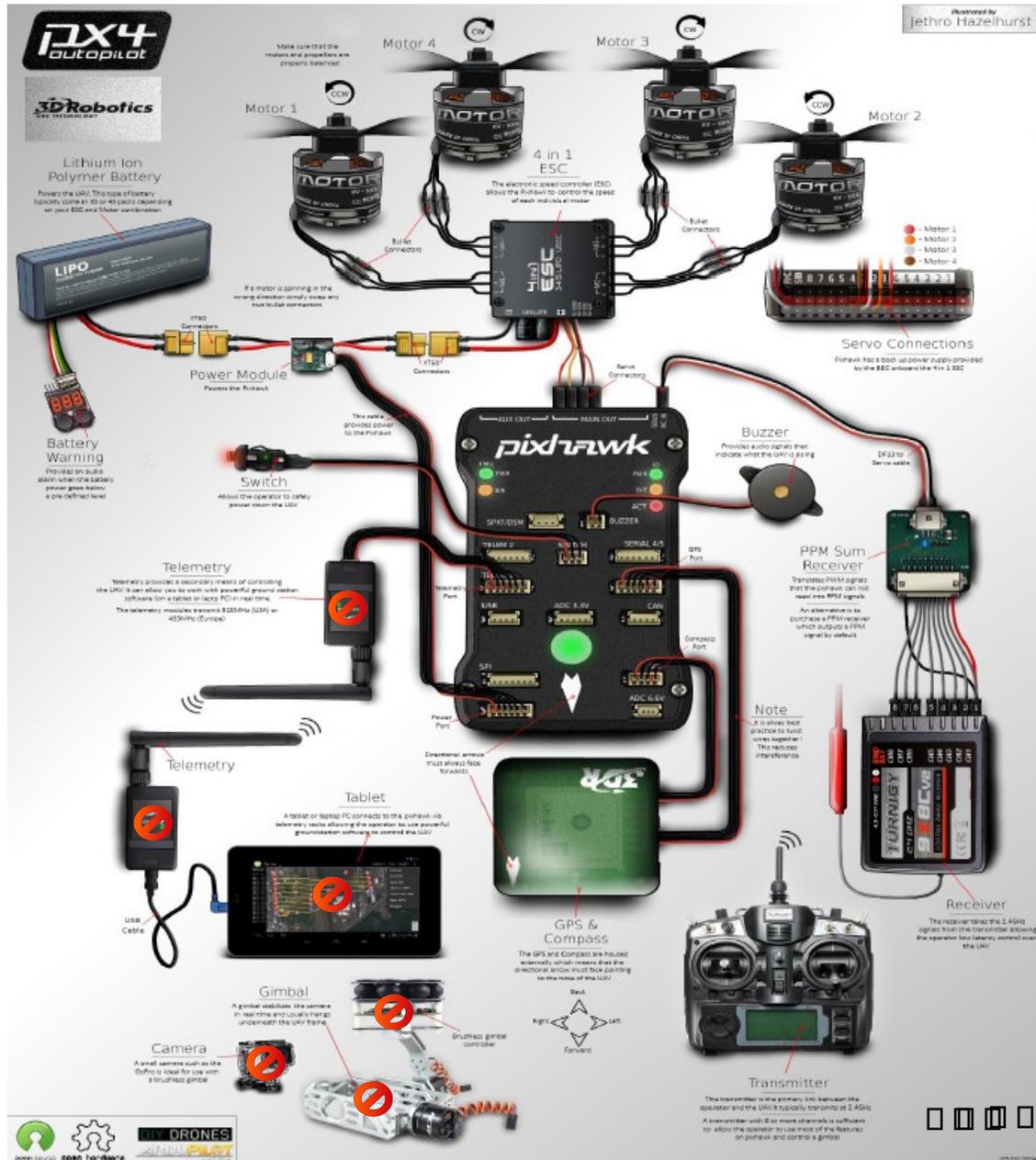


Figura B.8: Conexión PIX-HAKW® motores para motricidad del ULV.



⊘ Nota: Elemento no implementado en este trabajo

Figura B.9: PIX-HAWK® elementos y componentes [21].

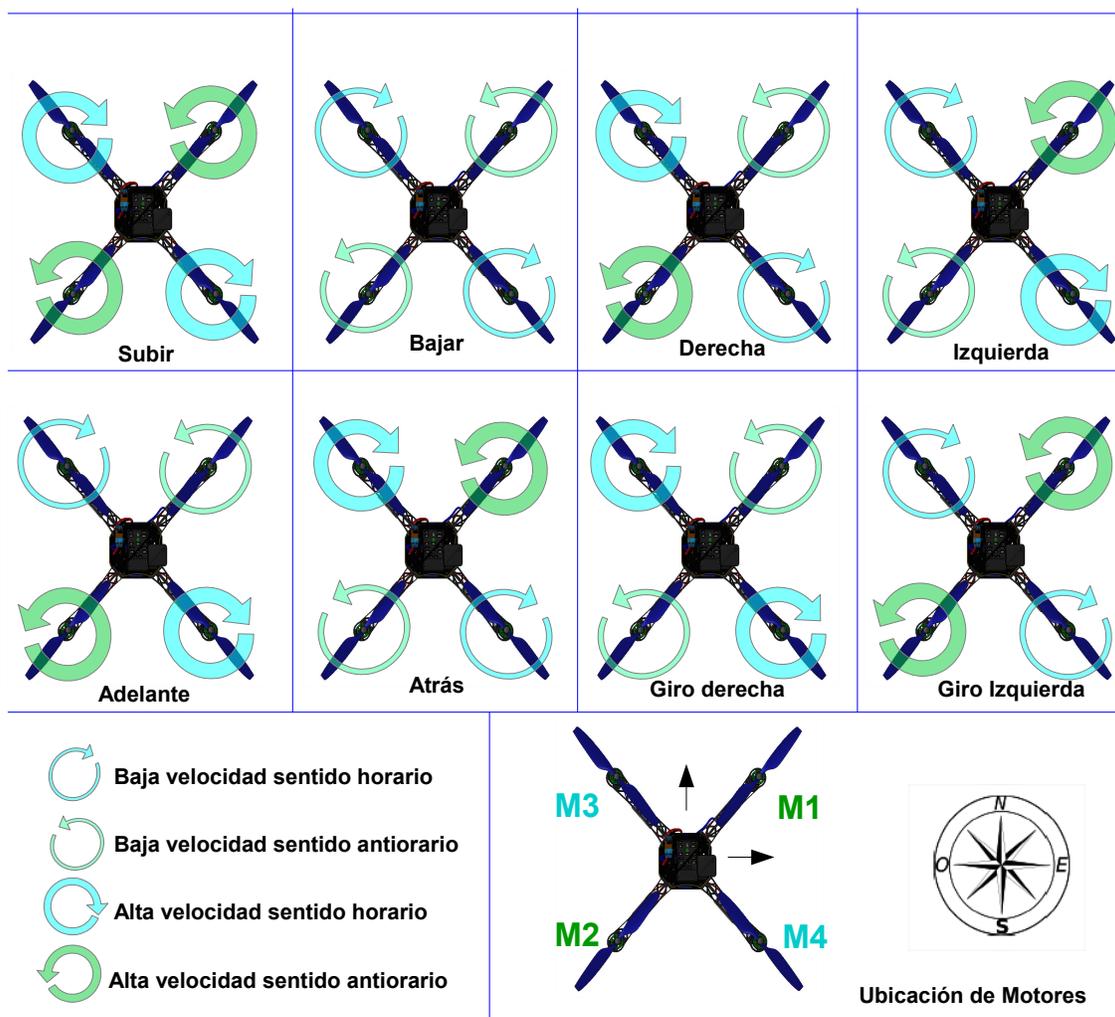


Figura B.10: Movimientos del UAV en función de las velocidades de sus motores.

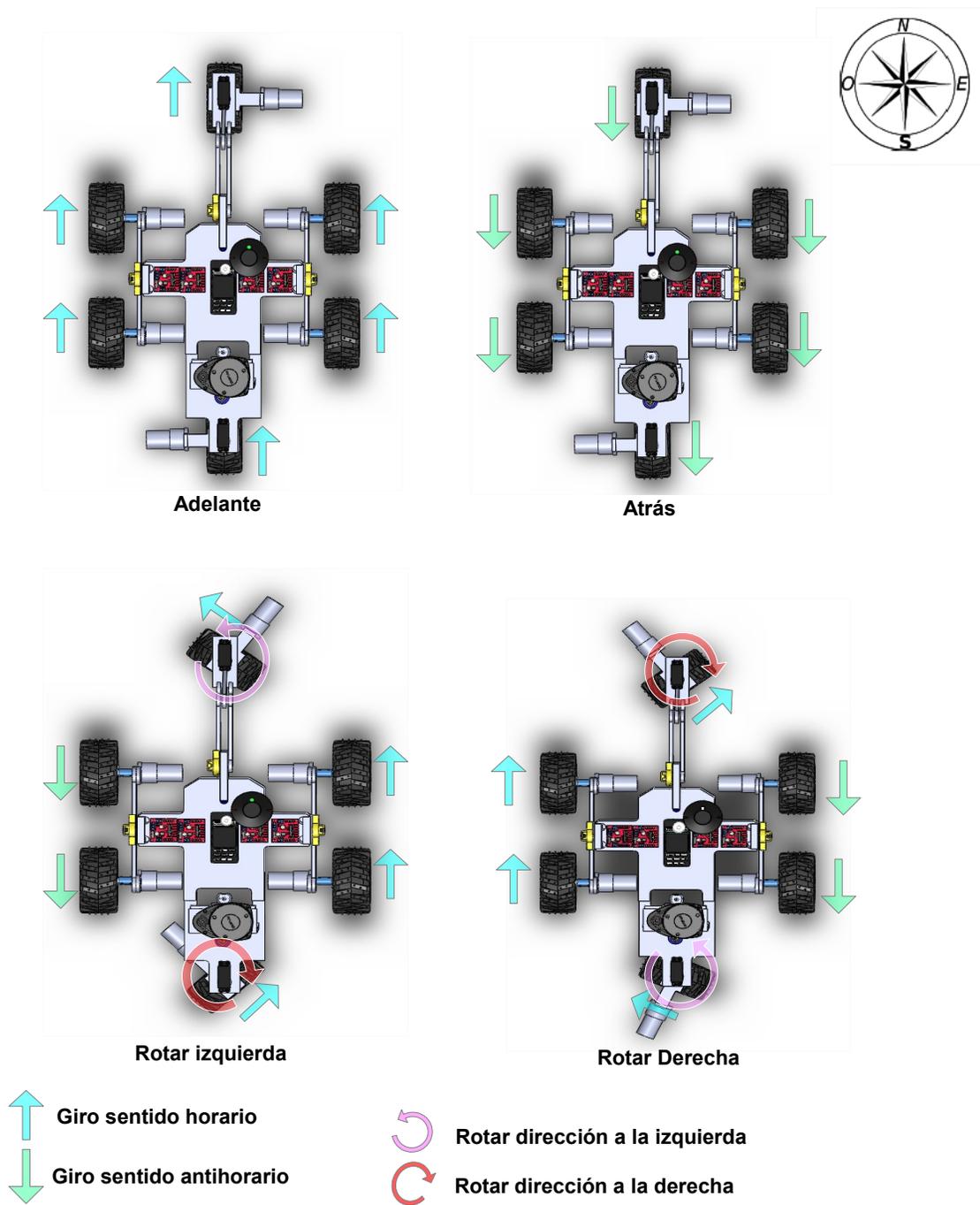


Figura B.11: Movimientos del ULV en función del giro de sus motores.

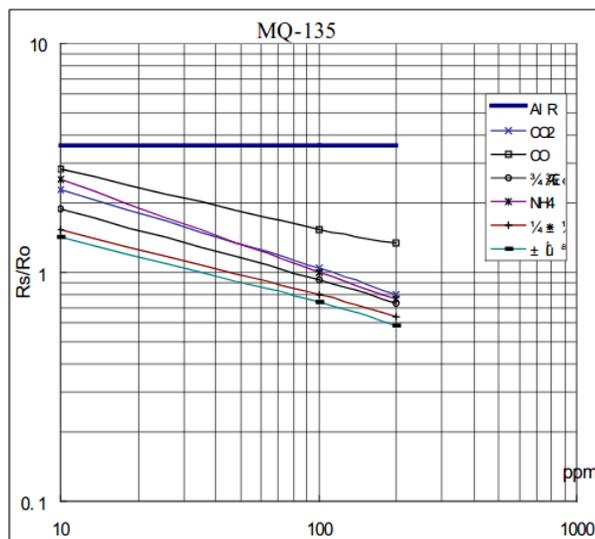


Fig.3 is shows the typical sensitivity characteristics of the MQ-135 for several gases. in their: Temp: 20 Humidity: 65% O_2 concentration 21% $R_L=20k\Omega$ R_o : sensor resistance at 100ppm of NH_3 in the clean air. R_s :sensor resistance at various concentrations of gases.

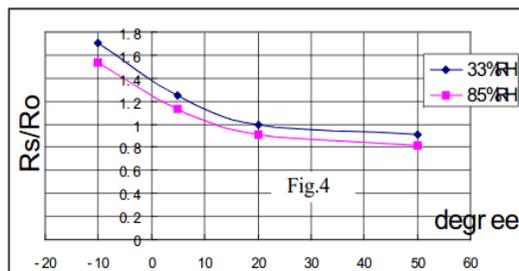


Fig.4 is shows the typical dependence of the MQ-135 on temperature and humidity. R_o : sensor resistance at 100ppm of NH_3 in air at 33%RH and 20 degree. R_s : sensor resistance at 100ppm of NH_3 at different temperatures and humidities.

Figura B.12: Curvas de comportamiento gases detectados por el sensor MQ135.

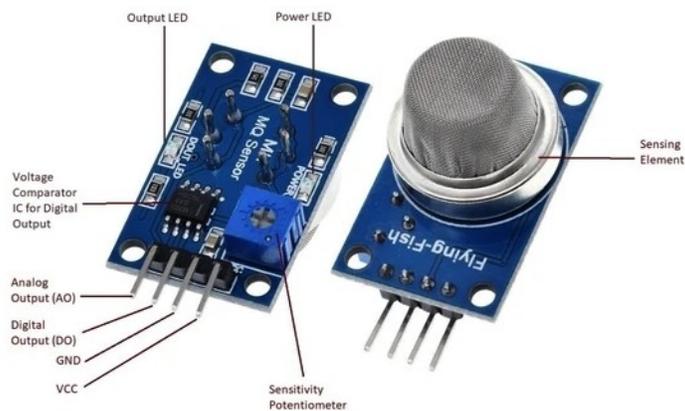


Figura B.13: Sensor MQ135.

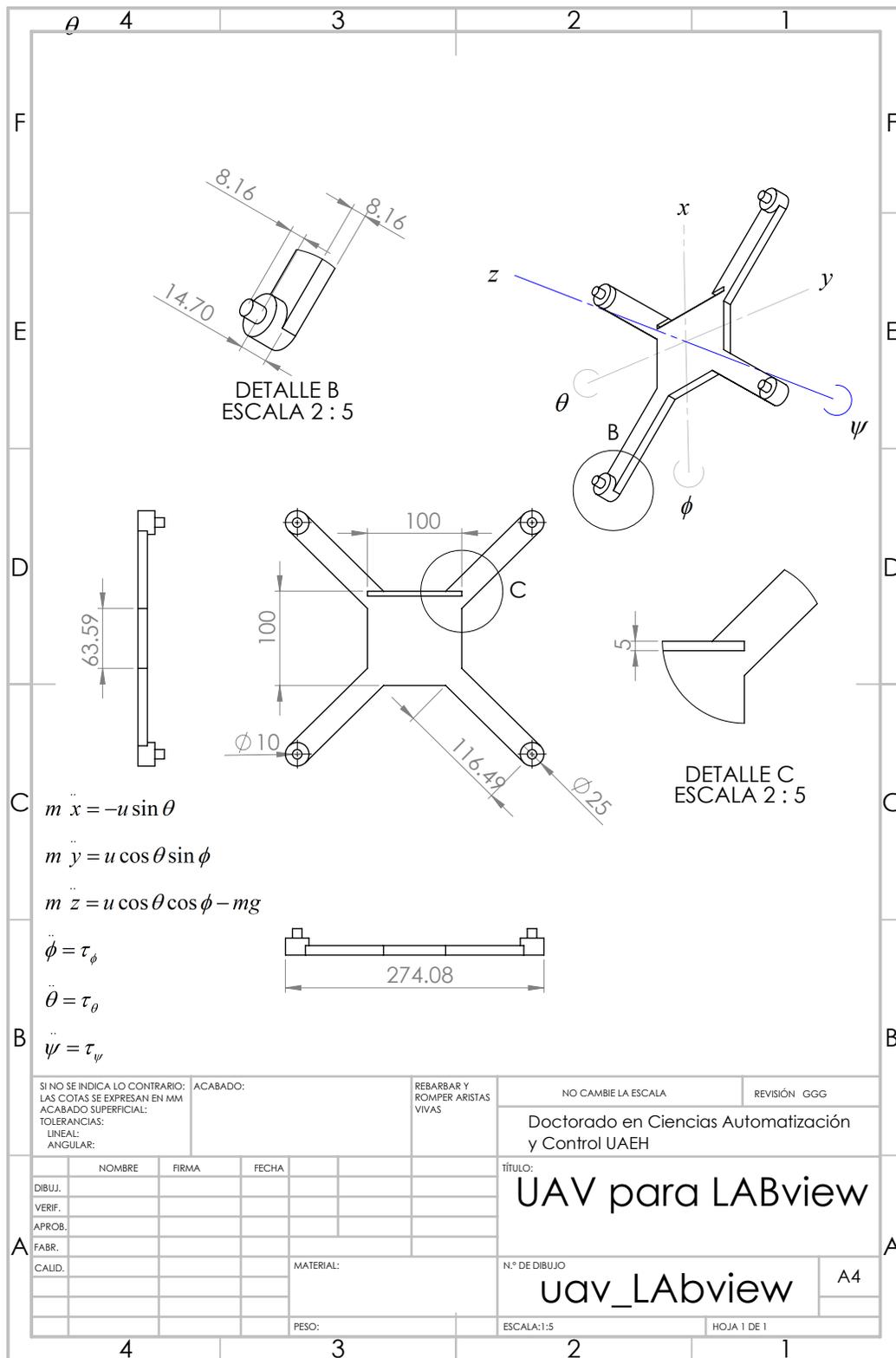


Figura B.14: Plano objeto 3D para LABview®.

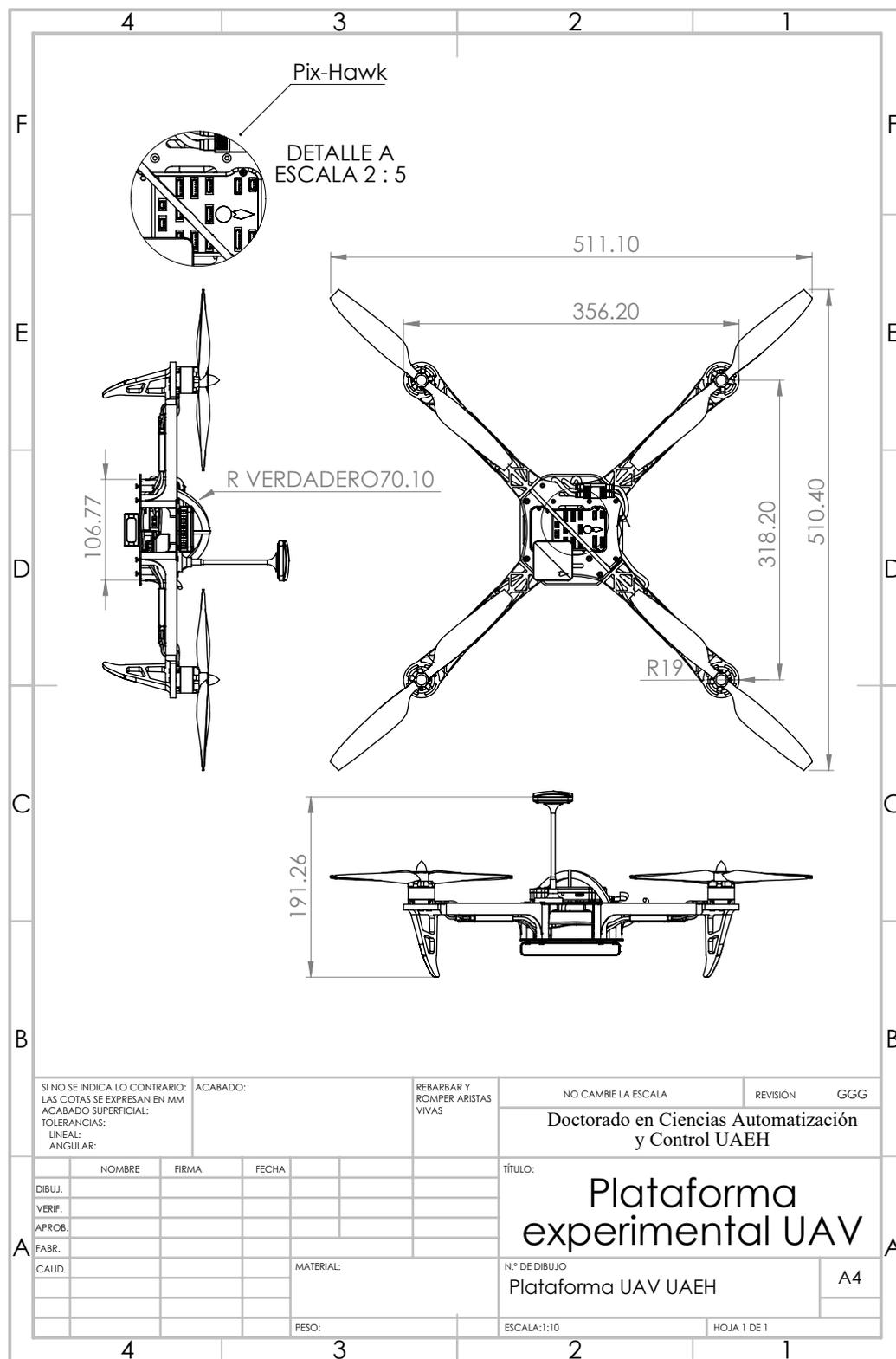


Figura B.15: Plataforma experimental UAV vistas.

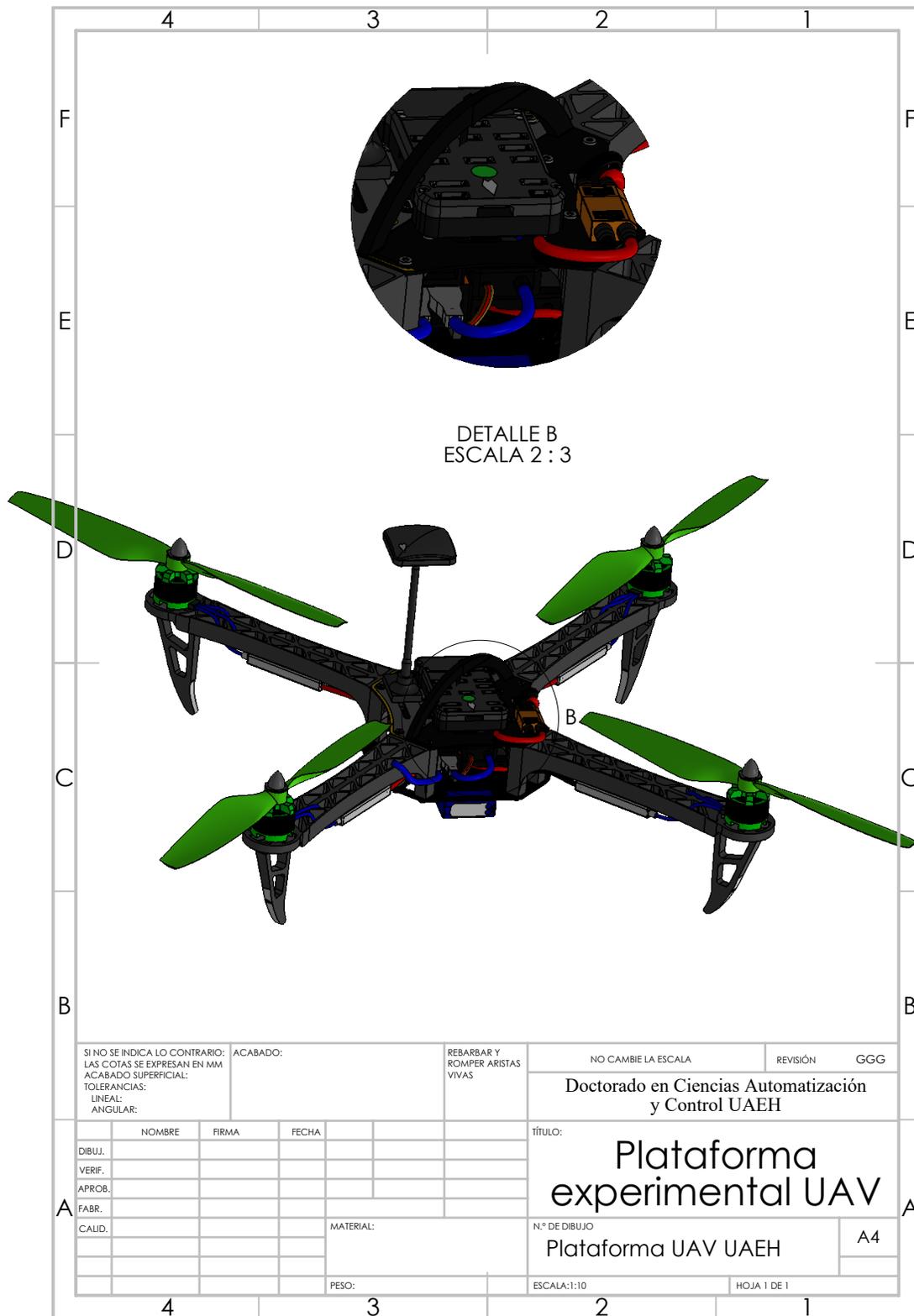


Figura B.16: Plataforma experimental UAV diseño.

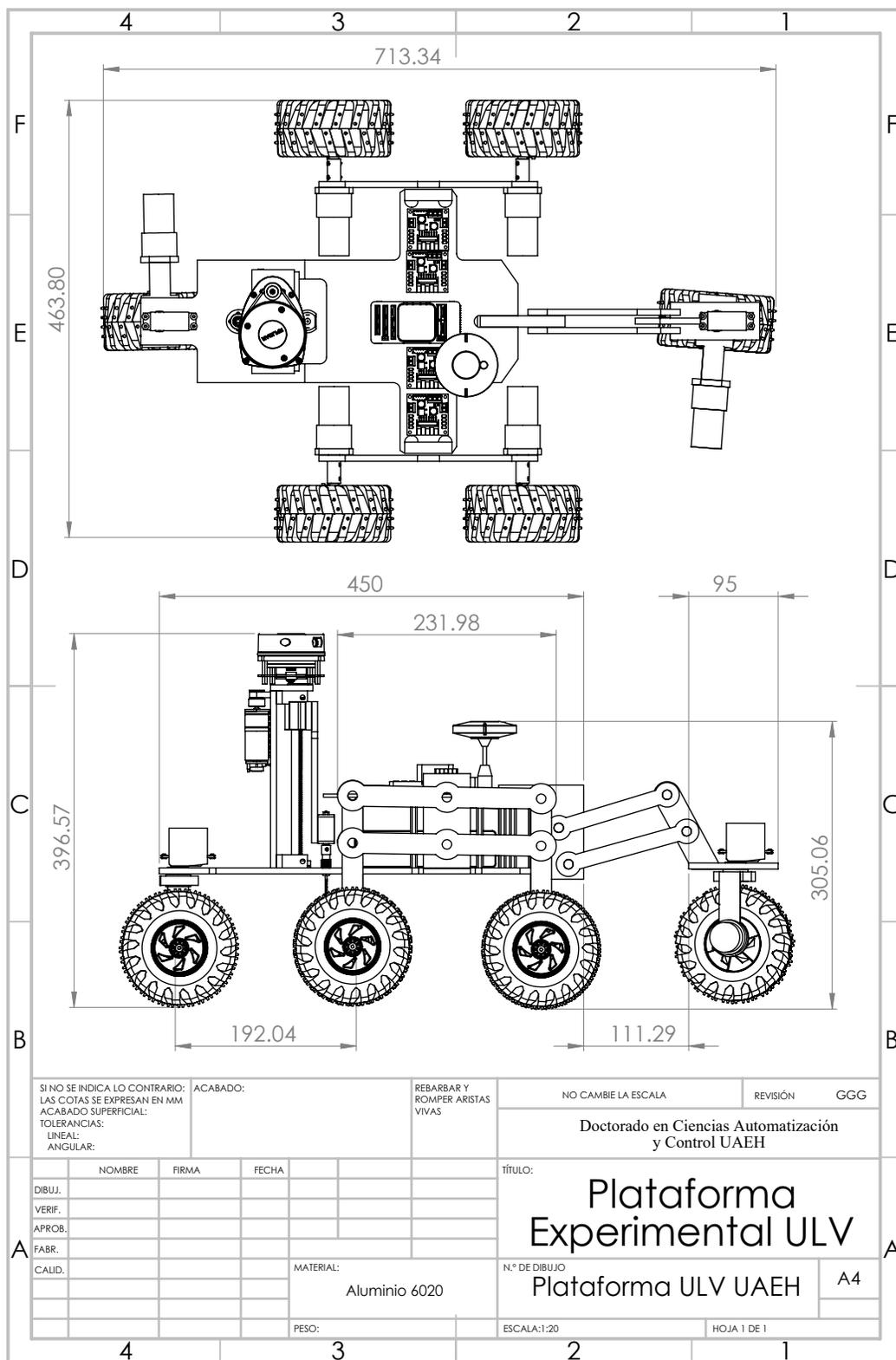


Figura B.17: Plataforma experimental ULV cotas horizontales.

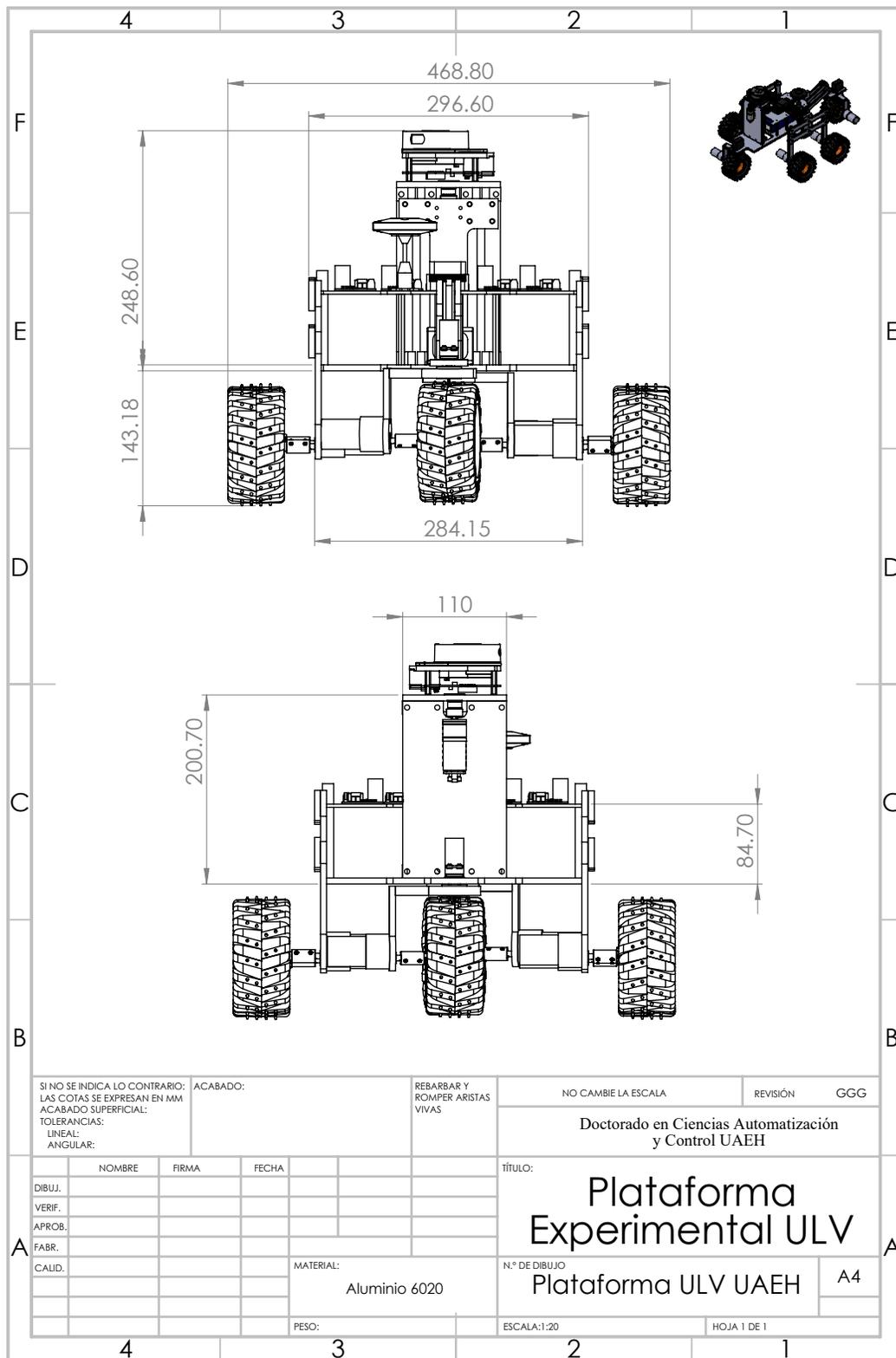


Figura B.18: Plataforma experimental ULV cotas verticales.

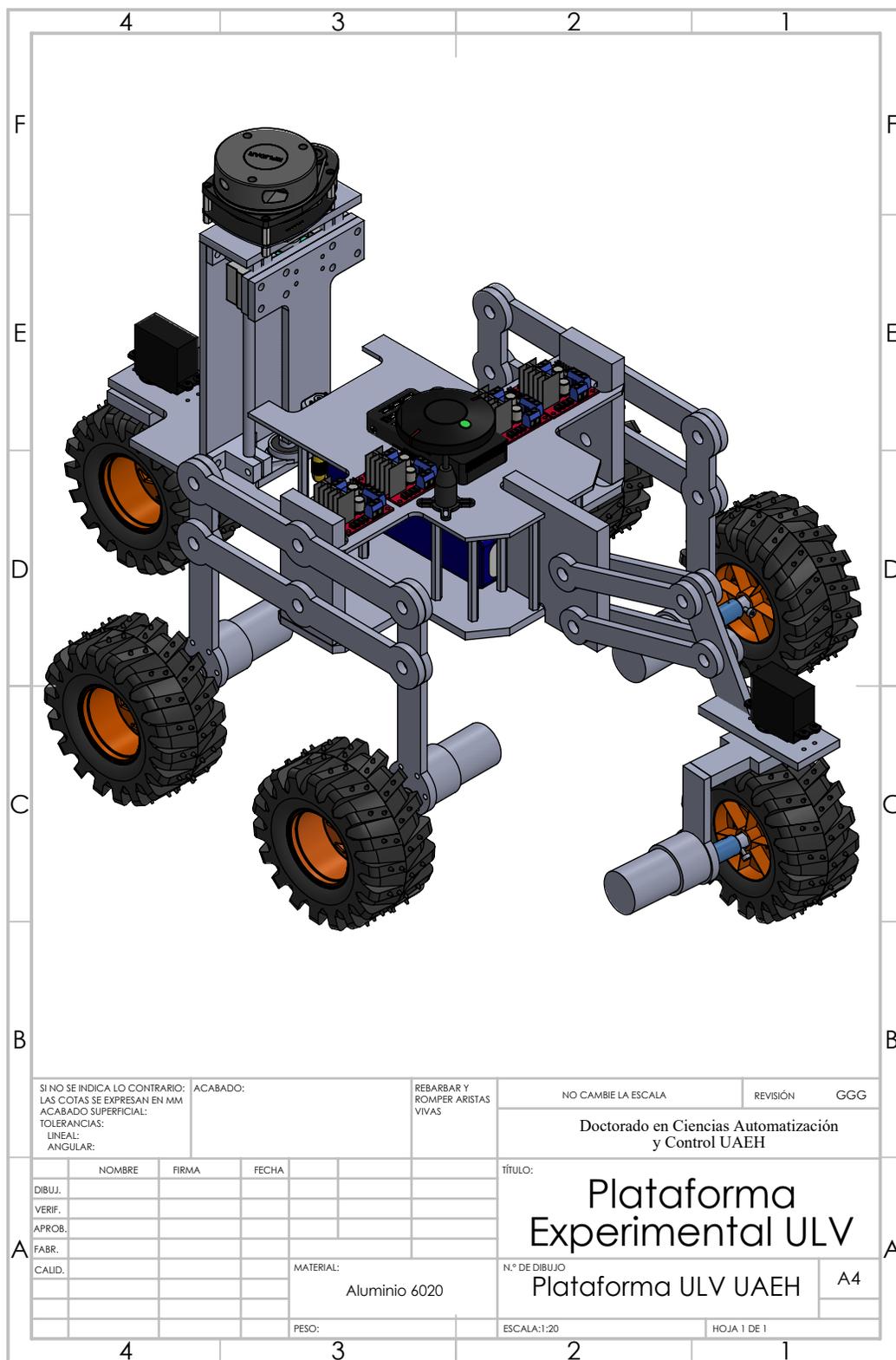


Figura B.19: Plataforma experimental ULV diseño.

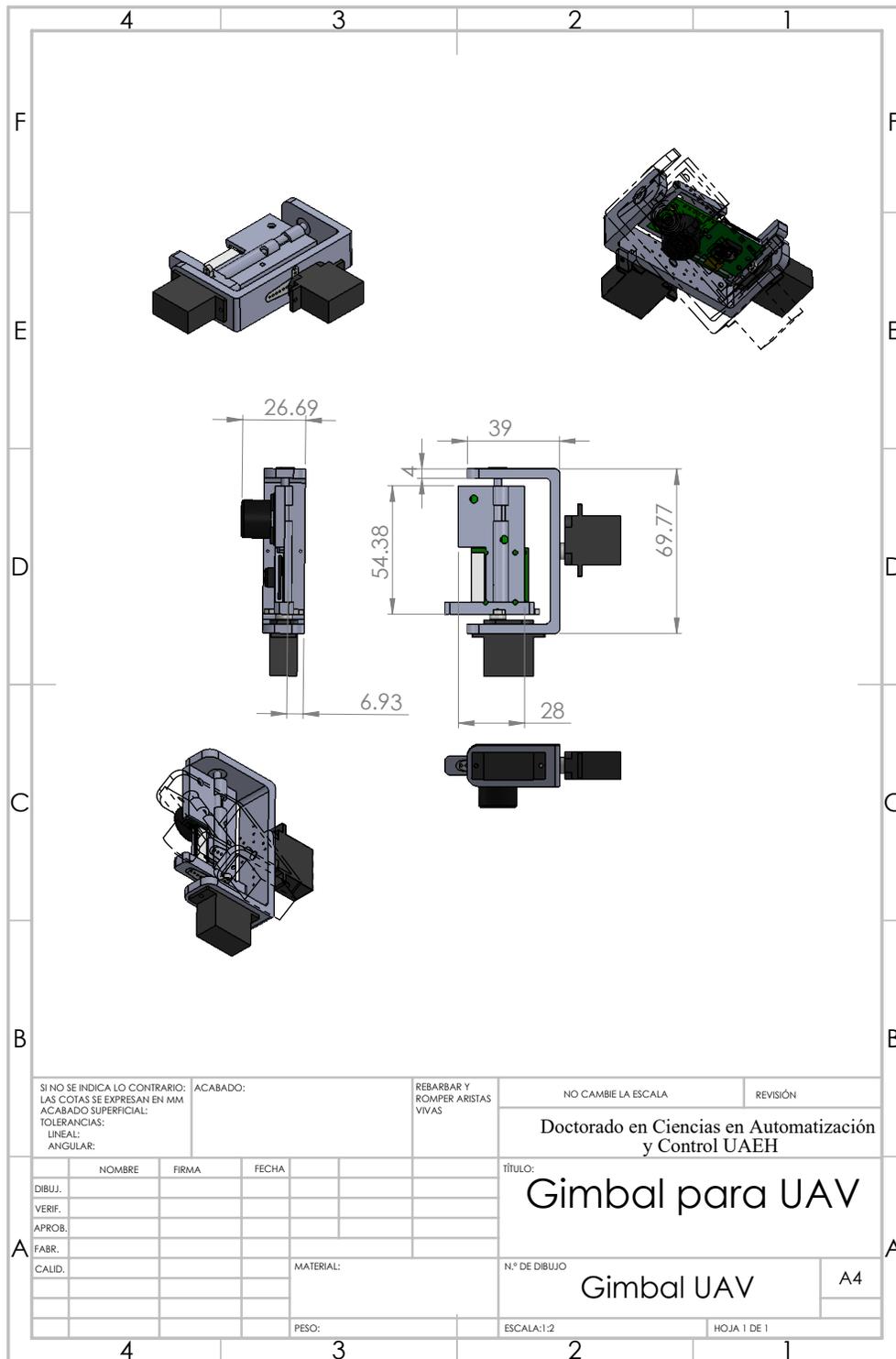


Figura B.20: gimbal UAV diseño.

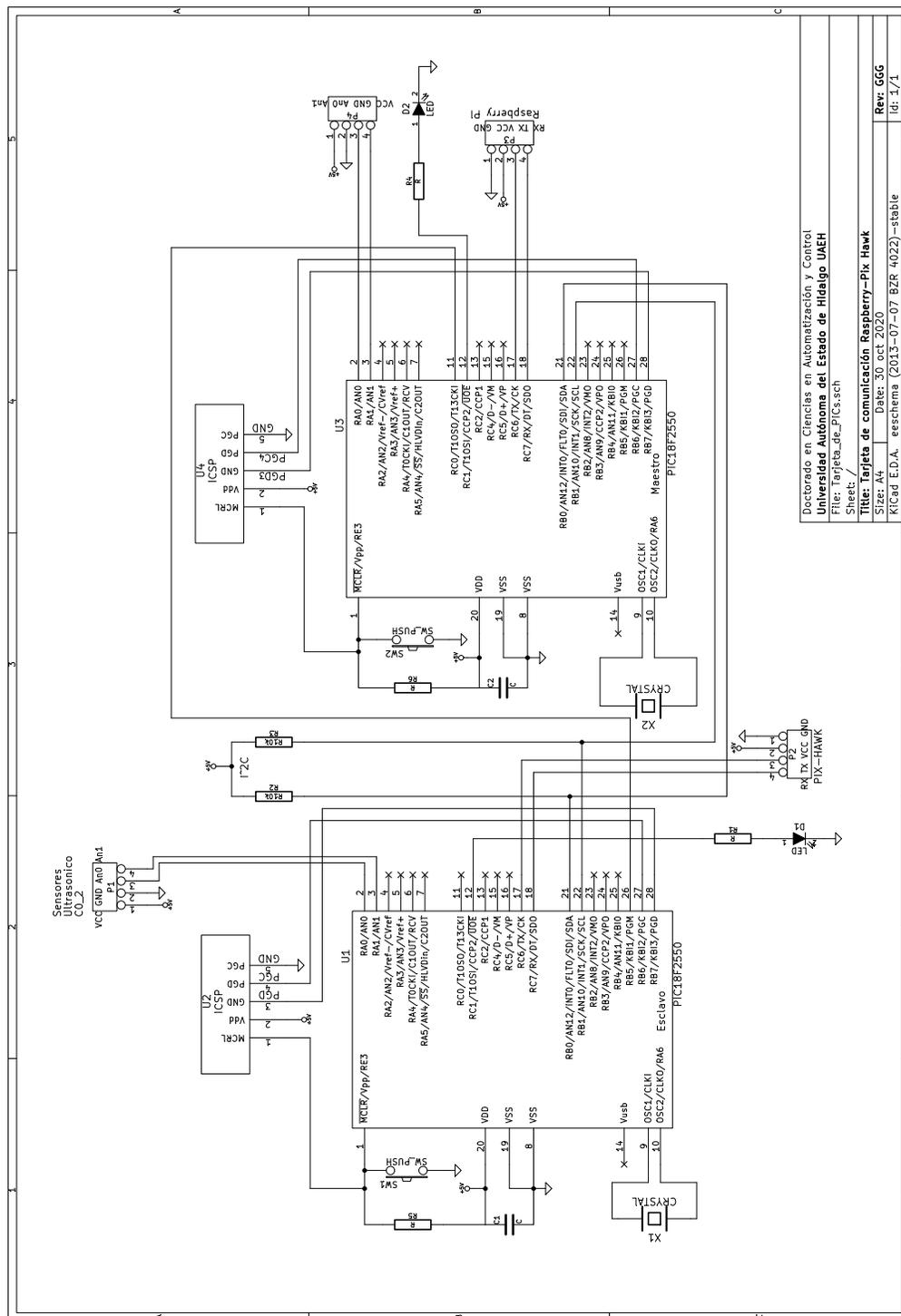


Figura B.21: Esquema de tarjeta electrónica RAS-PIX.

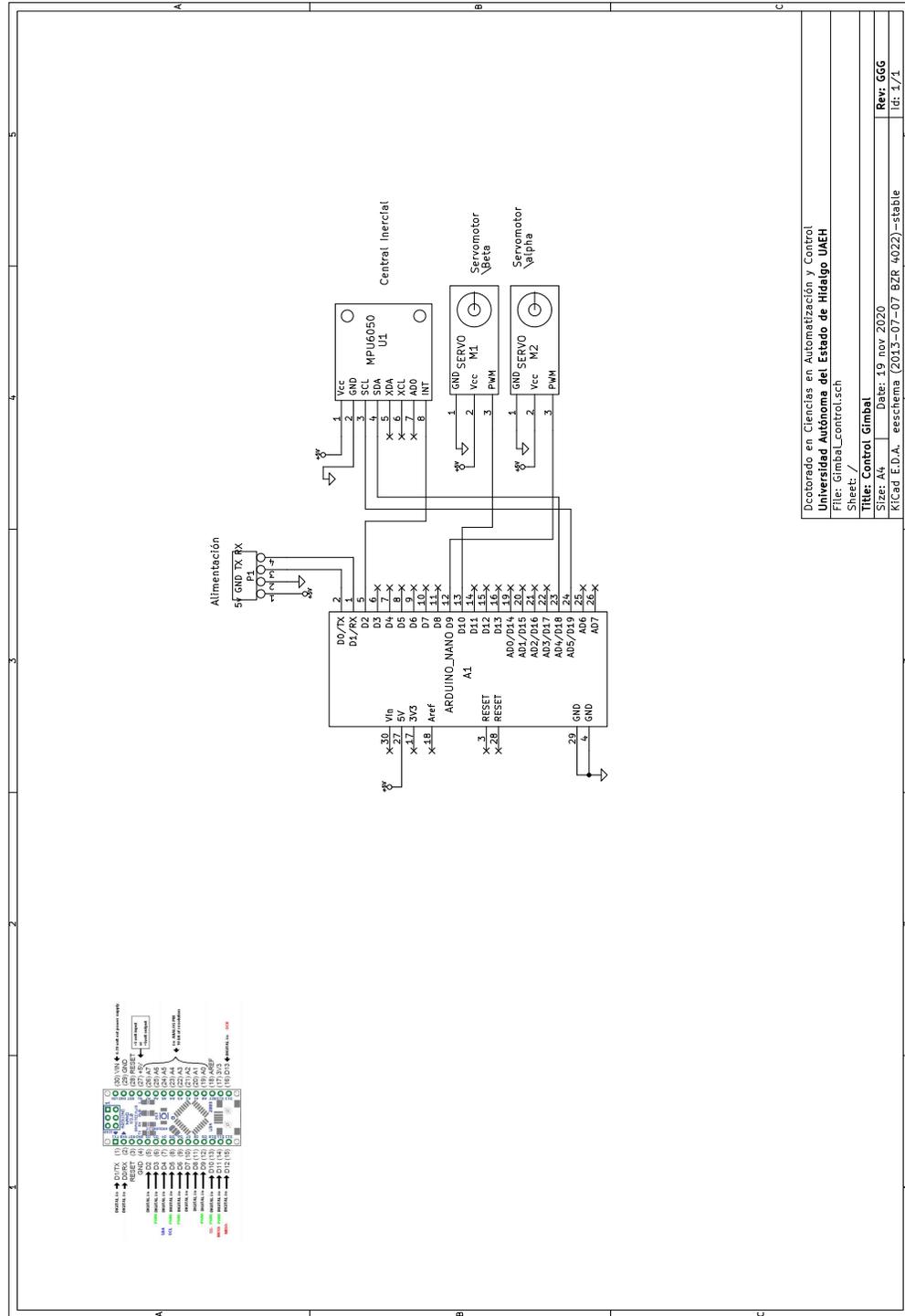


Figura B.22: Esquema de tarjeta electrónica usada para control de gimbal.

Article

Discrete Integral Optimal Controller for Quadrotor Attitude Stabilization: Experimental Results

Gildardo Godinez-Garrido ^{1,2} , Omar-Jacobo Santos-Sánchez ¹ , Hugo Romero-Trejo ^{1,*} 
and Orlando García-Pérez ¹ 

¹ Research Center in Information Technology and Systems, Autonomous University of the State of Hidalgo, Pachuca 42184, Mexico; gildardo.godinez@utectulancingo.edu.mx (G.G.-G.); omarj@uaeh.edu.mx (O.-J.S.-S.); orlando_garcia@uaeh.edu.mx (O.G.-P.)

² Industrial Electromechanical Area, Technological University of Tulancingo (UTEC), Tulancingo 43642, Mexico

* Correspondence: rhugo@uaeh.edu.mx; Tel.: +52-771-717-2197

Abstract: The Unmanned Aerial Vehicle (UAV) attitude stabilization problem has been dealt with in many previous works through applying a vast range of philosophies of control strategies. In this paper, a discrete controller based on a Linear Quadratic Regulator (LQR) plus integral action is synthesized to stabilize the attitude and altitude of a quadrotor helicopter. This kind of control strategy allows us to reduce the energy consumption rate, and the desired UAV behavior is properly achieved. Experimental tests are conducted with external disturbances such as crosswinds deliberately added to affect the performance of the aerial vehicle. This provides experimental evidence that the integral part considered in the proposed control strategy contributes to improving the performance of the vehicle under external disturbances. In fact, a comparative analysis of potential and kinetic energy consumption is developed between the Optimal Integral Controller (OIC) and a Proportional Integral Derivative Controller (PID), allowing us to determine the level of improvement of the closed-loop system when the discrete Integral Optimal Controller is applied.

Keywords: optimal controller; energy consumption; stabilization; quadcopter



Citation: Godinez-Garrido, G.; Santos-Sánchez, O.-J.; Romero-Trejo, H.; García-Pérez, O. Discrete Integral Optimal Controller for Quadrotor Attitude Stabilization: Experimental Results. *Appl. Sci.* **2023**, *13*, 9293. <https://doi.org/10.3390/app13169293>

Academic Editor: Luigi Fortuna

Received: 6 June 2023

Revised: 10 August 2023

Accepted: 12 August 2023

Published: 16 August 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, several works have been reported in the literature addressing the application of optimal control of quadrotors [1,2]. However, this issue remains interesting to the scientific community because it represents a current challenge from the point of view of control theory and engineering (mainly its real-time applications). Many techniques related to optimal control philosophy have been applied to UAVs, but this paper proposes to add an integral action to Linear Quadratic Regulator, in order to improve the performance of the closed-loop system. Moreover, both the mathematical model and synthesized control strategy are obtained in discrete time domain, ensuring a better representation to be programmed in a microcontroller such that Rabbit 4300. In this sense, recent reported works have demonstrated that it is possible to ensure the stability in discrete-time domain for multi-input-multi-output (MIMO) systems. This has been achieved through a generalization of the Letov formula, as mentioned in [3].

A typical control strategy with integral action is the Proportional-Integral-Derivative (PID) controller, which is widely used in industrial processes and devices because it can be heuristically tuned independently of the mathematical model of the system and can be used in both linear and nonlinear systems. However, there also exist many graphical and theoretical methods to tune it considering the dynamical behavior or the mathematical model of system.

It is well known that the integral action of a PID controller provides a correction for steady-state tracking error even in the presence of uncertainties [4], providing some degree

of robustness to the control loop. Although this three-action controller is a powerful idea, sometimes it is not efficient or appropriate for a specific tasks. Then, it can be improved by modifying the control parameters with an adaptive scheme or by adding a nonlinear part.

This contribution uses the advantages of the integral action of the PID controller combined with modern control techniques through regulation via optimal control in order to improve the performance of a quadrotor when it executes an attitude and altitude stabilization task. Integral control applied to Unmanned Aerial Vehicles has been explored in some reported works. An integral predictive and nonlinear robust control strategy was synthesized in [5], wherein the authors solve the path-following problem for a quadrotor helicopter. The proposed control structure has a hierarchical scheme consisting of a model predictive controller to track the reference trajectory together with a nonlinear H_∞ controller to stabilize the rotational movements of the considered vehicle; satisfactory simulation results were presented. In [6], an Integral Backstepping controller and motion planning are combined to stabilize the helicopter using point-to-point steering stabilization. Simulation results were presented to test the performance of the closed-loop system and its robustness. The same technique was presented in [7]: the goal of this work was to stabilize the attitude, altitude, and position of the vehicle. Satisfactory results for autonomous take-off, hover, landing and collision-avoidance tasks were presented, and all were validated on the OS4 simulation platform.

In addition, a comparison between PID, Linear Quadratic Regulator (LQR) and nonlinear controllers (Adaptive Integral Backstepping Controller) was exposed in [8]. A nonlinear control approach was proposed based on a recursive Lyapunov methodology using the Backstepping technique and an adaptive scheme. Satisfactory simulations and real-time experiments were conducted. In [9], LQR continuous control was used to stabilize the attitude and altitude of an Octocopter. Numerical simulations demonstrated the effectiveness of the control strategy under nominal conditions, and the authors improved the LQR by adding integral action to the altitude controller. In [10], once again an LQR methodology and integral state augmentation were adopted to achieve the desired performance of the control system. The unmeasured state variables were estimated by means of a reduced-order observer. Satisfactory simulation results for the UAV helicopter were presented. Toledo et al. [11] conducted similar work in which they used a control scheme based on Integral Backstepping with sliding modes applied to a multi-rotor vehicle. This control methodology was experimentally tested using the vision system Optitrack. Results were reported for the altitude z and displacement on the x - and y -axes. However, no analysis of energy consumption was presented, nor were real-time tests with external disturbances conducted and reported. In [12], Elkhateem published LQR and LQR-PI controllers that are applied to a quadcopter. The high performance and robustness of the Linear Quadratic Regulator controllers ensure the ability to reduce deviations in state trajectories with minimal control effort. In this work, the weighting matrices are automatically adjusted through a novel method using the full state of the flying robot. Feasibility and performance of the closed-loop system is only tested by simulation routines.

In the literature, it is possible to find documents in which the LQR technique is combined with other types of controllers, such as fuzzy controllers. For example, in [13], Malik presents the development of a longitudinal controller design for an autonomous unmanned aerial vehicle (UAV). In this work, the researchers proposed a dual-loop (inner-outer loop) control method based on intelligent algorithms. The inner feedback loop of controller uses a Linear Quadratic Regulator (LQR) to ensure adaptive stability. Meanwhile, the outer loop controller employs a Fuzzy-PID algorithm for deal with the trajectory tracking task.

Moreover, neuro-fuzzy controllers have also been recently used for the control of UAVs. In [14], Jinjun Rao et al. developed a position control approach for a quadrotor using a cascade Fuzzy Neural Network (FNN). This approach requires offline neural network training, combining the benefits of fuzzy systems and neural networks. According to authors, this fuzzy control strategy demonstrated its ability to minimize the overshoot and

the settling time. This was tested by conducting flight simulations and real-time flights using a DJI Tello quadrotor UAV, showing an acceptable position control performance. In fact, one of advantages of neuro-fuzzy controllers is their ability to handle nonlinear systems, which offers better adaptability. Particularly, the controller architecture proposed in that work allows to optimize the use of the robot's energy resources and also providing robustness to the control loop. Also, this produces a balance between adaptability and efficiency, making it particularly useful for UAV applications where energy efficiency and robust control are crucial. That type of controller can also be used in other fields such as the study of seismic structural control. In this field, the contributions reported works by Abbas and his collaborators in [15–17] should also be mentioned, they use neuro-fuzzy controllers and a PID controllers to tackle both stabilization and trajectory tracking task for aerial robots.

In this contribution, UAV stabilization is tackled using a controller that combines an optimal strategy (LQR) with the integral action. A difference from other reported works such as [5,9,10,12] is that this optimal synthesized controller is tested in a real-time setting, providing experimental results for the altitude stabilization problem in a four-rotor helicopter. To control the position and orientation of the vehicle, the system was subdivided into four subsystems, as was proposed in [18], and the integral action is added as an additional state variable [4] in the four subsystems. A previous exact linearization was performed on the dynamical model in order to remove the Coriolis terms and transform the rotational dynamics in the second-order differential equation depending on external torque inputs. So for control of the altitude and attitude of the UAV, four optimal controls with integral action are proposed: one for each subsystem. The controllers for rotational dynamics are synthesized using the assumption that pitch, roll and yaw are inside a bounded region around the origin, which is an equilibrium point for this aerial vehicle. The proposed optimal control strategy assumes that all state variables are available. Our proposal is experimentally tested in takeoff and altitude stabilization tasks. An Optitrack vision system is used to obtain the whole state of the vehicle, and satisfactory results are obtained when the optimal discretized controller with integral action is applied.

So the main contributions of this work are:

1. Synthesis, analysis and implementation of Optimal Integral Control (OIC) tuned under the QR approach and applied to trajectory tracking of takeoff and hover flight of UAVs.
2. Experimental validation of the OIC by real-time tests in the presence of induced crosswind disturbances applied during the trajectory tracking of takeoff and hover flight of a UAV, allowing the analysis of the robustness of the closed loop with the proposed control scheme.
3. Comparison of the kinetic and potential energy between the PID and the Optimal Integral Control when a trajectory tracking task is executed for the take-off and hover-flight phases of the UAV in the presence of induced crosswind.

The paper is organized as follows: The nomenclature and symbols used throughout this document are presented in Section "Nomenclature", while the introduction is reported in Section 1. Moreover, Section 2 is devoted to showing the mathematical model of the UAV together with the synthesis of the proposed control law. In Section 3, the experimental platform is shown. Real-time experimental results are displayed in Section 4, and finally, the conclusion and discussion are reported in Section 5.

2. Control Strategy

In this section, we synthesize the proposed optimal controller with integral action added. Firstly, some basic concepts about integral control are briefly recalled in order to set up a discrete time control strategy to be applied to a quadcopter.

2.1. Integral Control

Consider the nonlinear system:

$$\begin{aligned} \dot{\bar{x}} &= f(\bar{x}, \bar{u}) \\ \bar{y} &= h(\bar{x}), \end{aligned}$$

where the state $\bar{x} \in \mathbb{R}^n$, and the vector control $\bar{u} \in \mathbb{R}^p$. The variables f and h are continuously differentiable functions in a domain $D_{\bar{x}} \times D_{\bar{u}} \subset \mathbb{R}^n \times \mathbb{R}^p$, and $\bar{y} \in \mathbb{R}^p$ is the controlled output. Let $\bar{y}_R \in \mathbb{R}^p$ be a constant reference; the integral control is a feedback state such that

$$\bar{y}(t) \rightarrow \bar{y}_R, \quad t \rightarrow \infty$$

Assume that the controlled output \bar{y} can be measured. Note that for our case $\bar{y} = \bar{x}$ because when using the Optitrack vision system, the complete state is measurable and therefore available. The regulation task will be achieved by stabilizing the system at an equilibrium point where $\bar{y} = \bar{y}_R$. In order to maintain it in that equilibrium condition, there exists a pair $(\bar{x}_{ss}, \bar{u}_{ss}) \in D_{\bar{x}} \times D_{\bar{u}}$ such that:

$$\begin{aligned} 0 &= f(\bar{x}_{ss}, \bar{u}_{ss}), \\ 0 &= h(\bar{x}_{ss}) - \bar{y}_R. \end{aligned}$$

Assume that these equations have a unique solution $(\bar{x}_{ss}, \bar{u}_{ss})$. Now, the integral action is included as follows: consider the tracking error $\bar{e} = \bar{y} - \bar{y}_R$. Then, the following equivalence is defined

$$\dot{\bar{\sigma}} = \bar{e} = \bar{y} - \bar{y}_R,$$

So control will be obtained as a feedback function of \bar{x} and $\bar{\sigma}$ such that in the closed loop there is an equilibrium point $(\bar{x}, \bar{\sigma})$ with $\bar{x} = x_{ss}$. Assuming that the system is linearizable around $x_{ss}, \bar{\sigma}, u_{ss}$, it follows that:

$$\begin{aligned} \begin{bmatrix} \dot{\bar{\xi}} \\ \bar{\xi} \end{bmatrix} &= \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix} \bar{\xi} + \begin{bmatrix} B \\ 0 \end{bmatrix} v \triangleq \mathcal{A}\bar{\xi} + \mathcal{B}v, \end{aligned} \tag{1}$$

with

$$\bar{\xi} = \begin{bmatrix} \bar{x} - x_{ss} \\ \bar{\sigma} - \bar{\sigma} \end{bmatrix}$$

$v = \bar{u} - u_{ss}$ where:

$$\begin{aligned} A &= \left. \frac{\partial f}{\partial \bar{x}} \right|_{(\bar{x}, \bar{u})=(x_{ss}, u_{ss})}, \quad B = \left. \frac{\partial f}{\partial \bar{u}} \right|_{(\bar{x}, \bar{u})=(x_{ss}, u_{ss})}, \\ C &= \left. \frac{\partial h}{\partial \bar{x}} \right|_{\bar{x}=x_{ss}} \end{aligned}$$

Assume that the pair (A, B) is controllable and

$$\text{rank} \begin{bmatrix} A & B \\ C & 0 \end{bmatrix} = n + p.$$

Then $(\mathcal{A}, \mathcal{B})$ is controllable [4]. Then, design a matrix \mathcal{K} such that $\mathcal{A} + \mathcal{B}\mathcal{K}$ is Hurwitz [4]. Consider the partition for the matrix \mathcal{K} as $[\mathcal{K}_1 \ \mathcal{K}_2]$. The control signal is then defined by:

$$\bar{u} = \mathcal{K}_1(\bar{x} - x_{ss}) + \mathcal{K}_2(\bar{\sigma} - \bar{\sigma}) + u_{ss}.$$

It is not difficult to verify that the closed-loop nonlinear system has a unique equilibrium point $(x_{ss}, \bar{\sigma}_{ss})$ [4]. As is demonstrated in [4], the equilibrium point $(x_{ss}, \bar{\sigma}_{ss})$ is

exponentially stable, and all solutions starting close enough to this equilibrium point approach it as t tends to infinity. Then $\bar{y}(t) - \bar{y}_R \rightarrow 0$ as $t \rightarrow \infty$. Following these ideas, in this contribution, the matrix \mathcal{K} is designed using the optimal control approach. For altitude, we take advantage of the fact that in the mathematical model for the altitude of the quadrotor, it can be stabilized by exact linearization, and then, it is not necessary to linearize the nonlinear dynamics. The integral action is used to minimize the effects of external disturbances on vehicle performance. Figure 1 shows the basic scheme for the control of integral action.

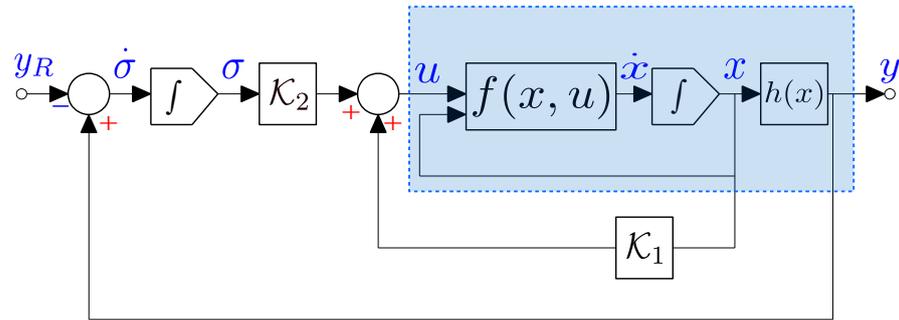


Figure 1. Integral control scheme [4].

2.2. Mathematical Model of the Quadrotor

The following assumptions are considered in this paper to obtain a simplified version of the mathematical model of a vehicle [7,18]

1. The quadcopter is a rigid and symmetric body.
2. The center of gravity of the vehicle coincides with the origin of the body frame.
3. The propellers are rigid and have a fixed pitch.
4. At low velocities, aerodynamic effects can be neglected.
5. The angles are restricted: $-\pi/2 < \phi < \pi/2$, $-\pi/2 < \theta < \pi/2$ and $\psi = 0$.

The dynamical model considered is that reported in [19,20] with the following structure:

$$\begin{aligned}
 m \ddot{x} &= -u \sin \theta \\
 m \ddot{y} &= u \cos \theta \sin \phi \\
 m \ddot{z} &= u \cos \theta \cos \phi - mg \\
 \ddot{\phi} &= \tau_\phi \\
 \ddot{\theta} &= \tau_\theta \\
 \ddot{\psi} &= \tau_\psi,
 \end{aligned}
 \tag{2}$$

where x and y are the displacements in the horizontal plane, z is the vertical position, ψ is the yaw angle around the z -axis, θ is the pitch angle relative to the y -axis, and ϕ is the roll angle around the x -axis. The control inputs are: u , τ_ϕ , τ_θ , and τ_ψ , with u the collective throttle generated by the four motors to lift the UAV; τ_ϕ , τ_θ , and τ_ψ are the torques generated around the axes x , y , and z , respectively.

Here, the authors have assumed that there exists a previous controller $\tilde{\tau} = C(\eta, \dot{\eta})\dot{\eta} + J\tau$ (see the mathematical model given by Equations (2.4) and (2.5), p. 34 of [18]), where $\eta = (\phi, \theta, \psi)^T$ is the angular position vector, $C(\eta, \dot{\eta})$ is the Coriolis terms matrix, and J is the inertia matrix. With this control, we arrive at the last three equations of the mathematical model given by (2), which define the resulting rotational dynamics $\tilde{\tau} = (\tau_\phi, \tau_\theta, \tau_\psi)^T$. Figure 2 shows a schematic representation of the positions and angles of the quadrotor.

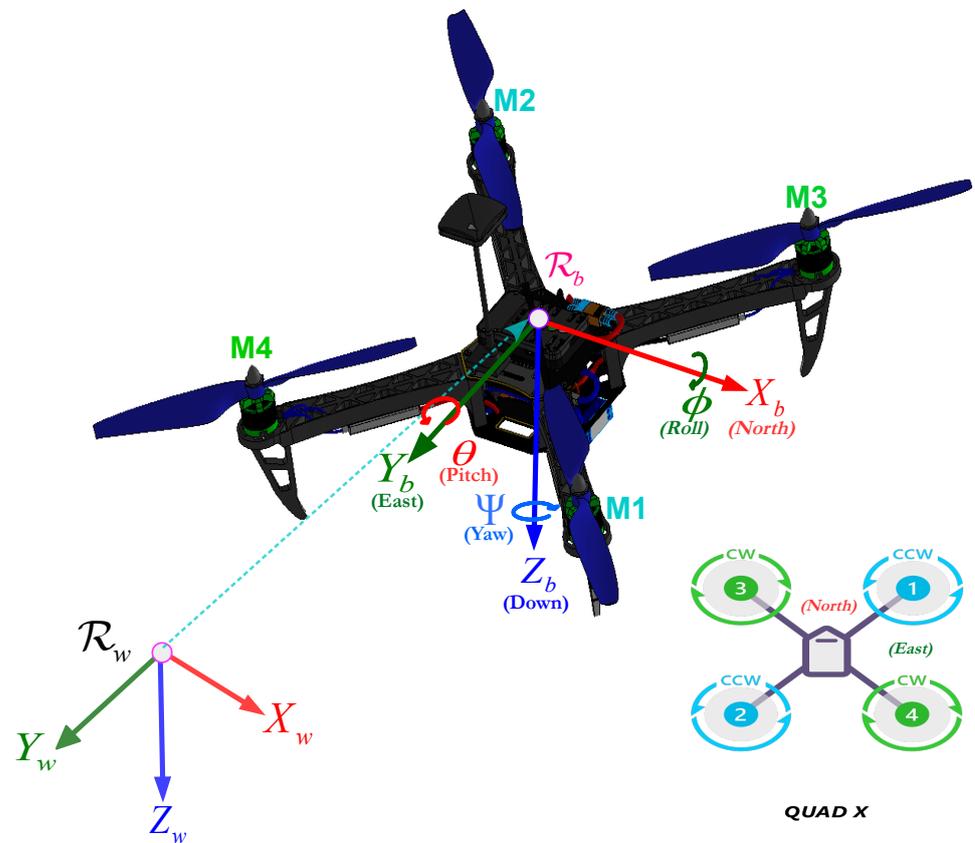


Figure 2. Positions and angles of the UAV.

Now, in order to obtain a discrete space state representation of the mathematical model of a flying robot, the following state variables are defined:

$$\begin{aligned}
 x_1 &= x, & x_2 &= \dot{x}, & x_3 &= y, & x_4 &= \dot{y}, \\
 x_5 &= z, & x_6 &= \dot{z}, & x_7 &= \phi, & x_8 &= \dot{\phi}, \\
 x_9 &= \theta, & x_{10} &= \dot{\theta}, & x_{11} &= \psi, & x_{12} &= \dot{\psi},
 \end{aligned}$$

and discretizing the related continuous model by applying the Euler approximation considering a sampled period T , it becomes:

$$\begin{aligned}
 x_1(k+1) &= Tx_2(k) + x_1(k) \\
 x_2(k+1) &= -\frac{T}{m}u(k) \sin(x_9(k)) + x_2(k) \\
 x_3(k+1) &= Tx_4(k) + x_3(k) \\
 x_4(k+1) &= \frac{T}{m}u(k) \cos(x_9(k)) \sin(x_7(k)) + x_4(k) - Tg \\
 x_5(k+1) &= Tx_6(k) + x_5(k) \\
 x_6(k+1) &= \frac{T}{m}u(k) \cos(x_9(k)) \cos(x_7(k)) + x_6(k) - Tg \\
 x_7(k+1) &= Tx_8(k) + x_7(k) \\
 x_8(k+1) &= T\tau_\phi(k) + x_8(k) \\
 x_9(k+1) &= Tx_{10}(k) + x_9(k) \\
 x_{10}(k+1) &= T\tau_\theta(k) + x_{10}(k) \\
 x_{11}(k+1) &= Tx_{12}(k) + x_{11}(k) \\
 x_{12}(k+1) &= T\tau_\psi(k) + x_{12}(k)
 \end{aligned}$$

All of the above was performed in order to apply a digital version of a synthesized optimal controller with integral action to a quadrotor helicopter.

2.3. Discrete Model and Integral Control

As mentioned previously, the mathematical model can be subdivided into four subsystems: subsystem $z(k)$, $\psi(k)$, $x(k) - \theta(k)$ and $y(k) - \phi(k)$. For subsystem $z(k)$, we have:

$$\begin{aligned} x_5(k+1) &= Tx_6(k) + x_5(k), \\ x_6(k+1) &= T \left[\frac{u(k)}{m} \cos(x_9(k)) \cos(x_7(k)) - g \right] + x_6(k) \end{aligned}$$

Now, we use an exact linearization as follows:

$$u(k) = m(u_1(k) + g)(\cos(x_9(k)) \cos(x_7(k)))^{-1},$$

with these control laws for each subsystem, which are linear and are defined as:

$$\begin{bmatrix} x_{1z}(k+1) \\ x_{2z}(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}}_{A_z} \begin{bmatrix} x_{1z}(k) \\ x_{2z}(k) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ T \end{bmatrix}}_{B_z} [u_1(k)],$$

where $x_5(k) \triangleq x_{1z}(k)$ and $x_6(k) \triangleq x_{2z}(k)$. Define $e_{1z}(k) = x_{1z}(k) - x_{1zR}(k)$, and $e_{2z}(k) \triangleq x_{2z}(k) - x_{2zR}(k)$; these are the errors for $x_{1z}(k)$ and $x_{2z}(k)$, respectively, and $x_{1zR}(k)$ and $x_{2zR}(k)$ are the references. The augmented system for the subsystem $z(k)$ is:

$$\zeta_z(k+1) = \underbrace{\begin{bmatrix} A_z & 0 \\ C_z & I_2 \end{bmatrix}}_{A_z} \zeta_z(k) + \underbrace{\begin{bmatrix} B_z \\ 0 \end{bmatrix}}_{B_z} u_1(k)$$

where $\zeta_z(k) = [x_{1z}(k) \ x_{2z}(k) \ \sigma_{1z}(k) \ \sigma_{2z}(k)]^T$ and $C_z = TI_2$. It is not a difficult task to verify that the pair (A_z, B_z) is controllable in a finite number of steps. Define the following performance index:

$$\mathcal{J}_z = \sum_{k=1}^{\infty} (\zeta_z^T(k) Q_z \zeta_z(k) + u_1^2(k) \mathcal{R}_z),$$

where Q_z is a semidefinite positive matrix of appropriate dimensions and \mathcal{R}_z is a real positive number. As the pair (A_z, B_z) is controllable, there exists an unique solution to the Riccati equation given by:

$$\mathcal{P}_z = A_z^T \mathcal{P}_z A_z - A_z^T \mathcal{P}_z B_z (\mathcal{R}_z + B_z^T \mathcal{P}_z B_z)^{-1} B_z^T \mathcal{P}_z + Q_z,$$

and the solution \mathcal{P}_z describes the optimal control for the subsystem $z(k)$ given by:

$$u_1^*(k) = -(\mathcal{R}_z + B_z^T \mathcal{P}_z B_z)^{-1} B_z^T \mathcal{P}_z A_z \zeta_z^*(k).$$

The discrete model for $\psi(k)$ is given by:

$$\begin{aligned} x_{11}(k+1) &= Tx_{12}(k) + x_{11}(k), \\ x_{12}(k+1) &= T\tau_\psi(k) + x_{12}(k), \end{aligned}$$

Let $x_{11}(k) \triangleq x_{1\psi}(k)$ and $x_{12}(k) \triangleq x_{2\psi}(k)$; then, the subsystem $\psi(k)$ can be rewritten as:

$$\begin{bmatrix} x_{1\psi}(k+1) \\ x_{2\psi}(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}}_{A_\psi} \begin{bmatrix} x_{1\psi}(k) \\ x_{2\psi}(k) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ T \end{bmatrix}}_{B_\psi} \tau_\psi(k)$$

Define $e_{1\psi}(k) \triangleq x_{1\psi}(k) - x_{1\psi R}(k)$ and $e_{2\psi}(k) \triangleq x_{2\psi}(k) - x_{2\psi R}(k)$, where $x_{1\psi R}(k)$ and $x_{2\psi R}(k)$ are the given references for the variables $x_{1\psi}(k)$, $x_{2\psi}(k)$, respectively. Then, the augmented vector for the subsystem $\psi(k)$:

$$\xi_{\psi(k)}(k) = [x_{1\psi}(k) \ x_{2\psi}(k) \ \sigma_{1\psi}(k) \ \sigma_{2\psi}(k)]^T.$$

The space state representation of the augmented system is:

$$\xi_{\psi(k)}(k + 1) = \mathcal{A}_\psi \xi_{\psi}(k) + \mathcal{B}_\psi \tau_\psi(k)$$

where:

$$\mathcal{A}_\psi = \begin{bmatrix} A_\psi & 0 \\ C_\psi & I_2 \end{bmatrix}, \mathcal{B}_\psi = \begin{bmatrix} B_\psi \\ 0 \end{bmatrix}, C_\psi = TI_2,$$

Consider the performance index:

$$\mathcal{J}_\psi = \sum_{k=1}^{\infty} (x_\psi^T(k) \mathcal{Q}_\psi x_\psi(k) + \tau_\psi^2(k) \mathcal{R}_\psi)$$

with an appropriate dimension matrix $\mathcal{Q}_\psi \geq 0$ and a real $\mathcal{R}_\psi > 0$. As the pair $(\mathcal{A}_\psi, \mathcal{B}_\psi)$ is controllable in a finite number of steps, the optimal control law with integral action is given by:

$$\tau_\psi^*(k) = -(\mathcal{R}_\psi + \mathcal{B}_\psi^T \mathcal{P}_\psi \mathcal{B}_\psi)^{-1} \mathcal{B}_\psi^T \mathcal{P}_\psi \mathcal{A}_\psi x_\psi^*(k),$$

where matrix \mathcal{P}_ψ is the unique solution to the algebraic equation:

$$\mathcal{P}_\psi = \mathcal{A}_\psi^T \mathcal{P}_\psi \mathcal{A}_\psi - \mathcal{A}_\psi^T \mathcal{P}_\psi \mathcal{B}_\psi (\mathcal{R}_\psi + \mathcal{B}_\psi^T \mathcal{P}_\psi \mathcal{B}_\psi)^{-1} \mathcal{B}_\psi^T \mathcal{P}_\psi + \mathcal{Q}_\psi.$$

For the subsystem y - $\phi(k)$:

$$\begin{aligned} x_3(k + 1) &= Tx_4(k) + x_3(k), \\ x_4(k + 1) &= T \left[\frac{u(k)}{m} \cos(x_9(k)) \sin(x_7(k)) \right] + x_4(k), \\ x_7(k + 1) &= Tx_8(k) + x_7(k) \\ x_8(k + 1) &= T\tau_\phi(k) + x_8(k) \end{aligned}$$

The control $u(k)$ was defined as an exact linearization, and the optimal control $u_1^*(k)$; then, the second equation is given by

$$x_4(k + 1) = T(u_1^*(k) + g) \tan x_7(k) + x_4(k),$$

but $u_1^*(k)$ tends to zero when k tends to infinity. Then, there exists $n \in \mathbb{Z}^+$ such that for all $k \geq nT$, $|u_1^*(k)|$ is bounded and neglected; it follows that:

$$x_4(k + 1) = gT \tan x_7(k) + x_4(k).$$

Let there be a control law $\tau_\phi(k)$ that guarantees that $\tan x_7(k) \approx x_7(k)$; therefore, $x_4(k + 1) = gTx_7(k) + x_4(k)$. Having this idea in mind, consider the following definition:

$$\begin{aligned} x_3(k) &\triangleq x_{1y}(k), \ x_4(k) \triangleq x_{2y}(k) \\ x_7(k) &\triangleq x_{3\phi}(k), \ x_8(k) \triangleq x_{4\phi}(k), \end{aligned}$$

with this definition, the space state representation of the linearized system is:

$$\begin{bmatrix} x_{1y}(k+1) \\ x_{2y}(k+1) \\ x_{3\phi}(k+1) \\ x_{4\phi}(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & gT & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{A_{y\phi}} \begin{bmatrix} x_{1y}(k) \\ x_{2y}(k) \\ x_{3\phi}(k) \\ x_{4\phi}(k) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ T \end{bmatrix}}_{B_{y\phi}} \tau_{\phi}(k)$$

In order to use the discrete integral control, define the errors:

$$\begin{aligned} e_{1y}(k) &\triangleq x_{1y}(k) - x_{1yR}(k) \\ e_{2y}(k) &\triangleq x_{2y}(k) - x_{2yR}(k) \\ e_{3\phi}(k) &\triangleq x_{3\phi}(k) - x_{3\phi R}(k) \\ e_{4\phi}(k) &\triangleq x_{4\phi}(k) - x_{4\phi R}(k) \end{aligned}$$

where $x_{1yR}(k)$, $x_{2yR}(k)$, $x_{3\phi R}(k)$ and $x_{4\phi R}(k)$ are the references for the variables $x_{1y}(k)$, $x_{2y}(k)$, $x_{3\phi}(k)$ and $x_{4\phi}(k)$, respectively. Then, the augmented state vector for the subsystem y - ϕ is given by:

$$\begin{aligned} \xi_{y\phi}(k) &= [x_{1y}(k) \quad x_{2y}(k) \quad x_{3\phi}(k) \quad x_{4\phi}(k) \quad \sigma_{1y}(k) \quad \sigma_{2y}(k) \quad \sigma_{3\phi}(k) \quad \sigma_{4\phi}(k)]^T \\ \xi_{y\phi}(k+1) &= \underbrace{\begin{bmatrix} A_{y\phi} & 0 \\ C_{y\phi} & I_2 \end{bmatrix}}_{\mathcal{A}_{y\phi}} \xi_{y\phi}(k) + \underbrace{\begin{bmatrix} B_{y\phi} \\ 0 \end{bmatrix}}_{\mathcal{B}_{y\phi}} \tau_{\phi}(k) \end{aligned}$$

where $C_{y\phi} = TI_2$. As the pair $(\mathcal{A}_{y\phi}, \mathcal{B}_{y\phi})$ is controllable, there exists an optimal control law $\tau_{\phi}^*(k)$

$$\tau_{y\phi}^*(k) = -(\mathcal{R}_{y\phi} + \mathcal{B}_{y\phi}^T \mathcal{P}_{y\phi} \mathcal{B}_{y\phi})^{-1} \mathcal{B}_{y\phi}^T \mathcal{P}_{y\phi} \mathcal{A}_{y\phi} x_{y\phi}^*(k)$$

which minimizes:

$$\mathcal{J}_{y\phi} = \sum_{k=1}^{\infty} (x_{y\phi}^T(k) \mathcal{Q}_{y\phi} x_{y\phi}(k) + \tau_{y\phi}^2(k) \mathcal{R}_{y\phi})$$

where matrix $\mathcal{Q}_{y\phi} \geq 0$ has appropriate dimensions and the real $\mathcal{R}_{y\phi} > 0$. The matrix $\mathcal{P}_{y\phi}$ is the unique solution to the algebraic equation:

$$\mathcal{P}_{\psi} = \mathcal{A}_{\psi}^T \mathcal{P}_{\psi} \mathcal{A}_{\psi} - \mathcal{A}_{\psi}^T \mathcal{P}_{\psi} \mathcal{B}_{\psi} (\mathcal{R}_{\psi} + \mathcal{B}_{\psi}^T \mathcal{P}_{\psi} \mathcal{B}_{\psi})^{-1} \mathcal{B}_{\psi}^T \mathcal{P}_{\psi} + \mathcal{Q}_{\psi}.$$

A similar procedure is used to obtain the optimal control $\tau_{x\theta}^*(k)$ for the subsystem x - θ :

$$\tau_{x\theta}^*(k) = -(\mathcal{R}_{x\theta} + \mathcal{B}_{x\theta}^T \mathcal{P}_{x\theta} \mathcal{B}_{x\theta})^{-1} \mathcal{B}_{x\theta}^T \mathcal{P}_{x\theta} \mathcal{A}_{x\theta} \tilde{x}_{x\theta}^*(k)$$

where:

$$\begin{aligned} x_1(k) &\triangleq x_{1x}(k), & x_2(k) &\triangleq x_{2x}(k), \\ x_9(k) &\triangleq x_{3\theta}(k), & x_{10}(k) &\triangleq x_{4\theta}(k), \\ e_{1x}(k) &\triangleq x_{1x}(k) - x_{1xR}(k), & e_{2x}(k) &\triangleq x_{2x}(k) - x_{2xR}(k), \\ e_{3\theta}(k) &\triangleq x_{3\theta}(k) - x_{3\theta R}(k), & e_{4\theta}(k) &\triangleq x_{4\theta}(k) - x_{4\theta R}(k). \end{aligned}$$

The augmented vector is:

$$\xi_{x\theta}(k) = [x_{1x}(k) \ x_{2x}(k) \ x_{3\theta}(k) \ x_{4\theta}(k) \ \sigma_{1x}(k) \ \sigma_{2x}(k) \ \sigma_{3\theta}(k) \ \sigma_{4\theta}(k)]^T,$$

and the following performance index is minimized:

$$\mathcal{J}_{x\theta} = \sum_{k=1}^{\infty} (\xi_{x\theta}^T(k) \mathcal{Q}_{x\theta} \xi_{x\theta}(k) + \tau_{x\theta}^2(k) \mathcal{R}_{x\theta})$$

where $\mathcal{Q}_{x\theta} \geq 0$, $\mathcal{R}_{x\theta} > 0$ and $\mathcal{P}_{x\theta}$ is the solution to the algebraic Riccati equation:

$$\mathcal{P}_{x\theta} = \mathcal{A}_{x\theta}^T \mathcal{P}_{x\theta} \mathcal{A}_{x\theta} - \mathcal{A}_{x\theta}^T \mathcal{P}_{x\theta} \mathcal{B}_{x\theta} (\mathcal{R}_{x\theta} + \mathcal{B}_{x\theta}^T \mathcal{P}_{x\theta} \mathcal{B}_{x\theta})^{-1} \mathcal{B}_{x\theta}^T + \mathcal{Q}_{x\theta}$$

where

$$\mathcal{A}_{x-\theta} = \begin{bmatrix} A_{x\theta} & 0 \\ C_{x\theta} & I_2 \end{bmatrix}, \mathcal{B}_{x\theta} = \begin{bmatrix} B_{x\theta} \\ 0 \end{bmatrix}$$

and

$$A_{x\theta} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & -gT & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}, B_{x\theta} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ T \end{bmatrix},$$

$$C_{x\theta} = TI_2.$$

These controllers were tested on an experimental platform described in the next section.

Remark 1. *The OIC strategy combines two advantages of the PID and LQR control approaches: the integral part of the error and the penalization of the energy consumption and convergence of the state, respectively. In contrast to the suboptimal nonlinear discrete control approach [21], which penalizes the energy consumption and convergence of the state with state feedback plus an offset, the OIC substitutes the offset part by the integral part, which provides more robustness in a closed loop in the presence of external disturbances.*

3. Experimental Platform

As shown in Figure 3, the experimental platform setup uses an Optitrack Flex 3 vision system to compute the UAV’s position and orientation. The UAV has 6 markers, which are looked at by 12 cameras; the information generated by the cameras is sent to Motive software using the USB protocol. This information is sent to Visual C++ software from Motive software using sockets; it allows to compute in C++ language the integral control laws applied to stabilize the UAV at hover. The PC and UAV are communicated via RS-232 wireless protocol through two Xbee Pro S1 modems at 38,400 bits per second. Furthermore, a Futaba RF radio control is used for manual wireless control of the quadcopter, acting as emergency control if a risk situation occurs in the UAV.

Figures 3 and 4 show how a Parrot’s frame was used to build the platform’s quadrotor. The UAV has an embedded Rabbit module RCM4300, one inertial measurement unit (IMU) 3DM-GX3 from MicroStrain, one radio receiver and one Xbee Pro S1 modem.

The vision system obtains the position and orientation of the UAV: $x(k)$, $y(k)$, $z(k)$, $\theta(k)$, $\phi(k)$ and $\psi(k)$. The velocity of each variable is estimated using Visual C++. Integral control laws are computed in the same way: $u_{PC}(k)$ (height control computed by the PC), $\tau_{\phi PC}(k)$ (roll control computed by the PC) and $\tau_{\theta PC}(k)$ (pitch control computed by the PC); these control signals are sent from the PC to the Rabbit microcontroller via RS232 wireless protocol using Xbee modems. The Futaba RF radio generates calibration signals $u_c(k)$, $\tau_{\phi c}(k)$, $\tau_{\theta c}(k)$ and $\tau_{\psi c}(k)$; the first one gives an offset of the $z(k)$ position to compensate for gravity; the last ones give orientation offsets of the UAV’s $\theta(k)$, $\phi(k)$ and $\psi(k)$, respectively. In this way, the UAV’s orientation and height position are calibrated. RF radio signals are

fixed and are only used to compensate for measurement errors caused by the IMU or the vision system.

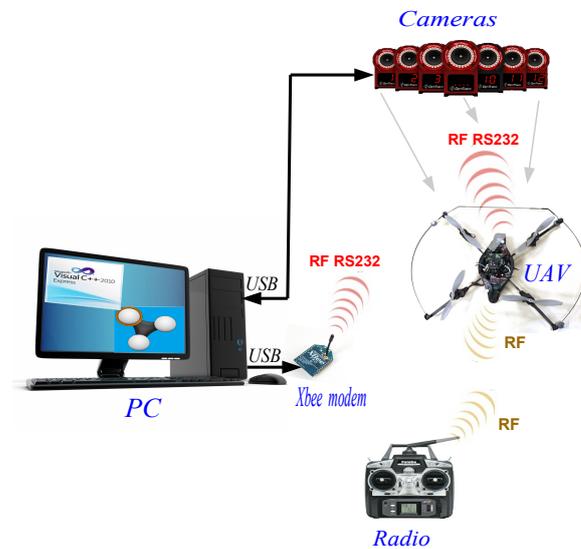


Figure 3. Experimental platform scheme.

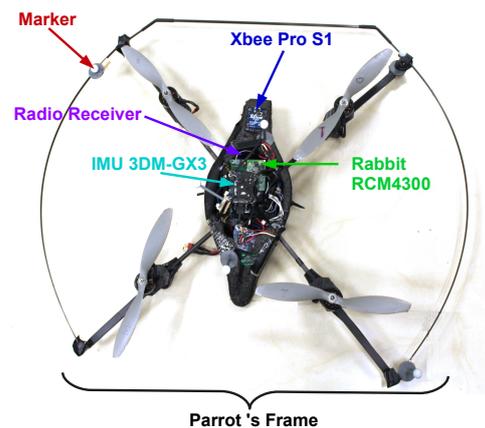


Figure 4. Quadrotor vehicle.

The UAV radio receiver sends the calibration signals to the Rabbit microcontroller using the I^2C protocol. The Rabbit microcontroller reads angular orientation from the IMU ($\theta_{IMU}(k)$, $\phi_{IMU}(k)$ and $\psi_{IMU}(k)$) by the I^2C protocol. The information generated by the IMU produces computer orientation control signals in a direct way because noise affects the vision system, causing errors in orientation measurement to the PC, so control signals used for orientation are computed directly in the microcontroller. PC control signals, radio calibration signals and the control signals computed in the microcontroller are added to the calculation of $u(k)$, $\tau_\phi(k)$, $\tau_\theta(k)$ and $\tau_\psi(k)$, which allow generation of the PWM signals applied to the four motors of the UAV (see Figure 5).

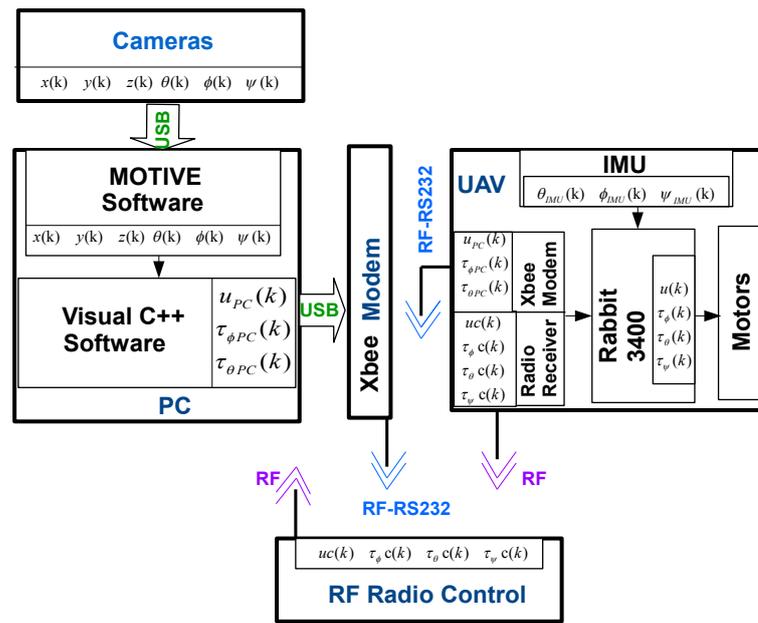


Figure 5. System variable block diagram.

4. Experimental Results

Two experiments were conducted: The first considers the trajectory tracking problem during takeoff and hover flight using an OIC controller in the UAV. The second one addresses the trajectory tracking problem during takeoff and hover flight by applying the PID and OIC controllers and incorporating crosswind disturbances to the UAV.

4.1. Implementation of Optimal Integral Control Algorithm in the UAV

By combining the Linear Quadratic Regulator (LQR) controller and the Integral Controller (IC), which is the main idea presented in this research work, an Optimal Integral Controller (OIC) is generated. The aforementioned is possible by applying an exact linearization to the model of the Unmanned Aerial Vehicle (UAV) presented in [20]. The process corresponding to this step is described in detail separately in Section 2.3 from a mathematical perspective. As can be seen, a discrete representation of the model is required for each subsystem of the aerial robot.

The mathematical model of the robot described in Section 2.2 is divided into four subsystems. For each of these subsystems $z(k)$, $\psi(k)$, $x(k) - \theta(k)$ and $y(k) - \phi(k)$, the integral control structure shown in Section 2.3 is applied, and the Riccati-type algebraic equation is numerically solved for every discretized subsystem. This is done considering the augmented penalty matrices Q and R , which have been assigned as bellow.

For the system $z(k)$, the two matrices are defined by

$$Q_z = \begin{bmatrix} 50 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad R_z = [490]$$

while matrices used to penalize the subsystem $\psi(t)$ are chosen as:

$$Q_\psi = \begin{bmatrix} 18 & 0 & 0 & 0 \\ 0 & 171 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad R_\psi = [180]$$

For the subsystems $y-\phi$ and $x-\theta$, the penalization matrices are:

$$Q_{y\phi} = Q_{x\theta} = \begin{bmatrix} 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 9 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$R_{x\theta} = [40], \quad R_{y\phi} = [31]$$

To test the OIC (Integral Optimal Control) algorithm, it is applied in a controlled environment to the UAV. Tracking a takeoff trajectory and maintaining hover flight with the UAV is described by the following conditions: The experiments consider a sampling period of $T = 50$ ms, set points of translation positions are selected at $x_{ref}(t) = y_{ref}(t) = 0$ m, and altitude $|z(t)|$ is chosen as $z_{ref} = 0.45t$ until 0.3 m; once this altitude value is reached, the helicopter then remains there. Set points related to orientation are fixed at $\theta_{ref} = \phi_{ref} = \psi_{ref} = 0$ deg.

The figures shown below were obtained from the real-time experimentation. Firstly, Figure 6 shows the translational position $x(t), y(t), z(t)$ behavior versus their respectively set points when the Integral Optimal Controller is applied to the quadrotor helicopter. Errors in those translational variables are shown in Figure 7. Moreover, the translational velocities related to each of the axes are shown in Figure 8.

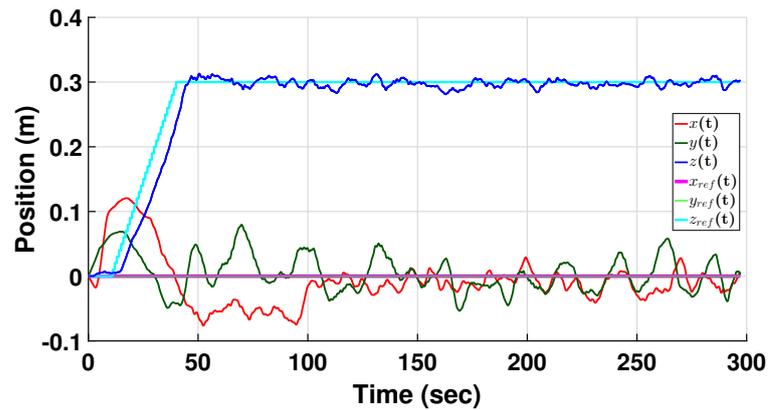


Figure 6. Numerical approximations of $x(t), y(t), |z(t)|$ and position references of the UAV when the OIC was used.

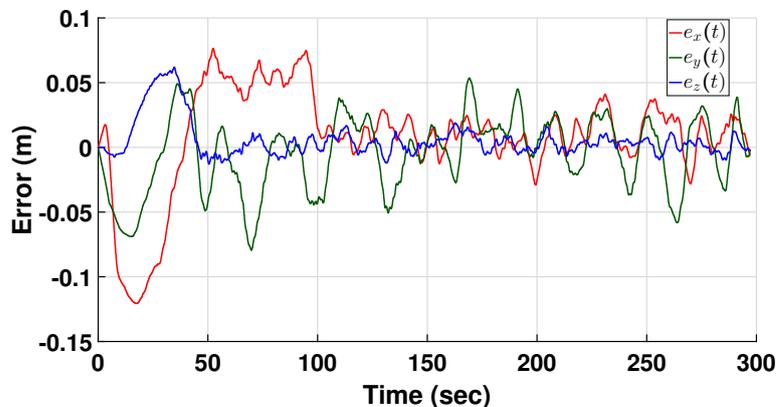


Figure 7. Numerical approximations of the position errors $e_x(t), e_y(t), e_z(t)$ when the OIC was applied to the UAV.

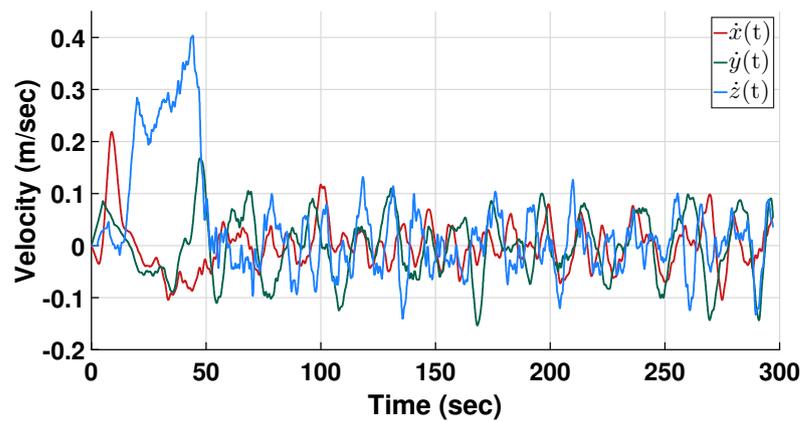


Figure 8. Numerical approximations of $\dot{x}(t)$, $\dot{y}(t)$, $\dot{z}(t)$ from UAV when the OIC was used.

The pitch, roll and yaw orientations are shown in Figure 9; angular rates for all orientation variables are shown in Figure 10.

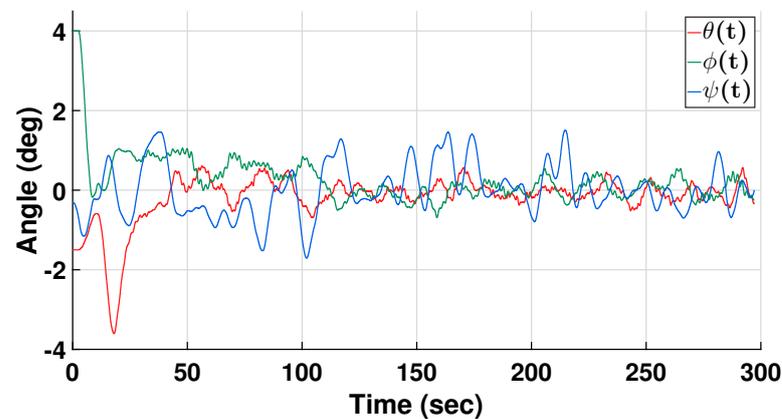


Figure 9. Numerical approximations of pitch, roll and yaw angles of the UAV when using the OIC.

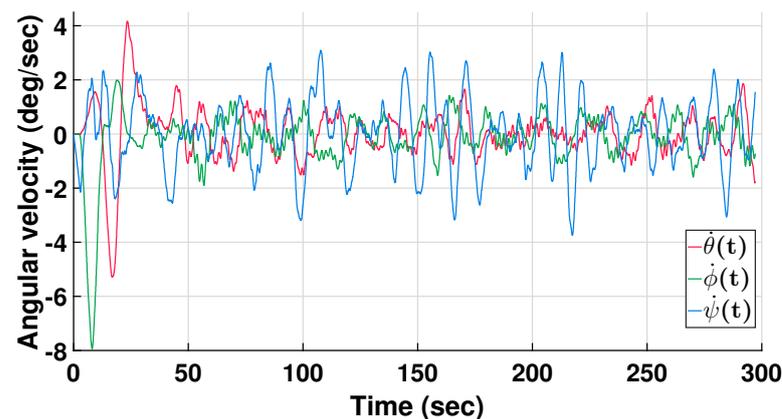


Figure 10. Numerical approximations of pitch, roll and yaw angular velocities of the UAV when applying the OIC.

The torque control signals are shown in Figure 11, the force control signal is shown in Figure 12, and these control signals are calculated using the integral control algorithm in the discrete quadcopter model.

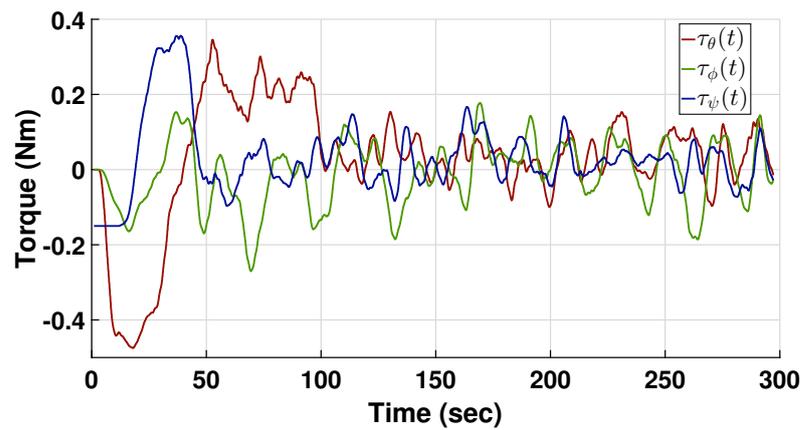


Figure 11. Numerical approximations of torque control signals using the OIC.

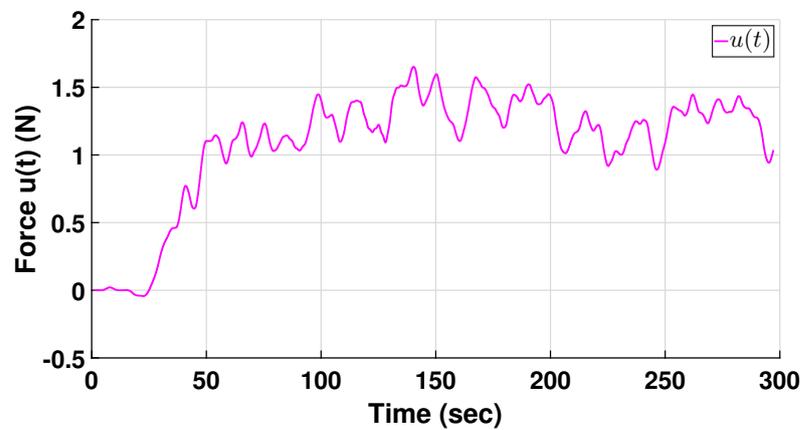


Figure 12. Numerical approximation of force control signal $u(t)$ using the OIC.

The quadrotor trajectory is shown in Figure 13; this trajectory was subdivided into three trajectories for takeoff, flight and landing of the UAV. In the blue line, one can observe the UAV’s takeoff, while the hover flight of the robot is represented in dark green, and the landing phase is visualized in light green. Each set of points signifies the division of the overall trajectory into these three phases: takeoff, flight, and landing.

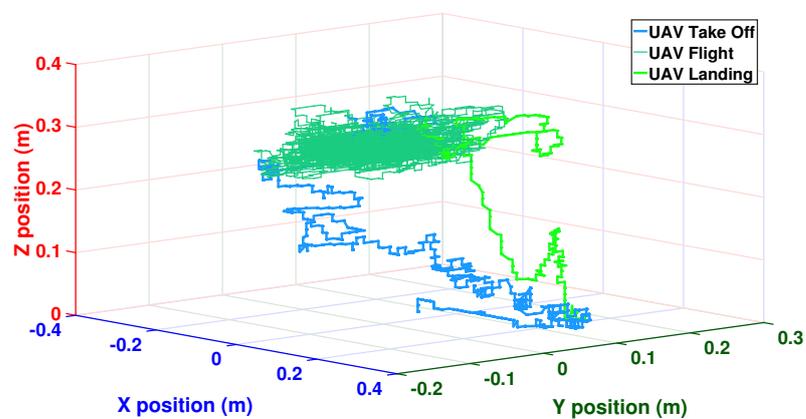


Figure 13. Full 3D path of the UAV using the OIC.

4.2. Robustness Test under Crosswind Conditions

Additional experiments were conducted to test the robustness of the integral control strategy. This experiment allows to evaluate how the robustness provided by the integral

action affects the UAV’s flight performance. So, a fan is used to apply a crosswind to aerial platform with a velocity of 4.3 m/s (at 19 °C). This external disturbance was supplied to the system from 50 to 150 s. These conditions were used for both PID and OIC controllers. Figure 14 shows the translational position behavior of the vehicle under this crosswind condition.

Moreover, Figure 15 shows the position errors experienced by the four-rotor rotorcraft, and Figure 16 depicts the control signals applied in real time. The magnitude of the control signals shows the feasibility of integral control. Finally, Figure 17 shows the force generated by the control signal $u(t)$ (corresponding to the collective throttle) when the flying platform was subjected to wind disturbances.

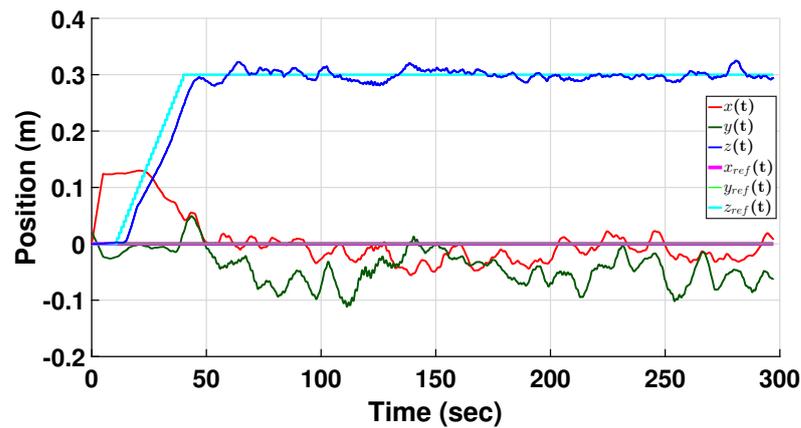


Figure 14. Numerical approximations of set-points and positions $x(t)$, $y(t)$, $|z(t)|$ of the quadcopter when a crosswind was applied.

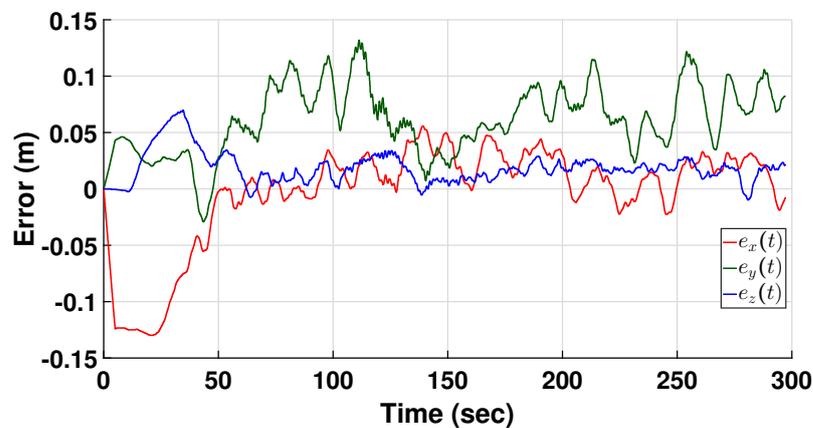


Figure 15. Numerical approximations of position errors of the quadcopter when it was disturbed.

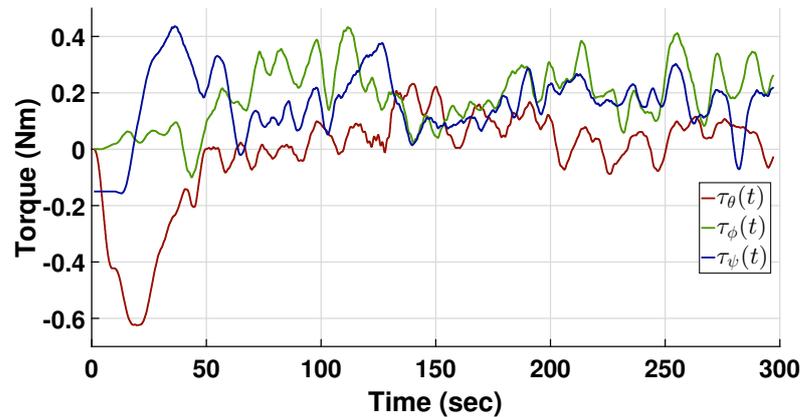


Figure 16. Numerical approximations of torque control signals using the OIC when the quadcopter was disturbed.

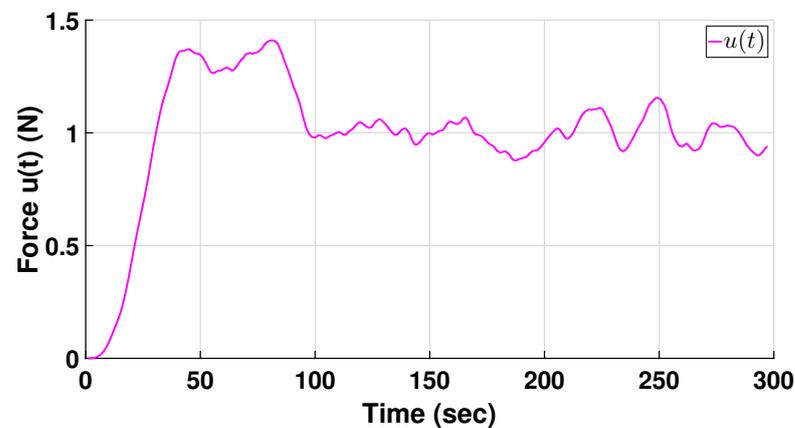


Figure 17. Force control signal $u(t)$ using the OIC applied when the quadcopter was disturbed.

4.3. Comparative Analysis of Energy Consumption between OIC and Conventional PID Controller

This experimental protocol enables a detailed evaluation of the efficacy of the integral control algorithm within an LQR control system applied to UAVs, providing empirical evidence of its performance, robustness and energy efficiency compared to a conventional PID controller. Total energy consumption was computed using potential and kinetic energies. This calculation was performed for both the tuned heuristic PID controller and the OIC optimal controller proposed in this research work. The total energy behavior for each scenario is illustrated in Figure 18. Furthermore, Table 1 presents the electrical energy savings comparison between the controllers. These data were derived from a series of 30 experiments.

As evident in Table 1, the OIC (Optimal Integral Controller) improved the electrical energy consumption by 53.05% compared to a conventional PID controller. This proves that the use of a LQR (Linear Quadratic Regulator) controller in conjunction with integral action can save energy in a four-rotor aerial robot. In addition, it provides robustness to the control loop, enabling it to absorb unmodeled dynamics or external disturbances, as has been demonstrated in Section 4.2. Although these experiments were conducted in a controlled environment, it is clear that the control algorithm will work efficiently in outdoor flights, proving that the robot state approximation is accurate and the robot model is correctly parameterized.

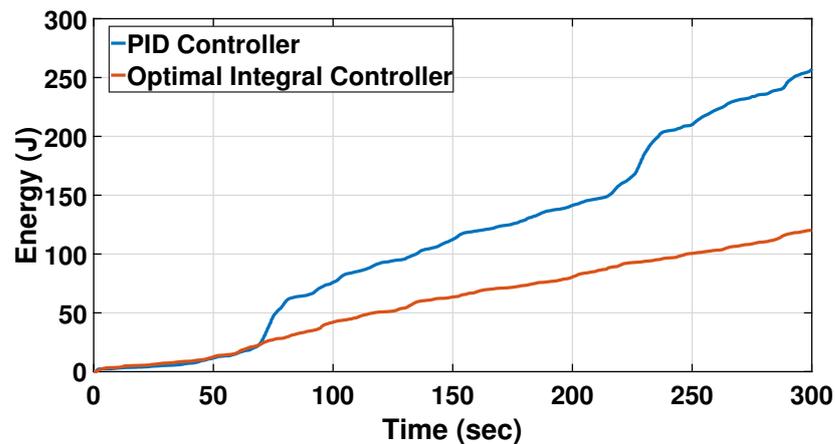


Figure 18. Total energy consumption comparison using a PID controller and the OIC.

Table 1. Energy saving.

Controller	Total Energy (kJ)	Saved Energy %
Proportional Integral Derivative	3.3718	0
Optimal Integral Controller	1.7888	53.0514

5. Conclusions

Satisfactory experimental results using integral control are obtained. So, can be stated that the Optimal Integral Controller (OIC) improves the LQR controller behavior, because it includes an integral term. Moreover, the tuning of integral action can be optimally conducted by solving the discrete Riccati Algebraic Equation associated to LQR problem, allowing to penalize the energy consumption and the convergence rate of the state. According to optimal control theory, the exact linearization applied to the altitude and yaw subsystems allows for stability of the closed loop, and then it is guaranteed. For the subsystems $y-\phi$ and $x-\theta$, although the optimal control obtained was synthesized using a linearized model, the experimental tests show the robustness of the controller, and the real-time results show an important energy savings rate. So using this strategy, it is possible to achieve proper UAV stabilization for both attitude and position. As future work, it is intended to apply this control strategy in external environments to verify the robustness and efficiency of the algorithm in outdoor flights. This will be carried out on a UAV with the same configuration as proposed in this investigation. Other future work includes experiments in a wind tunnel to evaluate the discharge time of the battery in order to compare the performance of the controllers more accurately.

Author Contributions: O.-J.S.-S. and H.R.-T. conceived the design of the control algorithm. G.G.-G. and O.G.-P. coded the control and navigation algorithm in the controller, built the prototype and carried out the experimental tests. O.-J.S.-S., H.R.-T. and G.G.-G. performed data validation, statistical analysis and writing of the article. H.R.-T. and O.-J.S.-S. conducted a critical review of the results. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The first autor is grateful to CONACyT for granting the scholarship to the CVU number: 552521.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

FNN	Fuzzy Neural Network
IMU	Inertial Measurement Unit
LQR	Linear Quadratic Regulator
OIC	Optimal Integral Controller
PC	Personal Computer
PID	Proportional Integral Derivative Controller
PWM	Pulse Width Modulation
RF	Radio Frequency
UAV	Unmanned Aerial Vehicle

Nomenclature

Table of variables, concepts, and units for UAV dynamics:

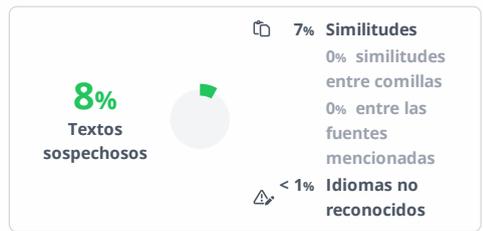
Variable	Concept	Units
x	Horizontal displacement in the x -axis	Meters (m)
y	Horizontal displacement in the y -axis	Meters (m)
z	Vertical position in the z -axis	Meters (m)
ψ	Yaw angle, rotation around the z -axis	Degrees (deg)
θ	Pitch angle, rotation around the y -axis	Degrees (deg)
ϕ	Roll angle, rotation around the x -axis	Degrees (deg)
u	Thrust force generated by the motors to lift the UAV	Newtons (N)
τ_ϕ	Torque generated around the x -axis	Newton meters (Nm)
τ_θ	Torque generated around the y -axis	Newton meters (Nm)
τ_ψ	Torque generated around the z -axis	Newton meters (Nm)
m	Mass	Kilograms (kg)
g	Gravity	Newton (N)
\bar{x}	System state	Depends on the system
\bar{u}	System control	Depends on the system
\bar{y}	Controlled output	Depends on the system
\bar{y}_R	Output reference	Depends on the system
D_x	State variable domain	Depends on the system
D_u	Control variable domain	Depends on the system
f	State function	Depends on the system
h	Output function	Depends on the system
x_{ss}	Steady-state	Depends on the system
u_{ss}	Steady-state control	Depends on the system
\bar{e}	Tracking error	Depends on the system
$\bar{\sigma}$	Accumulated error	Depends on the system
A, B, C	State, control and output matrices	Depends on the system
ξ	Adjusted state	Depends on the system
v	Adjusted control	Depends on the system
\mathcal{K}	Control gain matrix	Depends on the system
x_n	State variable	Depends on the system
$x_n(k)$	Discrete state variable at step k	Depends on the system
$x_n(k+1)$	Discrete state variable one step ahead	Depends on the system
$\xi_z(k)$	Adjusted state at step k	Depends on the system
A_m, B_m, C_m	State, control and output matrices of the m system	Depends on the system
\mathcal{J}_m	Performance index of the m system	Depends on the system
\mathcal{Q}_m	Weighting matrix for state of the m system	Depends on the system
\mathcal{R}_m	Weighting factor for control input for the m system	Depends on the system
$u_m(k)$	Control input at step k of the m system	Depends on the system
\mathcal{P}_m	Solution to the Riccati equation of the m system	Depends on the system

References

1. Dyer, E.; Sirouspour, S.; Jafarinasab, M. Energy Optimal Control Allocation in a Redundantly Actuated Omnidirectional UAV. In Proceedings of the International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 5316–5322.
2. Ding, Y.; Cao, R.; Liang, S.; Qi, F.; Yang, Q.; Yan, W. Density-based optimal UAV path planning for photovoltaic farm inspection in complex topography. Chinese Control And Decision Conference. In Proceedings of the International Conference on Robotics and Automation (ICRA), Chengdu, China, 14–16 June 2020; pp. 3931–3936.
3. Bucolo, M.; Buscarino, A.; Fortuna, L.; Gagliano, S. Generalizing the Letov formula for the discrete-time case. *Int. J. Dyn. Control* **2023**, *11*, 94–100. [[CrossRef](#)]
4. Khalil, H.K. *Nonlinear Systems*; Prentice Hall: Hoboken, NJ, USA, 2001.
5. Idrissi, M.; Salami, M.; Annaz, F. A review of quadrotor unmanned aerial vehicles: Applications, architectural design and control algorithms. *J. Intell. Robot. Syst.* **2022**, *104*, 22. [[CrossRef](#)]
6. Tahar, M.; Meguenni-Zemalache, K.; Omari, A. Control of an Under-Actuated X4-flyer using Integral Backstepping Controller. *Przełga Elektrotechniczny (Electr. Rev.)* **2011**, *87*, 251–256.
7. Bouabdallah, S.; Siegwart, R. Full Control of a Quadrotor. In Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA, 29 October–2 November 2007; pp. 153–158.
8. Al-Younes, Y.M.; Al-Jarrah, M.A.; Jhemi, A.A. Linear vs. Nonlinear Control Techniques for a Quadrotor Vehicle. In Proceedings of the 7th International Symposium on Mechatronics and Its Applications (ISMA10), Sharjah, United Arab Emirates, 20–22 April 2010; p. 10-1:10.
9. Adir, V.G.; Stoica, A.M. Integral LQR Control of a Star-Shaped Octorotor. *Incas Bull.* **2012**, *4*, 3–18.
10. Tang, Y.R.; Li, Y. Design of an Optimal Flight Control System with Integral Augmented Compensator for a Nonlinear UAV Helicopter. In Proceedings of the 10th World Congress on Intelligent Control and Automation, Beijing, China, 6–8 July 2012; pp. 3927–3932.
11. Antonio-Toledo, M.E.; Sanchez, E.N.; Alanis, A.Y.; Flórez, J.A.; Perez-Cisneros, M.A. Real-time integral backstepping with sliding mode control for a quadrotor UAV. *IFAC-PapersOnLine* **2018**, *51*, 549–554. [[CrossRef](#)]
12. Elkhatem, A.S.; Engin, S.N. Robust LQR and LQR-PI control strategies based on adaptive weighting matrix selection for a UAV position and attitude tracking control. *Alex. Eng. J.* **2021**, *61*, 6275–6292. [[CrossRef](#)]
13. Al-Isawi, M.M.; Attiya, A.J.; Adoghe, J.O. UAV Control Based on Dual LQR and Fuzzy-PID Controller. *Al-Khwarizmi Eng. J.* **2020**, *16*, 43–53. [[CrossRef](#)]
14. Rao, J.; Li, B.; Zhang, Z.; Chen, D.; Giernacki, W. Position Control of Quadrotor UAV Based on Cascade Fuzzy Neural Network. *Energies* **2022**, *15*, 1763. [[CrossRef](#)]
15. Zamani, A.; Etedali, S. Robust output feedback-based neuro-fuzzy controller for seismically excited tall buildings with ATMD accounting for variations in the type of supporting soil. *Soil Dyn. Earthq. Eng.* **2023**, *164*, 107614. [[CrossRef](#)]
16. Zamani, A.; Etedali, S. Seismic structural control using magneto-rheological dampers: A decentralized interval type-2 fractional-order fuzzy PID controller optimized based on energy concepts. *ISA Trans.* **2023**, *137*, 288–302. [[CrossRef](#)] [[PubMed](#)]
17. Sadegh, E.; Abbas-Ali, Z. Semi-active control of nonlinear smart base-isolated structures using MR damper: Sensitivity and reliability analyses. *Smart Mater. Struct.* **2022**, *31*, 065021
18. Castillo-Garcia, P.; Hernandez, L.E.M.; Gil, P.G. *Indoor Navigation Strategies for Aerial Autonomous Systems*; Butterworth-Heinemann: Oxford, UK, 2016.
19. Castillo, P.; Lozano, R.; Dzul, A.E. *Modelling and Control of Mini-Flying Machines*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2005.
20. Santos, O.; Romero, H.; Salazar, S.; Lozano, R. Discrete Optimal Control for a Quadrotor UAV: Experimental Approach. In Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Orlando, FL, USA, 27–30 May 2014; pp. 1138–1145.
21. Santos-Sánchez, O.; García, O.; Romero, H.; Salazar, S.; Lozano, R. Finite horizon nonlinear optimal control for a quadrotor: Experimental results. *Optim. Control Appl. Methods* **2021**, *42*, 54–80. [[CrossRef](#)]

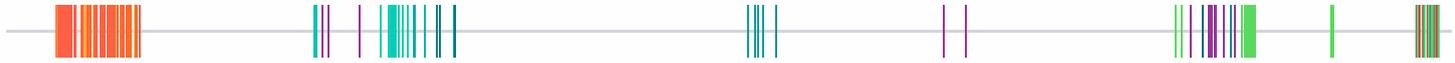
Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Tesis_Doctoral_Gildardo_Godinez_Garrido_o_Final__respaldo_(1)



Nombre del documento: Tesis_Doctoral_Gildardo_Godinez_Garrido_Final__respaldo_(1).pdf ID del documento: 8385975e52549e9adc903d8d7372beba444470e9 Tamaño del documento original: 16,81 MB	Depositante: Manuel Pineda Fecha de depósito: 11/3/2024 Tipo de carga: interface fecha de fin de análisis: 11/3/2024	Número de palabras: 47.671 Número de caracteres: 266.562
---	---	---

Ubicación de las similitudes en el documento:



Fuentes principales detectadas

Nº	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	upcommons.upc.edu https://upcommons.upc.edu/bitstream/2117/334704/2/TFG_Planos_-_Iván_Rodríguez_Grota.pdf 13 fuentes similares	85%		Palabras idénticas: 85% (33.949 palabras)
2	ru.dgb.unam.mx https://ru.dgb.unam.mx/bitstream/20.500.14330/TES01000766288/3/0766288.pdf 47 fuentes similares	2%		Palabras idénticas: 2% (2041 palabras)
3	oa.upm.es https://oa.upm.es/16634/1/04_HECTOR_ALONSO_MORENO_AVALOS.pdf 46 fuentes similares	2%		Palabras idénticas: 2% (2090 palabras)
4	dspace.ups.edu.ec https://dspace.ups.edu.ec/bitstream/123456789/7811/1/UPS-CT004660.pdf 39 fuentes similares	2%		Palabras idénticas: 2% (1913 palabras)
5	oa.upm.es https://oa.upm.es/71552/3/TFM_DIEGO_CERRILLO_VACAS_22.pdf 40 fuentes similares	1%		Palabras idénticas: 1% (1811 palabras)

Fuentes con similitudes fortuitas

Nº	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	repositorio.unican.es https://repositorio.unican.es/xmlui/bitstream/10902/8680/1/385653.pdf	< 1%		Palabras idénticas: < 1% (24 palabras)
2	ingenieria.ute.edu.ec https://ingenieria.ute.edu.ec/enfoqueute/index.php/revista/article/download/804/826	< 1%		Palabras idénticas: < 1% (24 palabras)
3	engineering.purdue.edu https://engineering.purdue.edu/HELPS/Publications/papers/MeiLuHuLee2005ICAR.pdf	< 1%		Palabras idénticas: < 1% (20 palabras)
4	link.springer.com Robust Self-Adjustable Path-Tracking Control for Autonomous ... https://link.springer.com/content/pdf/10.1007/s40815-020-00939-1.pdf	< 1%		Palabras idénticas: < 1% (14 palabras)
5	www.uaeh.edu.mx Maestría en Ciencias en Automatización y Control https://www.uaeh.edu.mx/campus/icbi/oferta/maestrias/mac/	< 1%		Palabras idénticas: < 1% (10 palabras)

Fuente ignorada Estas fuentes han sido retiradas del cálculo del porcentaje de similitud por el propietario del documento.

Nº	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	Tesis_Doctoral_Gildardo_Godinez_Garrido_Final__respaldo_(1).pdf Tes... #58c868 El documento proviene de mi biblioteca de referencias	100%		Palabras idénticas: 100% (47.671 palabras)

Fuentes mencionadas (sin similitudes detectadas) Estas fuentes han sido citadas en el documento sin encontrar similitudes.

1	https://ardupilot.org/copter/docs/common-telemetry-xbee.html
2	https://ardupilot.org/copter/docs/common-rplidar-a2.html
3	https://nanonets.com/blog/optical-flow/
4	https://www.sigmaelectronica.net/comunicacion-serial-raspberry-pi/
5	https://howtomechatronics.com/tutorials/arduino/arduino-and-mpu6050-accelerometer