



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS  
AVANZADOS DEL INSTITUTO POLITÉCNICO NACIONAL

**Unidad Zacatenco**

Programa de  
**Sistemas Autónomos de Navegación Aérea y Submarina**

**“Planeación de Trayectorias y Seguimiento  
Óptimo con un Cuadricóptero en Exteriores”**

TESIS

Que presenta

**ORLANDO GARCÍA PÉREZ**

Para obtener el grado de

**DOCTOR EN CIENCIAS**

En

**Sistemas Autónomos de Navegación Aérea y Submarina**

Directores de la Tesis:

**Dr. Sergio Rosario Salazar Cruz**

**Dr. Omar Jacobo Santos Sánchez**

Ciudad de México

Mayo, 2019.



## AGRADECIMIENTOS

Inicialmente quiero agradecer a mis padres y mis hermanas por todo su apoyo.

Agradezco profundamente a mis asesores al Dr. Sergio Salazar Cruz y al Dr. Omar Santos Sanchez, por su apoyo en la realización de esta tesis. Su dedicación en este proyecto de investigación fue indispensable desde el inicio hasta el final de la tesis. Agradezco también al Dr. Rogelio Lozano por todas sus observaciones para este proyecto, al Dr. Iván González, Dr. Hugo Romero, Dr. Patricio Ordáz, Dr. Aurélien Cabarbaye, gracias por todas sus aportaciones académicas.

También quiero agradecer a mis amigos del CINVESTAV por todo su apoyo personal y académico. A mis compañeros de Desarrollo Tecnológico DOI, Donovan Flores e Irving Rosales por toda la experiencia compartida. A mis alumnos de Servicio social José López, Pablo Hernández, Eduardo Mendoza, José Luis Torres y Héctor Huerta por todo el apoyo recibido.

Agradezco al CINVESTAV y al Laboratorio UMI-LAFMIA por darme la oportunidad de ser parte de este programa doctoral.

Finalmente quiero externar un profundo agradecimiento al Consejo Nacional de Ciencia y Tecnología, por el apoyo económico recibido durante todo el programa de estudio.





## RESUMEN

La tarea de navegación utilizando vehículos aéreos es aún un gran reto tanto en la parte tecnológica como en la parte de control, puesto que involucra otras tareas como el reconocimiento del entorno, la toma de decisiones, la detección y evasión de obstáculos, además de la planeación y seguimiento de trayectorias. Existen varias maneras de abordar este problema, la diferencia se basa principalmente en el tipo y cantidad de sensores que se utilizan. En este trabajo de tesis se aborda la navegación en presencia de obstáculos, específicamente árboles, es decir en un entorno real donde existen problemas por perturbaciones externas, limitaciones en los sensores, etc. La principal aportación de este trabajo es la implementación de un control subóptimo no lineal discreto de horizonte finito. Esta técnica se utilizó para hacer el seguimiento de las trayectorias generadas por un algoritmo de planeación y evasión de obstáculos con enfoque geométrico. La diferencia con otros trabajos es que aquí se implementó el algoritmo en tiempo real, sobre una computadora abordo de la aeronave y sin el conocimiento previo de las restricciones geométricas.

Además se propusieron otros algoritmos y esquemas de control los cuales fueron implementados en tiempo real para seguimiento de trayectorias, como el control basado en backstepping y modos deslizantes terminales o el control subóptimo no lineal finito discreto combinado con un control basado en el algoritmo super-twisting. Esto tuvo gran importancia, ya que el algoritmo propuesto de planeación y evasión de obstáculos actualiza la trayectoria en cada instante de tiempo. Si la ruta está libre de obstáculos la trayectoria a seguir es una recta, pero esto es un caso ideal, en la realidad conforme el vehículo avanza encuentra objetos que le impiden continuar con la ruta original. Al final se eligió el control subóptimo no lineal discreto de horizonte finito para hacer toda la tarea de navegación. Se obtuvieron muy buenos resultados experimentales comparados con los que existen actualmente reportados en la literatura especializada y tomando en cuenta que la plataforma experimental utilizada se construyó integrando una microcomputadora con diversos sensores lo cual forma parte del aporte tecnológico que se realizó.



## ABSTRACT

The task of navigation using aerial vehicles is still a great challenge in the technological part and in the control part, since it involves other tasks such as the recognition of the environment, decision making, detection and collision avoidance, in addition to the planning and tracking of trajectories. There are several ways to address this problem, the difference is mainly based on the type and quantity of sensors that are used. In this thesis the navigation is addressed in the presence of obstacles, specifically trees, that is, in a real environment where there are problems due to external disturbances, limitations in the sensors, etc. The main contribution of this work is the implementation of a suboptimal finite discrete non-linear control. This technique was used to track the trajectories generated by an algorithm of planning and collision avoidance with geometric approach. The difference with other works is that here the algorithm was implemented in real time, on a computer aboard the aircraft and without prior knowledge of geometric constraints.

In addition, other algorithms and control schemes were proposed, which were implemented in real time for tracking trajectories, such as a control based on backstepping and terminal sliding modes or suboptimal finite discrete non-linear control combined with a control based on the super-twisting algorithm. This was of great importance, since the proposed obstacle avoidance and planning algorithm updates the trajectory at each instant of time. If the route is free of obstacles the path to follow is a straight line, but this is an ideal case, in reality as the vehicle goes forward it finds objects that prevent it from continuing with the original route. At the end, the suboptimal finite discrete non-linear control was chosen to do the whole navigation task. Very good experimental results were obtained compared with the results that currently exist reported in the specialized literature and taking into account that the experimental platform used was built by integrating a micro computer with various sensors which is part of the technological contribution that was made.



# ÍNDICE GENERAL

AGRADECIMIENTOS	iii
RESUMEN	v
ABSTRACT	vii
ÍNDICE GENERAL	ix
<b>1 INTRODUCCIÓN GENERAL</b>	<b>1</b>
1.1 Revisión bibliográfica . . . . .	2
1.2 Objetivo general . . . . .	6
1.2.1 Objetivos particulares . . . . .	6
1.3 Justificación . . . . .	6
1.4 Planteamiento del problema . . . . .	7
1.5 Lista de publicaciones . . . . .	8
<b>2 ALGORITMOS DE CONTROL</b>	<b>9</b>
2.1 Introducción . . . . .	9
2.2 Modelo del cuadricóptero . . . . .	10
2.3 Control óptimo de horizonte infinito en tiempo discreto . . . . .	11
2.3.1 Subsistema $z$ . . . . .	11
2.3.2 Control óptimo del subsistema $\psi$ . . . . .	13
2.3.3 Control óptimo del subsistema $y - \varphi$ . . . . .	14

2.3.4	Control óptimo del subsistema $x - \theta$ . . . . .	16
2.4	Control Óptimo de horizonte finito en tiempo discreto . . . . .	17
2.4.1	Subsistema $z$ . . . . .	17
2.4.2	Subsistema $\psi$ . . . . .	19
2.4.3	Subsistema $y - \varphi$ . . . . .	20
2.4.4	Subsistema $x - \theta$ . . . . .	22
2.5	Control subóptimo no lineal para sistemas afines en tiempo discreto de horizonte finito . . . . .	24
2.5.1	Control subóptimo discreto no lineal de horizonte finito aplicado al cuadricóptero . . . . .	26
2.6	Algoritmo de control super-twisting con Control subóptimo no lineal . . . . .	34
2.6.1	Algoritmo super-twisting aplicado al cuadricóptero . . . . .	35
2.7	Algoritmo de control Backstepping . . . . .	39
2.7.1	Algoritmo de backstepping y modos deslizantes terminales aplicado a un Cuadricóptero . . . . .	42
2.8	Conclusiones . . . . .	54
<b>3</b>	<b>DISEÑO Y DESARROLLO DEL PROTOTIPO EXPERIMENTAL</b>	<b>55</b>
3.1	Introducción . . . . .	55
3.2	Autopiloto Pixhawk . . . . .	57
3.3	GPS . . . . .	60
3.4	Px4 Flow . . . . .	62
3.5	Sensor láser Sweep 360 . . . . .	63
3.6	Cámara Pixy . . . . .	64
3.7	Conclusiones . . . . .	66
<b>4</b>	<b>RESULTADOS</b>	<b>67</b>
4.1	Introducción . . . . .	67
4.2	Control óptimo de horizonte infinito en tiempo discreto . . . . .	67

4.2.1	Prueba 1: Control en altura y posición . . . . .	67
4.2.2	Prueba 2: Seguimiento de trayectoria . . . . .	73
4.3	Control óptimo de horizonte finito, caso discreto . . . . .	74
4.3.1	Prueba 1: Seguimiento de trayectorias en exteriores . . . . .	74
4.4	Control subóptimo no lineal discreto de horizonte finito . . . . .	78
4.4.1	Prueba 1: Seguimiento de una trayectoria . . . . .	78
4.5	Algoritmo de control Super-twisting combinado con el control subóptimo no lineal . . . . .	84
4.5.1	Prueba 1: Estimación de perturbaciones externas y seguimiento de trayectoria . . . . .	84
4.6	Despegue y aterrizaje autónomo utilizando visión artificial . . . . .	91
4.6.1	Prueba 1: Simulación del despegue y aterrizaje . . . . .	91
4.6.2	Prueba 2: Despegue y aterrizaje, experimental . . . . .	93
4.7	Resultados experimentales utilizando el control no lineal basado en Backstepping	96
4.8	Navegación en exteriores utilizando un sensor de barrido láser . . . . .	104
4.8.1	Adquisición y tratamiento de los datos . . . . .	106
4.8.2	Algoritmo de repulsión . . . . .	110
4.8.3	Algoritmo de planeación y evasión de obstáculos . . . . .	111
4.8.4	Resultados experimentales . . . . .	114
4.9	Conclusiones . . . . .	120
<b>5</b>	<b>CONCLUSIONES Y TRABAJO FUTURO</b>	<b>123</b>
5.1	Conclusiones . . . . .	123
5.2	Trabajo futuro . . . . .	125
	<b>BIBLIOGRAFÍA</b>	<b>127</b>





# CAPÍTULO 1

## INTRODUCCIÓN GENERAL

Actualmente existen muchas investigaciones en el área de vehículos aéreos, muchos se centran en la parte aerodinámica, en la capacidad de carga o en seguimiento de trayectorias, este último es común ya que la navegación es una de las tareas más importantes en esta clase de vehículos. En esta tesis se aborda la tarea de navegación basada en sensores es decir sin conocimiento de las restricciones (obstáculos) que el vehículo puede encontrarse durante su trayectoria.

Este trabajo de investigación puede dividirse en dos grandes fases, en la primera se implementaron varios algoritmos de control para hacer seguimiento de trayectorias. El objetivo de esta fase es tener varias técnicas de control que se puedan aplicar al vehículo durante la navegación. La segunda fase es la implementación de alguna de las estructuras de control propuestas para hacer la tarea de seguimiento pero calculando la trayectoria en tiempo real, para ello se utiliza un sensor láser que hace un barrido en 360 grados el cual provee un mapa en  $2D$  lo cual da la posibilidad de que ambas tareas, la de seguimiento y planeación de la trayectoria pueden llevarse a cabo en tiempo real y de manera autónoma. La planeación de la trayectoria sigue un conjunto de reglas el cual involucra una zona segura similar a [34] además de un punto de inicio (origen) y un punto de llegada (destino) como en [18], la zona segura es aquella en la cual el vehículo no puede acercarse más cuando detecta un obstáculo, esta zona

está representada con una circunferencia virtual alrededor del obstáculo. La trayectoria se genera automáticamente de acuerdo a los datos obtenidos por el sensor láser, esta trayectoria sigue la forma de la circunferencia y si otro obstáculo aparece en la navegación se vuelve a repetir el procedimiento. Es decir una nueva circunferencia rodeará al obstáculo, y así hasta llegar al destino final.

## 1.1 Revisión bibliográfica

Los trabajos que se han realizado en esta área abarcan distintos enfoques como en [18] donde utilizan visión estereoscópica para poder detectar obstáculos. La planeación de las trayectorias se realizaron con un método probabilístico basado en el algoritmo de *D\* Lite*. El procedimiento comienza con dos puntos uno de inicio y otro de llegada, el vehículo empieza a moverse hacia el punto final en línea recta, durante el camino va detectando los obstáculos y va actualizando la trayectoria original. Este enfoque fue implementado en un software de simulación junto con una plataforma cableada. Otro trabajo representativo es [10] donde se aborda el problema de seguimiento de una trayectoria con un UAV pero con la capacidad de evitar obstáculos en movimiento o que van apareciendo durante la ruta. Para lograr esta tarea se utilizó un control Lógico Difuso el cual activaban cuando aparecía un obstáculo, para ello siempre debían conocer la posición del vehículo a través del GPS. En este trabajo se obtuvieron resultados de simulación y los obstáculos fueron dados de manera aleatoria. En [32] se propuso un modo de control dual el cual llaman "modo seguro" y "modo de peligro" aplicado a un UAV. En el modo seguro el vehículo navega sin obstáculos, en este modo se generan trayectorias utilizando estrategias de optimización, en el modo de peligro se utiliza un control basado en un Modelo de Control Predictivo. La evasión de obstáculos se realiza con un algoritmo modificado de Redes Neuronales de Grossberg el cual proponen en el artículo.

Un artículo similar es [34] donde se abordó la evasión de obstáculos con múltiples robots móviles, se utilizaron datos de un sensor láser, estos datos se organizaron en tuplas de dis-

tancias y ángulos con los cuales se definieron áreas de análisis tanto en el ángulo máximo de visión como el rango máximo de distancia. Los datos obtenidos se procesaron siguiendo aspectos como el *umbral de distancia*, *una zona segura*, *repulsión de obstáculos*, *atracción hacia el punto final de destino* y *la seguridad de navegación entre cada robot móvil*. El resultado final de este procedimiento fue la obtención de un criterio de optimización el cual definía el ángulo óptimo al cual debía moverse el robot en presencia de un obstáculo. Se presentaron únicamente resultados de simulación.

Lo que se puede observar es que la tarea de navegación utilizando vehículos aéreos se aborda de distintas maneras, esto depende en gran medida de los sensores utilizados por ejemplo en [22] se desarrolló un esquema de control para evasión de obstáculos de manera autónoma en un vehículo de 4 rotores, se utilizaron estrategias de control lineal y no lineal, así como modelos de control adaptables y estrategias de control backstepping. Para el seguimiento traslacional se utilizó el sistema de cámaras V-con, el algoritmo de planeación de trayectorias se realizó en una estación en tierra. En [1] se implementó un algoritmo híbrido utilizando SLAM y visión estéreo. La unión de estos algoritmos permitió obtener un mapa de 3D. Finalmente, para la planificación de trayectorias seguras se utilizó el algoritmo RTT en un espacio tridimensional. El algoritmo se implementó en un cuadricóptero, los datos se recopilaron haciendo vuelos en el exterior y finalmente se analizaron fuera de línea en un banco de datos “KITTI” para odometría.

Otro trabajo similar donde se utilizó visión estereoscópica para hacer navegación con UAV's es [5], donde se utilizaron librerías de Open CV para el análisis de las disparidades entre puntos característicos. Este análisis se realizó sólo para una distancia de profundidad lo cual permitió analizar imágenes a alta velocidad. Este algoritmo se implementó en un UAV pequeño que pudo detectar y evadir obstáculos mientras volaba a una velocidad de hasta  $20\text{ m/s}$ . Otros trabajos incluye métodos de optimización para encontrar una ruta sin obstáculos como en [7], donde presentaron un método para planificar trayectorias en en-

tornos conocidos. El algoritmo de control propuesto fue una fusión de técnicas basadas en optimización. Los resultados experimentales se obtuvieron utilizando el sistema de captura de movimiento Optitrack para posicionamiento. El control se calculó en una estación en tierra.

También hay otros trabajos que realizaron una combinación de uno o más sensores para detectar obstáculos como en [3] donde se usó un sensor de imagen para estimar la posición de los obstáculos y un sensor ultrasónico para detectar la distancia real de los puntos detectados, este sistema se implementó en un vehículo de 4 rotores utilizando un control PID. Estos sensores de imagen se han convertido en una solución cuando se requieren detectar objetos bajo ciertas condiciones de iluminación, en [6] también se utilizó visión estereoscópica. Este método permitió construir un mapa de disparidades que se utilizó para obtener las medidas de profundidad reales de los obstáculos. El algoritmo se ejecutó en una computadora integrada Raspberry Pi y se implementó en un cuadricóptero. Los resultados experimentales se obtuvieron en un ambiente interior donde se usó un Control PID para seguir las trayectorias sólo en la dinámica rotacional del vehículo.

En [26] se desarrolló un algoritmo de evasión de obstáculos para un vehículo de 4 rotores, utilizando un sensor ultrasónico. El proceso consistió en el seguimiento de una ruta preprogramada, una vez que el sensor ultrasónico detectaba una distancia de posible colisión, el UAV permanecía en vuelo estacionario para después escanear la zona. Esto se realizó con un servo-motor que hacía girar el sensor ultrasónico, con el fin de mapear la zona de colisión y generar una ruta elipsoidal para evadir el obstáculo, mediante un algoritmo de generación de rutas. La información obtenida fue procesada en un Arduino UNO. Se realizaron experimentos con resultados satisfactorios en el proceso de evasión. En [33] se propuso un modelo de evasión de obstáculos basado en el uso de una cámara monocular con una red neuronal adaptativa en línea, la cual fue utilizada para la estimación de profundidad real de los objetos detectados por la cámara monocular. El procesamiento se realizó en una estación

externa equipada con un procesador Intel Xeon @ 3.6 Ghz y 2 NVIDIA GM200. El resultado fue la lectura y operación de datos para la estimación de profundidad en ambientes estáticos y volúmenes cerrados, esto permitió obtener una ruta libre de obstáculos para vuelo de UAV's.

En [31] se formuló un control simple no lineal para la evasión de obstáculos, basado en aproximaciones geométricas en un entorno estático. El modelo consideró aproximaciones geométricas y cinemáticas del modelo del cuadricóptero propuesto. Se generó un algoritmo de evasión de obstáculos con los datos del seguimiento de la ruta, del error en los ángulos de navegación de la aeronave y las distancias relativas al obstáculo. El algoritmo fue validado de manera efectiva mediante simulaciones numéricas.

Este trabajo de tesis está enfocado a la navegación en exteriores en presencia de obstáculos, la mayoría de los trabajos aquí presentados únicamente muestran resultados de simulación. Algunos utilizan bancos de datos para simular el comportamiento de un UAV en ambientes con obstáculos. Otros resuelven el problema de la generación de rutas fuera de línea debido al alto costo computacional. También hay trabajos que utilizan redes neuronales para la tarea de detección y evasión de obstáculos así como planeación de nuevas rutas. La diferencia principal con estos trabajos es que esta tesis aborda la parte experimental en ambientes no controlados, específicamente la navegación en presencia de árboles. Es decir no se conoce previamente la posición de los obstáculos como en otros trabajos, además el algoritmo propuesto es capaz de detectar objetos en movimiento ya que todo el procedimiento se realiza en tiempo real.

## 1.2 Objetivo general

Planear y seguir trayectorias que permitan la navegación en presencia de obstáculos con un cuadricóptero utilizando un sensor de barrido láser.

### 1.2.1 Objetivos particulares

- Estimar la posición de los obstáculos para evitar colisiones mediante las mediciones del sensor láser sweep v1
- Planear una trayectoria que involucre las restricciones estimadas a partir del análisis de cada obstáculo.
- Desarrollar una técnica de control para seguir la ruta encontrada mediante un algoritmo basado en optimización.

## 1.3 Justificación

En este proyecto se aborda la planeación y el seguimiento de trayectorias con un vehículo tipo cuadricóptero basado en sensores, sin embargo se desea como resultado final un vehículo que pueda evitar obstáculos mientras se mueve a una referencia final pero con un costo computacional reducido. Existen trabajos que abordan este enfoque pero a diferencia de este trabajo utilizan otras técnicas que requieren mayor costo computacional como en [18] donde hacen navegación de un vehículo aéreo utilizando visión estereoscópica, o en [32] que utilizan redes neuronales para la evasión de obstáculos. El beneficio del método que se plantea en este trabajo es su simplicidad en el procesamiento de datos lo cual impacta directamente en el hardware utilizado. Esto combinado con técnicas de control óptimas y robustas permiten disminuir el peso, la energía y a su vez aumentar la velocidad de procesamiento y la precisión durante la misión aérea.

## 1.4 Planteamiento del problema

En la actualidad existen drones comerciales que realizan tareas de seguimiento de trayectorias, y con la integración de distintos sensores de posicionamiento pueden evadir obstáculos durante su recorrido, sin embargo cuando se habla de navegación aún hay tareas sin resolver. Una de ellas es la navegación, pues esta tarea involucra aspectos como el reconocimiento del entorno, la toma de decisiones, la capacidad de generar nuevas rutas de vuelo en tiempo real. La complejidad de este tema atrae a muchos investigadores para la generación de algoritmos que puedan ser capaces de optimizar factores como la distancia, el tiempo, el consumo energético y la precisión durante el vuelo.

## 1.5 Lista de publicaciones

### Revistas internacionales

- Omar Santos, Hugo Romero, Sergio Salazar, Orlando García-Pérez, y Rogelio Lozano. “Optimized discrete control law for quadrotor stabilization: Experimental results”. *Journal of Intelligent Robotic Systems*, pages 1-15, 2016. **Publicado**
- Omar Santos, Orlando García, Hugo Romero, Sergio Salazar, y Rogelio Lozano. “Finite horizon nonlinear optimal control for a quadrotor: experimental results”. *ISA Transactions*, Diciembre, 2018. **Sometido**
- Aurélien Cabarbaye, Orlando García, Donovan Flores. “Design and model of an advance propeller well suited for multi-rotor applications”. *International Journal of Micro Air Vehicles*, 2019. **Sometido**
- Iván González, Orlando García, Omar Santos, Sergio Salazar, y Rogelio Lozano. “Robust and suboptimal control for an uav: Control synthesis and experimental validation under cross wind disturbances”. 2018. **En revisión**
- Orlando García, Patricio Ordáz, Omar J. Santos-Sánchez, Sergio Salazar, y Rogelio Lozano. “Backstepping and robust control for a quadrotor in outdoor environment: an experimental approach”. *IEEE Access*, Marzo, 2019. **Publicado**

### Congresos internacionales

- Orlando García, Omar Santos, Hugo Romero y Sergio Salazar. “On the tracking trajectory using optimal control in a quadrotor helicopter: Experimental results”. *In 2015 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, pages 142-151. *IEEE*, 2015.
- Orlando García, Donovan Flores, Omar Santos, Hugo Romero, Sergio Salazar, y Rogelio Lozano. “Autonomous take-off and landing on a colored platform”. *In Unmanned Aircraft Systems (ICUAS)*, 2017. *International Conference on* (pp. 877-884). *IEEE*.



# CAPÍTULO 2

## ALGORITMOS DE CONTROL

### 2.1 Introducción

En este Capítulo se abordan los diferentes algoritmos de control que se implementaron en el cuadricóptero. En la sección 2.3 se hace uso de un algoritmo de control óptimo de horizonte infinito el cual se aplica al modelo del cuadricóptero. La implementación se realiza sobre el modelo reducido, como en [21], posteriormente se divide en 4 subsistemas y se aplica un control linealizante sobre el subsistema de  $z$ . En la segunda sección 2.4 se realiza algo similar, pero a diferencia del algoritmo anterior éste se obtiene de horizonte finito, lo cual impacta en el hecho de que permite modificar el tiempo de convergencia acorde a la planta estudiada. La síntesis del control de estas dos subsecciones se realizó en la tesis de maestría [14], aquí sólo se aborda la implementación directamente en el cuadricóptero en el exterior.

En la sección 2.5 se muestra otro enfoque basado en optimización; se propone la síntesis de un control en un sentido subóptimo, utilizando el enfoque de programación dinámica. La síntesis de control se realiza sin quitar las no linealidades que se presentan en el modelo reducido, lo cual nos permite tener un mejor desempeño en el control, puesto que contempla algunos términos que a diferencia de los enfoques anteriores podrían estarse perdiendo.

Otra estrategia utilizada se describe en la sección 2.6 en el cual se combinaron dos técnicas de control. El objetivo es hacer una estimación de posibles perturbaciones externas con el método de mínimos cuadrados y decidir si aplicar un control robusto basado en el algoritmo de Super-twisting ó bien un algoritmo basado en el control subóptimo no lineal estudiando en la sección 2.5.

En la sección 2.7 se hace uso de un algoritmo de control basado en Backstepping, se propone una modificación en la entrada virtual, lo cual ayuda en la síntesis del control específicamente para el modelo del cuadricóptero. Este algoritmo se implementó en un UAV combinado con un algoritmo basado en modos deslizantes terminales, este algoritmo se implementó bajo la dirección del Dr. Patricio Ordaz a quien se le agradece por su colaboración.

## 2.2 Modelo del cuadricóptero

El modelo matemático del cuadricóptero se tomó de [21], éste se utilizará en las siguientes secciones 2.3, 2.4 y 2.5. Este modelo puede dividirse en subsistemas como en [8] con el objetivo de hacer más fácil el control.

$$m\ddot{x} = -u \sin \theta \tag{2.1}$$

$$m\ddot{y} = u \cos \theta \sin \varphi$$

$$m\ddot{z} = u \cos \theta \cos \varphi - mg$$

$$\ddot{\varphi} = \tau_\varphi$$

$$\ddot{\theta} = \tau_\theta$$

$$\ddot{\psi} = \tau_\psi,$$

donde,  $m$  es la masa del vehículo,  $x$  y  $y$  son las coordenadas en el plano horizontal,  $z$  es la coordenada en el plano vertical, es decir la altura del vehículo respecto de la tierra.  $\varphi$  es el ángulo de alabeo (roll),  $\theta$  es el ángulo de cabeceo (pitch),  $\psi$  es el ángulo de guiñada (yaw),

$\tau_\varphi$ ,  $\tau_\theta$ ,  $\tau_\psi$  son los momentos para cada ángulo, y  $u$  es la entrada de control que mantiene a la planta en la posición vertical.

## 2.3 Control óptimo de horizonte infinito en tiempo discreto

En esta sección se implementa un control óptimo discreto de horizonte infinito combinado con una linealización exacta, el objetivo es aplicarlo para estabilización del cuadricóptero para vuelo en exteriores, esta técnica de control se desarrolló en la tesis de maestría [14]. Una de las principales ventajas de utilizar una linealización exacta es que permite fijar de manera más fácil las matrices  $Q$  y  $R$  asociados al problema del LQR, puesto que la expresión de control no lineal aún conserva partes no lineales del modelo utilizado.

El modelo 2.1 puede ser estabilizado de manera óptima con el control basado en el algoritmo del LQR combinado con una linealización exacta. Primero se estabilizan las dinámicas de altura y ángulo de guiñada respectivamente, posteriormente las dinámicas acopladas  $y - \varphi$  y  $x - \theta$ .

### 2.3.1 Subsistema $z$

Sea el subsistema de altura el cual se muestra en 2.1,

$$m\ddot{z} = u \cos \theta \cos \varphi - mg, \quad (2.2)$$

el cual puede escribirse en la representación de espacio estado, con  $x_{1,z} = z$  y  $x_{2,z} = \dot{z}$ :

$$\begin{aligned} \dot{x}_{1,z} &= x_{2,z} \\ \dot{x}_{2,z} &= -g + (\cos \theta \cos \varphi) \frac{u}{m}, \end{aligned}$$

cuya representación en el dominio discreto puede obtenerse utilizando la aproximación de Euler,

$$\begin{aligned} x_{1,z}(k+1) &= T x_{2,z}(k) + x_{1,z}(k) \\ x_{2,z}(k+1) &= T \left( \frac{\cos \theta(k) \cos \varphi(k)}{m} \right) u(k) + x_{2,z}(k) - Tg, \end{aligned} \quad (2.3)$$

donde  $T$  es el tiempo de muestreo y se propone el siguiente control por linealización exacta que se sustituirá en el subsistema (2.3):

$$u(k) = m(u_1(k) + g) (\cos \theta(k) \cos \varphi(k))^{-1}, \quad (2.4)$$

con  $\cos \theta(k) \cos \varphi(k) \neq 0$ , y  $\theta, \varphi \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$ . Entonces, al sustituir la ley de control (2.4) en el subsistema (2.3) se obtiene:

$$x_z(k+1) = A_z x_z(k) + B_z u_1(k), \quad (2.5)$$

donde el par de matrices  $A_z = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$ ,  $B_z = \begin{bmatrix} 0 \\ T \end{bmatrix}$  son controlables. Entonces el objetivo es hallar un control óptimo de horizonte infinito para el sistema (2.5) tal que minimice el índice de desempeño siguiente:

$$J_z = \sum_{k=1}^{\infty} (x_z^T(k) Q_z x_z(k) + u_1^2(k) R_z), \quad (2.6)$$

donde las matrices  $Q_z \geq 0$ ,  $R_z > 0$  penalizan la convergencia del estado y el consumo de energía respectivamente. Se desea obtener una ley de control  $u_1(k)$ , tal que minimice al índice  $J_z$  sujeto al sistema lineal (2.5). El par  $(A_z, B_z)$  es controlable, entonces la ecuación algebraica discreta de Riccati está dada como en [20], esta ecuación se resuelve utilizando el software Matlab [4]:

$$P_z = A_z^T P_z A_z - A_z^T P_z B_z (R_z + B_z^T P_z B_z)^{-1} B_z^T P_z A_z + Q_z. \quad (2.7)$$

El cual tiene una única solución y este define la secuencia óptima:

$$u_1^*(k) = -(R_z + B_z^T P_z B_z)^{-1} B_z^T P_z A_z x_z^*(k), \quad \forall k \geq 0. \quad (2.8)$$

De acuerdo a la teoría dada en [20] el sistema (2.5) en lazo cerrado y con la ley de control óptima (2.8) es estable y minimiza el índice de desempeño (2.6).

### 2.3.2 Control óptimo del subsistema $\psi$

Consideramos para el sistema del ángulo  $yaw$ ,

$$\ddot{\psi} = \tau_\psi.$$

Se define  $x_{1,\psi} = \psi$  y  $x_{2,\psi} = \dot{\psi}$  y la representación en espacio estado queda de la siguiente manera:

$$\dot{x}_{1,\psi} = x_{2,\psi}$$

$$\dot{x}_{2,\psi} = \tau_\psi,$$

también se obtiene su representación en el dominio discreto,

$$x_\psi(k+1) = A_\psi x_\psi(k) + B_\psi \tau_\psi(k),$$

donde  $A_\psi = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$ ,  $B_\psi = \begin{bmatrix} 0 \\ T \end{bmatrix}$  son controlables. Se define el siguiente índice de desempeño.

$$J_\psi = \sum_{k=1}^{\infty} (x_\psi^T(k) Q_\psi x_\psi(k) + \tau_\psi^2(k) R_\psi), \quad (2.9)$$

donde las matrices  $Q_\psi \geq 0$  y  $R_\psi > 0$ . La ley de control óptima queda definida como:

$$\tau_\psi^*(k) = -(R_\psi + B_\psi^T P_\psi B_\psi)^{-1} B_\psi^T P_\psi A_\psi x_\psi^*(k), \forall k \geq 0.$$

La matriz  $P_\psi$  satisface la ecuación algebraica discreta de Riccati: (DARE, por sus siglas en inglés) y es similar a (2.7).

### 2.3.3 Control óptimo del subsistema $y - \varphi$

Sea el subistema  $y - \varphi$ , el cual describe la traslación sobre el eje  $y$  y el ángulo de *roll*,

$$m\ddot{y} = u \cos \theta \sin \varphi \quad (2.10)$$

$$\ddot{\varphi} = \tau_\varphi. \quad (2.11)$$

Obtenemos inicialmente su representación en espacio estado, con ( $x_{1y} = y$ ,  $x_{2y} = \dot{y}$ ,  $x_{3\varphi} = \varphi$ ,  $x_{4\varphi} = \dot{\varphi}$ ):

$$\begin{aligned} \dot{x}_{1y} &= x_{2y} \\ \dot{x}_{2y} &= \frac{1}{m} u \cos \theta \sin x_{3\varphi} \\ \dot{x}_{3\varphi} &= x_{4\varphi} \\ \dot{x}_{4\varphi} &= \tau_\varphi. \end{aligned}$$

También obtenemos su representación en el dominio discreto:

$$\begin{aligned} x_{1y}(k+1) &= x_{1y}(k) + T x_{2y}(k) \\ x_{2y}(k+1) &= \frac{T}{m} u(k) \cos \theta(k) \sin x_{3\varphi}(k) + x_{2y}(k) \\ x_{3\varphi}(k+1) &= T x_{4\varphi}(k) + x_{3\varphi}(k) \\ x_{4\varphi}(k+1) &= T \tau_\varphi(k) + x_{4\varphi}(k). \end{aligned}$$

De acuerdo a la definición de  $u(k)$  (2.4), se tiene:

$$x_{2y}(k+1) = T(u_1^*(k) + g) \tan x_{3\varphi}(k) + x_{2y}(k).$$

Según la teoría del control óptimo obtenido de [20],  $u_1^*(k)$  tiende a cero cuando  $k$  tiende al infinito. Entonces consideramos que  $\exists n \in \mathbb{Z}^+$  tal que para todo  $k \geq nT$ ,  $|u_1^*(k)|$  está

acotado, por lo que llegamos a:

$$x_{2y}(k+1) = gT \tan x_{3\varphi}(k) + x_{2y}(k).$$

Deseamos hallar un control  $\tau_\varphi^*(k)$ , tal que  $x_{y,\varphi}(k) = \begin{bmatrix} x_{1y} & x_{2y} & x_{3\varphi} & x_{4\varphi} \end{bmatrix}^T$  se vaya a cero y el índice de desempeño

$$J_{y,\varphi} = \sum_{k=1}^{\infty} (x_{y,\varphi}^T(k) Q_{y,\varphi} x_{y,\varphi}(k) + \tau_\varphi^{*2}(k) R_\varphi)$$

se minimice. Si existe un control óptimo  $\tau_\varphi^*(k)$  el cual hace que  $\tan x_{3\varphi}(k) \rightarrow x_{3\varphi}(k)$  podemos diseñar una ley de control óptima  $\tau_\varphi^*(k)$  para el sistema aproximado:

$$x_{y,\varphi}(k+1) = A_{y,\varphi} x_{y,\varphi}(k) + B_{y,\varphi} \tau_\varphi^*(k),$$

donde

$$A_{y,\varphi} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & gT & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & T \end{bmatrix}, \quad B_{y,\varphi} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ T \end{bmatrix}, \quad (2.12)$$

es posible verificar que el par  $(A_{y,\varphi}, B_{y,\varphi})$  es controlable, entonces el control óptimo  $\tau_\varphi^*(k)$  es

$$\tau_\varphi^*(k) = -H_{y,\varphi}^{-1} B_{y,\varphi}^T P_{y,\varphi} A_{y,\varphi} x_{y,\varphi}^*(k), \quad \forall k \geq 0,$$

donde  $H_{y,\varphi} = R_\varphi + B_{y,\varphi}^T P_{y,\varphi} B_{y,\varphi}$  y  $P_{y,\varphi}$  es solución única a la *DARE* expresada como sigue:

$$P_{y,\varphi} = A_{y,\varphi}^T P_{y,\varphi} A_{y,\varphi} - A_{y,\varphi}^T P_{y,\varphi} B_{y,\varphi} H_{y,\varphi}^{-1} B_{y,\varphi}^T P_{y,\varphi} + Q_{y,\varphi}.$$

### 2.3.4 Control óptimo del subsistema $x - \theta$

Consideramos el subsistema  $x - \theta$  definido de la manera siguiente como:

$$\begin{aligned} m\ddot{x} &= -u \sin \theta \\ \ddot{\theta} &= \tau_\theta, \end{aligned}$$

cuya representación en espacio estado se obtiene a partir de las siguientes definiciones ( $x_{1x} = x$ ,  $x_{2x} = \dot{x}$ ,  $x_{3\theta} = \theta$ ,  $x_{4\theta} = \dot{\theta}$ ):

$$\begin{aligned} \dot{x}_{1x} &= x_{2x} \\ \dot{x}_{2x} &= -\frac{1}{m}u \sin x_{3\theta} \\ \dot{x}_{3\theta} &= x_{4\theta} \\ \dot{x}_{4\theta} &= \tau_\theta, \end{aligned}$$

y su representación en el dominio discreto es:

$$\begin{aligned} x_{1x}(k+1) &= x_{1x}(k) + Tx_{2x}(k) \\ x_{2x}(k+1) &= \frac{T}{m}u(k) \sin x_{3\theta}(k) + x_{2x}(k) \\ x_{3\theta}(k+1) &= Tx_{4\theta}(k) + x_{3\theta}(k) \\ x_{4\theta}(k+1) &= T\tau_\theta(k) + x_{4\theta}(k). \end{aligned}$$

Se sustituye la ley de control (2.4) y se llega a:

$$x_{x,\theta}(k+1) = A_{x,\theta}x_{x,\theta}(k) + B_{x,\theta}\tau_\theta^*(k),$$

donde  $x_{x,\theta}(k) = [x_{1x}(k) \ x_{2x}(k) \ x_{3\theta}(k) \ x_{4\theta}(k)]^T$ , y las matrices  $A_{x,\theta}$  y  $B_{x,\theta}$  son similares a los dados en (2.12). Entonces la ley de control óptima  $\tau_\theta^*(k)$  está dada por:



$$\tau_\theta^*(k) = -H_{x,\theta}^{-1} B_{x,\theta}^T P_{x,\theta} A_{x,\theta} x_{x,\theta}^*(k), \quad \forall k \geq 0,$$

donde  $H_{x,\theta} = R_\theta + B_{x,\theta}^T P_{x,\theta} B_{x,\theta}$  y  $P_{x,\theta}$  es única solución a la Ecuación Algebraica Discreta de Riccati. Este enfoque de control se publicó en la revista Journal of Intelligent and Robotic Systems [29].

## 2.4 Control Óptimo de horizonte finito en tiempo discreto

En esta sección se estabilizará el cuadricóptero utilizando un control óptimo similar al anterior pero de horizonte finito. La síntesis de este control se explica a detalle en la tesis de maestría [14]. Se utilizará el mismo modelo reducido 2.1 dividido también por subsistemas utilizando de igual manera el control por linealización exacta.

**Definición 2.4.1.** Sea el sistema definido por  $x(k+1) = f(x(k), u(k))$ ,  $x(k), f(\cdot, \cdot) \in R^n$ ,  $u(k) \in R^m$ , el par  $(x_0, x_1)$  es controlable si existe un control admisible  $u(k)$  tal que transfiere el sistema de  $x_0$  a  $x_1$  en  $N$  número de pasos finito.

### 2.4.1 Subsistema z

Primero se estabilizará el subsistema  $z$ , aplicando una linealización exacta a (2.1). Definimos el siguiente subsistema, como en [27]:

$$m\ddot{z} = u \cos \theta \cos \varphi - mg.$$

Para la representación en espacio estado definimos el vector  $(x_z = \begin{bmatrix} x_{1,z} & x_{2,z} \end{bmatrix}^T)$ :

$$\begin{aligned}\dot{x}_{1,z} &= x_{2,z} \\ \dot{x}_{2,z} &= -g + \left( \frac{\cos \theta \cos \varphi}{m} \right) u.\end{aligned}$$

Utilizando la aproximación de Euler obtenemos la forma discretizada:

$$\begin{aligned}x_{1,z}(k+1) &= x_{1,z}(k) + Tx_{2,z}(k) \\ x_{2,z}(k+1) &= x_{2,z}(k) - Tg + T \left( \frac{\cos \theta(k) \cos \varphi(k)}{m} \right) u(k),\end{aligned}\tag{2.13}$$

donde  $T$  es el tiempo de muestreo. Sabemos que el subsistema discretizado (2.13) puede ser linealizada aplicando la siguiente ley de control:

$$u(k) = m(u_1(k) + g) (\cos \theta(k) \cos \varphi(k))^{-1},\tag{2.14}$$

donde  $\cos \theta(k) \cos \varphi(k) \neq 0$ , si  $\theta, \varphi \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$  el cual es una suposición razonable para las condiciones de operación del vehículo en este proyecto.

Ahora, sustituyendo el control (2.14) en el subsistema (2.13) obtenemos:

$$x_z(k+1) = A_z x_z(k) + B_z u_1(k),\tag{2.15}$$

$$\text{donde } A_z = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, B_z = \begin{bmatrix} 0 \\ T \end{bmatrix}.$$

Es posible verificar que el par  $(A_z, B_z)$  es controlable en un número finito de pasos. Por lo tanto podemos encontrar una ley de control en lazo cerrado con el control linealizante (2.14), tal que el índice de desempeño:

$$J_z = \frac{1}{2} \sum_{k=0}^{N_z-1} (x_z^T(k) Q_z x_z(k) + u_1^2(k) R_z) + \frac{1}{2} x_z^T(N) H_z x_z(N)\tag{2.16}$$

es minimizado, donde  $H_z, Q_z \geq 0$ ,  $R_z > 0$  están dadas y penalizan la convergencia del estado y el consumo de energía respectivamente,  $N_z T = t f_1$ . Así, nosotros deseamos hallar una ley de control  $u_1(k)$ , el cual minimice  $J_z$  sujeto a (2.15). Este problema es conocido, la solución está dada en [20]:

$$u_1^*(k) = -F_z(N-k)x_z^*(k), \quad \forall k = 0, \dots, N-1, \quad (2.17)$$

donde

$$\begin{aligned} F_z(N-k) &= [R_z + B_z^T P_z(k-1)B_z]^{-1} B_z^T P_z(k-1)A_z, \\ P_z(k) &= [A_z + B_z F_z(N-k)]^T P_z(k-1) [A_z + B_z F_z(N-k)] \\ &\quad + F_z^T(N-k)R F_z(N-k) + Q_z, \text{ con } P_z(0) = H_z. \end{aligned}$$

Este control genera la trayectoria óptima del sistema (2.5) el cual minimiza el índice de desempeño (2.16).

### 2.4.2 Subsistema $\psi$

El siguiente subsistema se refiere al ángulo de yaw:

$$\ddot{\psi} = \tau_\psi,$$

cuya representación de espacio estado es

$$\begin{aligned} \dot{x}_{1,\psi} &= x_{2,\psi} \\ \dot{x}_{2,\psi} &= \tau_\psi, \end{aligned}$$

el cual puede ser representado en el dominio discreto como:

$$x_\psi(k+1) = A_\psi x_\psi(k) + B_\psi \tau_\psi(k),$$

donde  $A_\psi = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$ ,  $B_\psi = \begin{bmatrix} 0 \\ T \end{bmatrix}$ .

Definimos el siguiente índice de desempeño

$$J_\psi = \frac{1}{2} \sum_{k=0}^{N_\psi-1} (x_\psi^T(k) Q_\psi x_\psi(k) + \tau_\psi^2(k) R_\psi) + \frac{1}{2} x_\psi^T(N_\psi) H_\psi x_\psi(N_\psi), \quad (2.18)$$

donde  $H_\psi, Q_\psi \geq 0$  y  $R_\psi > 0$ ,  $N_\psi T = tf_2$  entonces el control óptimo está dado por la ecuación siguiente:

$$\tau_\psi^*(k) = -F_\psi(N_\psi - k) x_\psi^*(k), \forall k = 0, \dots, N_\psi - 1,$$

donde

$$\begin{aligned} F_\psi(N_\psi - k) &= [R_\psi + B_\psi^T P_\psi(k-1) B_\psi]^{-1} B_\psi^T P_\psi(k-1) A_\psi, \\ P_\psi(k) &= [A_\psi + B_\psi F_\psi(N_\psi - k)]^T P_\psi(k-1) [A_\psi + B_\psi F_\psi(N_\psi - k)] \\ &\quad + F_\psi^T(N_\psi - k) R_\psi F_\psi(N_\psi - k) + Q_\psi, \text{ con } P_\psi(0) = H_\psi. \end{aligned}$$

Es una ley de control variante en el tiempo que garantiza que el índice de desempeño (2.18) es minimizada.

### 2.4.3 Subsistema $y - \varphi$

El sistema  $y - \varphi$  se representa como sigue:

$$\begin{aligned} m\ddot{y} &= u \cos \theta \sin \varphi \\ \ddot{\varphi} &= \tau_\varphi. \end{aligned}$$

Con su representación en espacio estado ( $x_{1y} = y$ ,  $x_{2y} = \dot{y}$ ,  $x_{3\varphi} = \varphi$ ,  $x_{4\varphi} = \dot{\varphi}$ ):

$$\begin{aligned} \dot{x}_{1y} &= x_{2y} \\ \dot{x}_{2y} &= \frac{1}{m} u \cos \theta \sin x_{3\varphi} \\ \dot{x}_{3\varphi} &= x_{4\varphi} \\ \dot{x}_{4\varphi} &= \tau_\varphi, \end{aligned}$$

al igual que los subsistemas anteriores hallamos su representación en tiempo discreto,

$$\begin{aligned}
x_{1y}(k+1) &= x_{1y}(k) + Tx_{2y}(k) \\
x_{2y}(k+1) &= x_{2y}(k) + \frac{T}{m}u(k) \cos \theta(k) \sin x_{3\varphi}(k) \\
x_{3\varphi}(k+1) &= x_{3\varphi}(k) + Tx_{4\varphi}(k) \\
x_{4\varphi}(k+1) &= T\tau_\varphi(k) + x_{4\varphi}(k).
\end{aligned}$$

De acuerdo a la ley de control linealizante  $u(k)$  (2.14) se tiene para el segundo estado.

$$x_{2y}(k+1) = T(u_1^*(k) + g) \tan x_{3\varphi}(k) + x_{2y}(k),$$

cuando la ley de control óptima,  $u_1^*(k)$  es cero,  $k$  es igual al horizonte. Entonces, consideramos que  $\exists n \in \mathbb{Z}^+$  tal que, para todo  $k \geq nT$ ,  $|u_1^*(k)|$  está acotado, por lo tanto llegamos a:

$$x_{2y}(k+1) = gT \tan x_{3\varphi}(k) + x_{2y}(k).$$

Deseamos hallar el control  $\tau_\varphi^*(k)$ , tal que  $x_{y,\varphi}(k) = \begin{bmatrix} x_{1y} & x_{2y} & x_{3\varphi} & x_{4\varphi} \end{bmatrix}^T$  converja muy rápido,

$$J_{y,\varphi} = \frac{1}{2} \sum_{k=0}^{N_\varphi-1} (x_{y,\varphi}^T(k) Q_{y,\varphi} x_{y,\varphi}(k) + \tau_\varphi^2(k) R_\varphi) + \frac{1}{2} x_{y,\varphi}^T(N_\varphi) H_\varphi x_\varphi(N_\varphi), \quad (2.19)$$

donde  $N_\varphi T = tf_3$  y el índice de desempeño es minimizado. Si existe un control óptimo  $\tau_\varphi^*(k)$  el cual realiza esta tarea, entonces  $\tan x_{3\varphi}(k) \approx x_{3\varphi}(k)$  y podemos así diseñar un control óptimo  $\tau_\varphi^*(k)$  para el sistema aproximado:

$$x_{y,\varphi}(k+1) = A_{y,\varphi} x_{y,\varphi}(k) + B_{y,\varphi} \tau_\varphi(k),$$

donde

$$A_{y,\varphi} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & gT & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & T \end{bmatrix}, \quad B_{y,\varphi} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ T \end{bmatrix}. \quad (2.20)$$

Se puede corroborar que el par  $A_{y,\varphi}, B_{y,\varphi}$  es controlable en un número de pasos finito. Así, de acuerdo a la teoría del control óptimo, la ley de control  $\tau_\varphi^*(k)$  es:

$$\tau_\varphi^*(k) = -F_\varphi(N_\varphi - k) x_{y,\varphi}^*(k), \forall k = 0, \dots, N_\varphi - 1,$$

y

$$\begin{aligned} F_\varphi(N_\varphi - k) &= [R_\varphi + B_{y,\varphi}^T P_\varphi(k-1) B_{y,\varphi}]^{-1} B_{y,\varphi}^T P_\varphi(k-1) A_{y,\varphi}, \\ P_\varphi(k) &= [A_{y,\varphi} + B_{y,\varphi} F_\varphi(N_\varphi - k)]^T P_\varphi(k-1) [A_{y,\varphi} + B_{y,\varphi} F_\varphi(N_\varphi - k)] \\ &\quad + F_\varphi^T(N_\varphi - k) R_\varphi F_\varphi(N_\varphi - k) + Q_\varphi, \text{ con } P_\varphi(0) = H_\varphi. \end{aligned}$$

#### 2.4.4 Subsistema $x - \theta$

Para el subsistema que involucra el eje  $x$  y el ángulo *pitch* se obtiene un control de manera similar al subsistema anteriormente estudiado, para ello definimos:

$$\begin{aligned} m\ddot{x} &= -u \sin \theta \\ \ddot{\theta} &= \tau_\theta, \end{aligned}$$

cuya representación de estado, asignando la siguientes variables ( $x_{1x} = x, x_{2x} = \dot{x}, x_{3\theta} = \theta, x_{4\theta} = \dot{\theta}$ ) es:

$$\begin{aligned} \dot{x}_{1x} &= x_{2x} \\ \dot{x}_{2x} &= -\frac{1}{m} u \sin x_{3\theta} \\ \dot{x}_{3\theta} &= x_{4\theta} \\ \dot{x}_{4\theta} &= \tau_\theta. \end{aligned}$$

De modo similar que los subsistemas anteriores obtenemos su forma discretizada:

$$\begin{aligned}
x_{1x}(k+1) &= x_{1x}(k) + Tx_{2x}(k), \\
x_{2x}(k+1) &= -\frac{T}{m}u(k) \sin x_{3\theta}(k) + x_{2x}(k), \\
x_{3\theta}(k+1) &= Tx_{4\theta}(k) + x_{3\theta}(k), \\
x_{4\theta}(k+1) &= T\tau_{\theta}(k) + x_{4\theta}(k),
\end{aligned}$$

y utilizando los argumentos anteriormente aplicados llegamos a la siguiente aproximación discreta lineal:

$$x_{x,\theta}(k+1) = A_{x,\theta}x_{x,\theta}(k) + B_{x,\theta}\tau_{\theta}^*(k),$$

donde  $x_{x,\theta}(k) = [x_{1x}(k) \ x_{2x}(k) \ x_{3\theta}(k) \ x_{4\theta}(k)]^T$ , y la matriz  $A_{x,\theta}$  y  $B_{x,\theta}$  son similares y están dadas en (2.20). Entonces la ley de control óptima  $\tau_{\theta}^*(k)$  que minimiza un criterio cuadrático similar como en (2.19) es:

$$\tau_{\theta}^*(k) = -F_{\theta}(N_{\theta} - k)x_{x,\theta}^*(k), \forall k = 0, \dots, N_{\theta} - 1,$$

$$\begin{aligned}
F_{\theta}(N_{\theta} - k) &= [R_{\theta} + B_{x,\theta}^T P_{\theta}(k-1)B_{x,\theta}]^{-1} B_{x,\theta}^T P_{\theta}(k-1)A_{x,\theta}, \\
P_{\theta}(k) &= [A_{x,\theta} + B_{x,\theta}F_{\theta}(N_{\theta} - k)]^T P_{\theta}(k-1) [A_{x,\theta} + B_{x,\theta}F_{\theta}(N_{\theta} - k)] \\
&\quad + F_{\theta}^T(N_{\theta} - k)R_{\theta}F_{\theta}(N_{\theta} - k) + Q_{\theta}, \text{ con } P_{\theta}(0) = H_{\theta}.
\end{aligned}$$

Es importante denotar que para hallar estos controles hemos utilizado una linealización exacta, asumiendo ángulos pequeños. Este enfoque de control se publicó en un congreso de Sistemas Autónomos Aéreos (UAS) [13].

## 2.5 Control subóptimo no lineal para sistemas afines en tiempo discreto de horizonte finito

Considere el siguiente sistema no lineal discreto en su forma afín

$$\bar{x}(k+1) = f_0(\bar{x}(k)) + f_1(\bar{x}(k))u(k), \quad (2.21)$$

donde  $k = 0, 1, \dots, N$ ,  $\bar{x}(k), f_0(\bar{x}(k)) \in R^n$ ,  $f_1(\bar{x}(k)) \in R^{n \times m}$  y  $u(k) \in R^m$ , siendo  $T$  el tiempo de muestreo. Se define el siguiente índice de desempeño:

$$J = \frac{1}{2}\bar{x}^T(N)H\bar{x}(N) + \frac{1}{2}\sum_{k=0}^{N-1}\{\bar{x}^T(k)Q\bar{x}(k) + u^T(k)Ru(k)\}, \quad (2.22)$$

donde  $t_f = TN$  es el horizonte y  $Q \geq 0$  y  $R > 0$  son matrices con dimensiones acordes al sistema. Se asume que el par de puntos  $(x_0, x_{t_f})$  del sistema (2.21) satisface la definición 2.4.1. Siguiendo el enfoque de programación dinámica en tiempo discreto tomando de [20], se define lo siguiente,

$$J_{N,N}^* = \frac{1}{2}\bar{x}^T(N)H\bar{x}(N).$$

Observe que este término no depende de la ley de control  $u(N)$ , por lo que puede ser considerado como el valor óptimo de  $J$  en el tiempo discreto  $N$ . Por lo que el siguiente paso se define de la siguiente manera:

$$\begin{aligned} J_{N-1,N}^* &= \min_{u(N-1)} \left\{ \frac{1}{2}\bar{x}^T(N)H\bar{x}(N) + \frac{1}{2}\bar{x}^T(N-1)Q\bar{x}(N-1) + \frac{1}{2}u^T(N-1)Ru(N-1) \right\} \\ &= \min_{u(N-1)} \left\{ J_{N,N}^* + \frac{1}{2}\bar{x}^T(N-1)Q\bar{x}(N-1) + \frac{1}{2}u^T(N-1)Ru(N-1) \right\}. \end{aligned}$$

El valor del estado  $\bar{x}(N)$  se puede calcular por la ecuación dada en 2.21, entonces:

$$\begin{aligned} J_{N-1,N}^*(\bar{x}(N-1), u(N-1)) &= \min_{u(N-1)} \left\{ \frac{1}{2}[f_0(\bar{x}(N-1)) + f_1(\bar{x}(N-1))u(N-1)]^T H \dots \right. \\ &\quad [f_0(\bar{x}(N-1)) + f_1(\bar{x}(N-1))u(N-1)] \\ &\quad \left. + \frac{1}{2}\bar{x}^T(N-1)Q\bar{x}(N-1) + \frac{1}{2}u^T(N-1)Ru(N-1) \right\}, \quad (2.23) \end{aligned}$$

en este paso se resuelve de la manera conocida es decir se busca el mínimo valor de  $J_{N-1,N}$ . Entonces:



$$u^*(N-1) = - [f_1^T(\bar{x}(N-1)) H f_1(\bar{x}(N-1)) + R]^{-1} f_1^T(\bar{x}(N-1)) H f_0(\bar{x}(N-1)), \quad (2.24)$$

ya que  $R$  es una matriz estrictamente definida positiva, garantiza que

$$[f_1^T(\bar{x}(N-1)) H f_1(\bar{x}(N-1)) + R]^{-1}$$

exista siempre. Entonces el control dado por 2.28 es el control óptimo debido a

$$\frac{\partial^2 J_{N-1,N}(\bar{x}(N-1), u(N-1))}{\partial^2 u(N-1)} = R > 0. \quad (2.25)$$

La existencia de un mínimo está garantizado porque el lado derecho de la ecuación 2.23, es fuertemente convexo respecto a  $u(N-1)$ . Para el paso  $N-2$ , observe que el valor dado por  $u^*(N-1)$  es el valor óptimo en ese paso, es decir en  $N-1$ , y de acuerdo con el principio de optimalidad de R. Bellman, este control genera el valor óptimo para  $\bar{x}(N-1)$ . Entonces para este paso tenemos que:

$$\begin{aligned} \tilde{J}_{N-2,N}(\bar{x}(N-2), u(N-1), u(N-2)) &= \min_{u(N-1), u(N-2)} \left\{ \frac{1}{2} \bar{x}(N)^T H \bar{x}(N) + \frac{1}{2} \bar{x}^T(N-1) Q \bar{x}(N-1) \right. \\ &\quad + \frac{1}{2} \bar{x}^T(N-2) Q \bar{x}(N-2) + \frac{1}{2} u^T(N-1) R u(N-1) \\ &\quad \left. + \frac{1}{2} u^T(N-2) R u(N-2) \right\}. \end{aligned} \quad (2.26)$$

Observe que, el término el cual involucra  $\bar{x}(N)$ , depende del control  $u(N-1)$  y el término con  $\bar{x}(N-2)$  depende del control  $u(N-3)$ , únicamente el término con  $\bar{x}(N-1)$  depende del control  $u(N-2)$ . Sin embargo, el control  $u(N-1)$  encontrado en el paso anterior es óptimo y  $\bar{x}(N-1)$  está dado por la ecuación de estado 2.21, lo cual permite que

$$\begin{aligned} \tilde{J}_{N-2,N}(\bar{x}(N-2), u(N-2)) &= \min_{u(N-2)} \left\{ \frac{1}{2} \bar{x}(N)^T H \bar{x}(N) + \frac{1}{2} [f_0(\bar{x}(N-2)) + f_1(\bar{x}(N-2)) u(N-2)]^T Q \dots \right. \\ &\quad [f_0(\bar{x}(N-2)) + f_1(\bar{x}(N-2)) u(N-2)] \\ &\quad + \frac{1}{2} \bar{x}^T(N-2) Q \bar{x}(N-2) + \frac{1}{2} u^T(N-1) R u(N-1) \\ &\quad \left. + \frac{1}{2} u^T(N-2) R u(N-2) \right\}. \end{aligned} \quad (2.27)$$

Después se utiliza esta ecuación para obtener el control  $u(N-2)$  en un sentido subóptimo, con esto se evita la obtención de las ecuaciones de Riccati, el cual es un problema complejo para el sistema propuesto. Sin embargo, observe que la ecuación dada por 2.27 es fuertemente convexa con respecto de  $u(N-2)$  lo cual garantiza la existencia de un mínimo, sin embargo como se dijo anteriormente esto sólo es una aproximación del valor óptimo de  $u(N-2)$ ,

por que como es sabido la señal de control óptimo para  $u(N-2)$  debe hallarse aplicando la solución de la ecuación de Riccati para el caso no lineal, el cual no es fácil de encontrarse. Se procede de la misma manera para hallar el control subóptimo  $u(N-2)$ :

$$\tilde{u}(N-2) = - [f_1^T(\bar{x}(N-2)) Q f_1(\bar{x}(N-1)) + R]^{-1} f_1^T(\bar{x}(N-1)) Q f_0(\bar{x}(N-1)), \quad (2.28)$$

se puede hallar fácilmente las ecuaciones generales de la siguiente manera,

$$\begin{aligned} \tilde{u}(N-k) &= - [f_1^T(\bar{x}(N-k)) Q f_1(\bar{x}(N-k)) + R]^{-1} f_1^T(\bar{x}(N-k)) Q f_0(\bar{x}(N-k)), \\ \tilde{J}_{N-k,N}(\bar{x}(N-k), u(N-k)) &= \tilde{J}_{N-k+1,N} + \frac{1}{2} \{ \bar{x}^T(N-k) Q \bar{x}(N-k) + \tilde{u}^T(N-k) R \tilde{u}(N-k) \}, \\ \text{para todo } k &= 2, \dots, N. \end{aligned}$$

Esta secuencia de control subóptima garantiza que el índice de desempeño 2.22 alcanza un valor mínimo de manera aproximada, véase la ecuación 2.25.

### 2.5.1 Control subóptimo discreto no lineal de horizonte finito aplicado al cuadricóptero

En esta sección se sintetiza un control utilizando una estrategia de optimización en el dominio discreto con un horizonte fijo. Esta técnica se aplica al modelo del cuadricóptero en su forma afín. Este problema se resuelve mediante el uso de algunas suposiciones en un sentido subóptimo ya que la secuencia de control se obtiene hacia atrás paso a paso utilizando una aproximación de la función de Bellman. En este caso a diferencia del caso anterior, las matrices de penalización permiten obtener diferentes estructuras de la señal de control subóptima no lineal.

La síntesis del control se realiza sobre el modelo reducido [8] y está basado en el enfoque de programación dinámica. El modelo reducido se divide en subsistemas como en el enfoque anterior, pero la diferencia es que aquí no se utiliza la linealización exacta ni suposiciones de ángulos pequeños.

**Control de las dinámicas  $z$  y  $\psi$** 

Sea el subsistema que representa la altura y el ángulo de guiñada:

$$\begin{aligned} m\ddot{z}(t) &= u \cos \theta(t) \cos \varphi(t) - mg \\ \ddot{\psi}(t) &= \tau_\psi. \end{aligned}$$

Estas dinámicas no están relacionadas directamente, sin embargo no causa ningún problema el hecho de agruparlas para sintetizar el control. Para hacer la representación en espacio estado utilizamos las siguientes variables. ( $x_{z,\psi} = \begin{bmatrix} x_{1,z} & x_{2,z} & x_{1,\psi} & x_{2,\psi} \end{bmatrix}^T$ ,  $U = \begin{bmatrix} u & \tau_\psi \end{bmatrix}^T$ ,  $x_{1,z} = z$ ,  $x_{2,z} = \dot{z}$ ,  $x_{1,\psi} = \psi$ ,  $x_{2,\psi} = \dot{\psi}$ ):

$$\dot{x}_{z,\psi} = f_{0,z,\psi}(x_{z,\psi}) + f_{1,z,\psi}(\theta, \varphi) U,$$

donde

$$f_{0,z,\psi}(x_{z,\psi}) = \begin{bmatrix} x_{2,z} \\ -mg \\ x_{4,\psi} \\ 0 \end{bmatrix}, f_{1,z,\psi}(\theta, \varphi) = \begin{bmatrix} 0 & 0 \\ \cos \theta \cos \varphi & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

Se debe primero discretizar el sistema, para ello se utiliza la aproximación de Euler, y se obtiene la siguiente expresión:

$$x_{z,\psi}(k+1) = f_{0,z,\psi}(x_{z,\psi}(k)) + f_{1,z,\psi}(\theta(k), \varphi(k)) U(k), \quad (2.29)$$

donde

$$f_{0,z,\psi}(x_{z,\psi}(k)) = \begin{bmatrix} x_{2,z}(k)t_s + x_{1,z}(k) \\ x_{2,z}(k) - gt_s \\ t_s x_{4,\psi}(k) + x_{3,\psi}(k) \\ x_{4,\psi}(k) \end{bmatrix}, f_{1,z,\psi}(k) = \begin{bmatrix} 0 & 0 \\ \frac{t_s}{m} \cos(\theta(k)) \cos(\varphi(k)) & 0 \\ 0 & 0 \\ 0 & t_s \end{bmatrix},$$

donde  $t_s$  representa el tiempo de muestreo. Se requiere encontrar el vector de control  $U(k)$  de tal manera que el índice de desempeño:

$$J_{z,\psi} = \frac{1}{2} x_{z,\psi}^T(N_1) H_0 x_{z,\psi}(N_1) + \frac{1}{2} \sum_{k=0}^{N_1-1} \{ x_{z,\psi}^T(k) Q_{z,\psi} x_{z,\psi}(k) + U^T(k) R_{z,\psi} U(k) \}, \quad (2.30)$$

$$H_0, Q_{z,\psi} \in R^{4 \times 4}, R_{z,\psi} \in R^{2 \times 2}, H_0, Q_{z,\psi} \geq 0, R_{z,\psi} > 0, \quad (2.31)$$

se minimice sujeto a (2.29),  $t_f = N_1 t_s$ , para algún entero  $N_1$ . Utilizando estas consideraciones se puede obtener el siguiente control subóptimo  $\tilde{U}$  de la siguiente manera:

$$\tilde{U}(N_1 - k) = \begin{cases} - \left[ f_{1,z\psi}^T(N_1 - k) H_0 f_{1,z\psi}(N_1 - k) + R_{z,\psi} \right]^{-1} f_{1,z\psi}^T(N_1 - k) H_0 f_{0,z\psi}(\tilde{x}_{z,\psi}(k)), & \text{para } k = 1, \\ - \left[ f_{1,z\psi}^T(N_1 - k) Q_{z,\psi} f_{1,z\psi}(N_1 - k) + R_{z,\psi} \right]^{-1} f_{1,z\psi}^T(N_1 - k) Q_{z,\psi} f_{0,z\psi}(\tilde{x}_{z,\psi}(k)), & \text{para todo } k > 1. \end{cases} \quad (2.32)$$

El control 2.32 asume que los ángulos  $\theta(k)$  y  $\varphi(k)$  están controlados en el intervalo  $[t_0, t_f]$ . Con estas suposiciones el vector  $f_{1,z\psi}^T(N_1 - k)$  está acotado, esto permite concluir que  $U^*(N_1 - k)$  es una ley de control admisible. Sustituyendo los vectores  $f_{1,z\psi}^T(N_1 - k)$ ,  $f_{0,z\psi}(\tilde{x}_{z,\psi}(k))$  y las matrices  $Q_{z,\psi}$  y  $R_{z,\psi}$ , se puede hallar el control en un sentido subóptimo  $\tilde{U}(N_1 - k)$ . Las matrices  $H_0$  y  $Q_{z,\psi}$  están definidas de la siguiente manera:

$$H_0 = \begin{bmatrix} h_{0,11} & h_{0,12} & 0 & 0 \\ h_{0,21} & h_{0,22} & 0 & 0 \\ 0 & 0 & h_{0,33} & h_{0,34} \\ 0 & 0 & h_{0,43} & h_{0,44} \end{bmatrix}, \quad Q_{z,\psi} = \begin{bmatrix} q_{z\psi,11} & q_{z\psi,12} & 0 & 0 \\ q_{z\psi,21} & q_{z\psi,22} & 0 & 0 \\ 0 & 0 & q_{z\psi,33} & q_{z\psi,34} \\ 0 & 0 & q_{z\psi,43} & q_{z\psi,44} \end{bmatrix},$$

las cuales son matrices semidefinidas positivas. Los controles óptimos  $\tilde{u}(N_1 - k)$  y  $\tilde{\tau}_\psi(N_1 - k)$  dados por (2.32) son:

$$\tilde{u}(N_1 - k) = -F_1(N_1 - k) \begin{bmatrix} x_{1,z}(N_1 - k) \\ x_{2,z}(N_1 - k) \end{bmatrix} + F_2(N_1 - k),$$

$$\tilde{\tau}_\psi(N_1 - k) = -G_1 \begin{bmatrix} x_{3,\psi}(N_1 - k) \\ x_{4,\psi}(N_1 - k) \end{bmatrix},$$

donde

$$\begin{aligned} F_1(N_1 - k) &= \left[ \frac{mt_s \cos(\theta(N_1 - k)) \cos(\varphi(N_1 - k))}{r_{11}m^2 + E_{22}t_s^2 \cos^2 \theta(N_1 - k) \cos^2 \varphi(N_1 - k)} \right] \begin{bmatrix} E_{21} & (E_{21}t_s + E_{22}) \end{bmatrix}, \\ F_2(N_1 - k) &= \left[ \frac{mt_s \cos(\theta(N_1 - k)) \cos(\varphi(N_1 - k))}{r_{11}m^2 + E_{22}t_s^2 \cos^2 \theta(N_1 - k) \cos^2 \varphi(N_1 - k)} \right] E_{22}gt_s, \\ G_1 &= \left( \frac{t_s}{E_{44}t_s^2 + r_{22}} \right) \begin{bmatrix} E_{43} & (E_{43}t_s + E_{44}) \end{bmatrix}. \end{aligned}$$

Y con,

$$E_{ij} = \begin{cases} h_{ij}, & \text{para } k = 1, \\ q_{ij}, & \text{para } k > 1. \end{cases}$$

Las expresiones de control final son lineales, estos dependen de la elección propia de las matrices  $H_0$  y  $Q_{z,\psi}$ .

### Subsistema $y - \varphi$

Considera el subsistema  $y - \varphi$ , el cual corresponde al movimiento traslacional en el eje  $y$  y el ángulo de alabeo.

$$m\ddot{y} = u \cos \theta \sin \varphi$$

$$\ddot{\varphi} = \tau_\varphi.$$

Su representación en espacio estado puede obtenerse utilizando las siguientes definiciones:

$$(x_{1y} = y, x_{2y} = \dot{y}, x_{1\varphi} = \varphi, x_{2\varphi} = \dot{\varphi}),$$

$$\dot{x}_{1y} = x_{2y}$$

$$\dot{x}_{2y} = u \cos \theta \sin(x_{1,\varphi})$$

$$\dot{x}_{1\varphi} = x_{2,\varphi}$$

$$\dot{x}_{2\varphi} = \tau_\varphi.$$

Al igual que el subsistema anterior se obtiene su representación en discreto:

$$\begin{aligned} x_{1y}(k+1) &= t_s x_{2y}(k) + x_{1y}(k) \\ x_{2y}(k+1) &= t_s u(k) \cos(\theta(k)) \sin(x_{1,\varphi}(k)) + x_{2y}(k) \\ x_{1\varphi}(k+1) &= t_s x_{2\varphi}(k) + x_{1\varphi}(k) \\ x_{2\varphi}(k+1) &= t_s \tau_\varphi(k) + x_{2\varphi}(k). \end{aligned}$$

En este subsistema se sustituye el control  $u(k)$  por  $u^*(k)$  el cual se obtuvo anteriormente y se llega a la siguiente expresión:

$$x_{y,\varphi}(k+1) = f_0(x_{y,\varphi}(k), u^*(k)) + f_1 \tau_\varphi(k), \quad (2.33)$$

donde

$$f_0(x_{y,\varphi}(k), u^*(k)) = \begin{bmatrix} t_s x_{2y}(k) + x_{1y}(k) \\ t_s u^*(k) \cos(\theta(k)) \sin(x_{1,\varphi}(k)) + x_{2y}(k) \\ t_s x_{2\varphi}(k) + x_{1\varphi}(k) \\ x_{2\varphi}(k) \end{bmatrix}, \quad f_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ t_s \end{bmatrix}.$$

Se quiere encontrar el control  $\tau_\varphi$  de tal manera que el índice de desempeño

$$\begin{aligned} J_{y,\varphi} &= \frac{1}{2} x_{y,\varphi}^T(N_2) H_1 x_{y,\varphi}(N) + \frac{1}{2} \sum_{k=0}^{N_2-1} \{ x_{y,\varphi}^T(k) Q_{y,\varphi} x_{y,\varphi} + \tau_\varphi^2(k) R_\varphi \}, \\ Q_{y,\varphi} &\in R^{4 \times 4}, \quad R_\varphi \in R, \quad Q_{y,\varphi} \geq 0, \quad R_\varphi > 0 \end{aligned}$$

se minimice, sujeto a (2.5.1). Entonces se puede llegar a la siguiente expresión:

$$\tau_\varphi^*(N_2 - k) = [f_1^T E_{y,\varphi} f_1 + R_\varphi]^{-1} f_1^T E_{y,\varphi} f_0(x_{y,\varphi}^*(N_2 - k), u^*(N_2 - k)),$$

donde

$$E_{y,\varphi}(i, j) = \begin{cases} h_{1,ij}, & \text{para } k = 1, \\ q_{y\varphi,ij}, & \text{para } k > 1. \end{cases}$$

Se considera que la matriz  $E_{y,\varphi}$  está definida como:

$$E_{y,\varphi} = \begin{bmatrix} e_{11} & 0 & 0 & e_{14} \\ 0 & e_{22} & 0 & 0 \\ 0 & 0 & e_{33} & e_{34} \\ e_{41} & 0 & e_{43} & e_{44} \end{bmatrix},$$

con esto, se tiene que

$$f_1^T Q_{y,\varphi} f_1 + R_\varphi = \begin{bmatrix} 0 \\ 0 \\ 0 \\ t_s \end{bmatrix}^T \begin{bmatrix} e_{11} & 0 & 0 & e_{14} \\ 0 & e_{22} & 0 & 0 \\ 0 & 0 & e_{33} & e_{34} \\ e_{41} & 0 & e_{43} & e_{44} \end{bmatrix} \dots$$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ t_s \end{bmatrix} + R_\varphi = e_{44}t_s^2 + R_\varphi$$

y

$$f_1^T Q_{y,\varphi} f_0(x_{y,\varphi}(N-k), u^*(N-k)) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ t_s \end{bmatrix}^T \begin{bmatrix} e_{11} & 0 & 0 & e_{14} \\ 0 & e_{22} & 0 & 0 \\ 0 & 0 & e_{33} & e_{34} \\ e_{41} & 0 & e_{43} & e_{44} \end{bmatrix} \dots$$

$$\begin{bmatrix} t_s x_{2y}(N-k) + x_{1,y}(N-k) \\ t_s u^* \cos(\theta(N-k)) \sin(\varphi(N-k)) + x_{2y}(N-k) \\ t_s x_{2,\varphi}(N-k) + x_{1,\varphi}(N-k) \\ x_{2,\varphi}(N-k) \end{bmatrix}$$

$$= e_{43}t_s x_{1,\varphi}(N-k) + (e_{43}t_s^2 + \bar{q}_{44}t_s) x_{2,\varphi}(N-k) + \dots$$

$$\dots + e_{41}t_s x_{1,y}(N-k) + e_{41}t_s^2 x_{2,y}(N-k),$$

lo cual permite

$$\tau_\varphi^*(N-k) = -F_\varphi x_{y,\varphi}^*(N-k),$$

donde

$$F_\varphi = \frac{1}{e_{44}t_s^2 + R_\varphi} \begin{bmatrix} e_{41}t_s & e_{41}t_s^2 & e_{43}t_s & e_{43}t_s^2 + e_{44}t_s \end{bmatrix}.$$

La elección de las matrices  $Q_{y,\varphi}$  y  $H$  permiten también obtener una expresión final lineal.

### Subsistema $x - \theta$

Considere el subsistema del eje de traslación  $x$  y el ángulo de cabeceo  $\theta$ :

$$\begin{aligned} m\ddot{x} &= -u \sin \theta \\ \ddot{\theta} &= \tau_\theta, \end{aligned}$$

con las siguientes variables de estado ( $x_{1,x} = x$ ,  $x_{2,x} = \dot{x}$ ,  $x_{1,\theta} = \theta$ )

$$\begin{aligned} \dot{x}_{1,x} &= x_{2,x} \\ \dot{x}_{2,x} &= -u \sin(x_{1,\theta}) \\ \dot{x}_{1,\theta} &= x_{2,\theta} \\ \dot{x}_{2,\theta} &= \tau_\theta. \end{aligned}$$

Se obtiene de igual modo su representación en discreto:

$$\begin{aligned} x_{1,x}(k+1) &= t_s x_{2,x}(k) + x_{1,x}(k) \\ x_{2,x}(k+1) &= -t_s u(k) \sin(x_{1,\theta}(k)) + x_{2,x}(k) \\ x_{1,\theta}(k+1) &= t_s x_{2,\theta}(k) + x_{1,\theta}(k) \\ x_{2,\theta}(k+1) &= t_s \tau_\theta(k) + x_{2,\theta}(k). \end{aligned}$$

También se sustituye  $u(k)$  por  $u^*(k)$  y se obtiene:

$$x_{x,\theta}(k+1) = \bar{f}_0(x_{x,\theta}(k), u^*(k)) + f_1 \tau_\theta(k),$$

donde

$$\bar{f}_0(x_{x,\theta}(k), u^*(k)) = \begin{bmatrix} t_s x_{2,x}(k) + x_{1,x}(k) \\ -t_s u^*(k) \sin(x_{1,\theta}(k)) + x_{2,x}(k) \\ t_s x_{2,\theta}(k) + x_{1,\theta}(k) \\ x_{2,\theta}(k) \end{bmatrix}, \quad f_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ t_s \end{bmatrix}.$$



Se requiere encontrar  $\tau_\theta^*$  tal que se minimice el índice de desempeño siguiente:

$$J_{x,\theta} = \frac{1}{2} x_{x,\theta}^T(N_3) H_2 x_{x,\theta}(N_3) + \frac{1}{2} \sum_{k=0}^{N-1} \{ x_{x,\theta}^T(k) Q_{x,\theta} x_{x,\theta} + \tau_\theta^2(k) R_\theta \},$$

$$Q_{x,\theta} \in R^{4 \times 4}, R_\theta \in R, Q_{x,\theta} \geq 0, R_\theta > 0,$$

por lo que

$$\tau_\theta^*(N_3 - k) = - [ f_1^T Q_{x,\theta} f_1 + R_\theta ]^{-1} f_1^T E_{x,\theta} \bar{f}_0(x_{x,\theta}^*(N_3 - k), u^*(N_3 - k)),$$

se considera la siguiente matriz  $E_{x,\theta}$  como:

$$E_{x,\theta} = \begin{bmatrix} \tilde{e}_{11} & 0 & 0 & \tilde{e}_{14} \\ 0 & \tilde{e}_{22} & 0 & 0 \\ 0 & 0 & \tilde{e}_{33} & \tilde{e}_{43} \\ \tilde{e}_{41} & 0 & \tilde{e}_{43} & \tilde{e}_{44} \end{bmatrix}, E_{y,\varphi}(i, j) = \begin{cases} h_{2,ij}, & \text{para } k = 1, \\ q_{x,\theta,ij}, & \text{con } k > 1. \end{cases}$$

Y se tiene que

$$\begin{aligned} f_1^T Q_{x,\theta} f_1 + R_\theta &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ t_s \end{bmatrix}^T \begin{bmatrix} \tilde{e}_{11} & 0 & 0 & \tilde{e}_{14} \\ 0 & \tilde{e}_{22} & 0 & 0 \\ 0 & 0 & \tilde{e}_{33} & \tilde{e}_{34} \\ \tilde{e}_{41} & 0 & \tilde{e}_{43} & \tilde{e}_{44} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ t_s \end{bmatrix} + R_\theta \\ &= \tilde{e}_{44} t_s^2 + R_\theta, \end{aligned}$$

y

$$\begin{aligned}
f_1^T Q_{x,\theta} \bar{f}_0(x_{x,\theta}(N-k), u^*(N-k)) &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ t_s \end{bmatrix}^T \begin{bmatrix} \tilde{e}_{11} & 0 & 0 & \tilde{e}_{14} \\ 0 & \tilde{e}_{22} & 0 & 0 \\ 0 & 0 & \tilde{e}_{33} & \tilde{e}_{34} \\ \tilde{e}_{41} & 0 & \tilde{e}_{43} & \tilde{e}_{44} \end{bmatrix} \dots \\
&= \begin{bmatrix} t_s x_{2,x}(N-k) + x_{1,x}(N-k) \\ t_s u^* \cos(\theta(N-k)) \sin(x_{1,\varphi}(N-k)) + x_{2,x}(N-k) \\ t_s x_{2,\theta}(N-k) + x_{1,\theta}(N-k) \\ x_{2,\theta}(N-k) \end{bmatrix} \\
&= \tilde{e}_{43} t_s x_{1,\theta}(N-k) + (\tilde{e}_{43} t_s^2 + \tilde{e}_{44} t_s) x_{2,\theta}(N-k) + \dots \\
&\quad \dots + \tilde{e}_{41} t_s x_{1,x}(N-k) + \tilde{e}_{41} t_s^2 x_{2,x}(N-k),
\end{aligned}$$

esto permite obtener:

$$\tau_\theta^*(N-k) = -F_\theta x_{x,\theta}^*(N-k),$$

donde

$$F_\theta = \frac{1}{\tilde{e}_{44} t_s^2 + R_\theta} \begin{bmatrix} \tilde{e}_{41} t_s & \tilde{e}_{41} t_s^2 & \tilde{e}_{43} t_s & \tilde{e}_{43} t_s^2 + \tilde{e}_{44} t_s \end{bmatrix}.$$

De manera similar, las elecciones de las matrices  $Q$  y  $H$  en cada controlador permiten encontrar estructuras de control más simples.

## 2.6 Algoritmo de control super-twisting con Control subóptimo no lineal

En esta sección se aborda la combinación de dos enfoques de control, uno robusto y otro basado en una estrategia de optimización el cual se abordó en la sección anterior, el objetivo es tener un buen desempeño mientras existan perturbaciones externas pero tener un ahorro en el consumo energético. El método propuesto en esta tesis, es hacer una estimación de las perturbaciones externas que afectan al vehículo utilizando el método de mínimos cuadrados y utilizar el criterio de la estimación para decidir que enfoque de control utilizar. El enfoque de

control super-twisting fue dirigido por el Dr. Iván González Hernández profesor-investigador del laboratorio UMI-LAFMIA.

### 2.6.1 Algoritmo super-twisting aplicado al cuadricóptero

En esta subsección se aborda la síntesis de control basado en el algoritmo super-twisting para hacer seguimiento de trayectorias con un aeronave. Este enfoque fue originalmente propuesto en [16]. Con el fin de demostrar la robustez del control propuesto, en el modelo reducido 2.1 se toma en cuenta el término de Coriolis como una incertidumbre en el modelo dinámico, por tanto, se considera la siguiente entrada de control:

$$\tau_\eta = \tilde{\tau}_\eta + 2C(\eta, \dot{\eta}) \dot{\eta} \quad (2.34)$$

para el cual se asume que

$$C(\eta, \dot{\eta}) \dot{\eta} = \gamma_\eta(\eta, \dot{\eta}) \leq M$$

, en donde  $\gamma_\eta(\cdot)$  es el término de incertidumbre que resulta de la fuerza de Coriolis el cual contiene el giroscopio y términos centrífugos. Con  $M > 0$ , el cual es el máximo valor de incertidumbre en una aplicación real,  $\eta = (\varphi, \theta, \psi)^T \in \mathbb{R}^3$ . Además, los valores  $\gamma_\eta(\cdot)$  dependen estrictamente de la velocidad y la posición angular que afecta al vehículo cuando realiza el seguimiento de la trayectoria. Por lo tanto de las ecuaciones del modelo dinámico presentado en [21] se obtiene lo siguiente.

$$\begin{aligned} m\ddot{x} &= -u \sin \theta \\ m\ddot{y} &= u \cos \theta \sin \varphi \\ m\ddot{z} &= u \cos \theta \cos \varphi - mg \end{aligned} \quad (2.35)$$

$$\begin{aligned} I_{xx}\ddot{\varphi} &= \tilde{\tau}_\varphi + \gamma_x(\eta, \dot{\eta}) \\ I_{yy}\ddot{\theta} &= \tilde{\tau}_\theta + \gamma_y(\eta, \dot{\eta}) \\ I_{zz}\ddot{\psi} &= \tilde{\tau}_\psi + \gamma_z(\eta, \dot{\eta}) \end{aligned} \quad (2.36)$$

donde  $I_{xx}$ ,  $I_{yy}$ ,  $I_{zz}$  son los elementos de la diagonal principal de la matriz de inercia del vehículo. Se asume que  $u$  es lo suficientemente pequeña, lo que significa que el vehículo ha alcanzado la altura deseada. Esta es la condición principal para poder iniciar el seguimiento de la trayectoria. Para controlar la dinámica de  $z$ , como en los enfoques anteriores se aplica una linealización exacta:

$$u = \frac{mc_1 + mg}{\cos(\varphi) \cos(\theta)} \quad (2.37)$$

donde  $c_1$  es un controlador Proporcional Derivativo, y se asume que  $c_1 \rightarrow 0$ , las dinámicas  $x$  y  $y$  están aproximadas como:

$$\begin{aligned} \ddot{x} &\approx -g \frac{\tan \theta}{\cos \varphi} \\ \ddot{y} &\approx g \tan \varphi; \end{aligned} \quad (2.38)$$

### Control de trayectoria en $y$

Para poder implementar este control en la dinámica  $y$ , primero se consideran las dinámicas del ángulo de alabeo  $\varphi$  dado en 2.36 y la dinámica de  $y$  dada en 2.38. Se van a implementar dos controles no lineales, el control subóptimo y este control basado en el algoritmo super-twisting para garantizar una mejor convergencia. Para simplificar la síntesis del control, se considerarán las siguientes suposiciones para vuelo estacionario:

$$|\varphi| < \frac{\pi}{6} \quad (2.39)$$

el cual es una suposición razonable para la planta. Si se deriva dos veces la dinámica de  $y$  dado en 2.38, se obtiene:

$$y^{(iv)} = \tilde{\tau}_\varphi \kappa_t + f_1(\varphi, \dot{\varphi}) \quad (2.40)$$

donde  $f_1(\cdot)$  es el término de Coriolis y  $\kappa_t$  es el coeficiente del tensor de inercia el cual es conocido, [16]. El sistema dado en 2.40 representa un modelo con incertidumbres de 4 integradores en cascada.

Si se asumen perturbaciones externas en la entrada de control del sistema 2.40 se obtiene

$$y^{(iv)} = \tilde{\tau}_\varphi \kappa_t + f_1(\varphi, \dot{\varphi}) + \gamma_{ext}, \quad (2.41)$$

donde  $\gamma_{ext}$  es una perturbación externa el cual existe en un problema real y puede afectar directamente al desempeño del vehículo cuando sigue una trayectoria. Además, esta suposición considera que el sistema dinámico funciona en una región lineal ( $|\varphi| < \pi/6$ ) y que las perturbaciones en la entrada del sistema afectan directamente a la estabilidad.

Dada la suposición 2.39, se establecen las siguientes desigualdades:

$$\begin{aligned} |f_1(\varphi, \dot{\varphi})| &\leq \delta_1 \\ |\dot{\gamma}_{ext}| &\leq \delta_2 \end{aligned} \quad (2.42)$$

para algunas constantes  $\delta_1, \delta_2 > 0$ . Por lo que se propone un control basado en el algoritmo super-twisting para asegurar la estabilidad durante el seguimiento de una trayectoria aún en presencia de perturbaciones externas. Dadas las consideraciones anteriores se puede obtener el siguiente conjunto de ecuaciones:

$$\begin{aligned} \dot{y}_1 &= \dot{y} \\ \dot{y}_2 &= \ddot{y} = \varphi \\ \dot{y}_3 &= y^{(3)} = \dot{\varphi} \\ \dot{y}_4 &= y^{(iv)} = \tilde{\tau}_\varphi \kappa_t + f_1(\varphi, \dot{\varphi}) + \gamma_{ext} = \ddot{\varphi}, \end{aligned} \quad (2.43)$$

se definen los siguientes errores de seguimiento

$$\begin{aligned} e_1 &= y - y_d(t) \\ e_2 &= \dot{y} - \dot{y}_d(t) \\ e_3 &= \ddot{y} - \ddot{y}_d(t) = \varphi - \ddot{y}_d(t) \\ e_4 &= \dot{\varphi}, \end{aligned} \quad (2.44)$$

donde  $y_d(t)$  es una ecuación paramétrica la cual describe una circunferencia. El seguimiento de la trayectoria se alcanza con el diseño de una ley de control en lazo cerrado que asegure que  $e_i(t)$  con  $i = 1, 2, 3, 4$  esté acotada y que converja a cero en tiempo finito. Para este análisis se restringe para el caso cuando el sistema:

$$\dot{v} = f_0(v, \zeta) \quad (2.45)$$

con  $v = [e_1, e_2, e_3]^T$ ,  $\dot{v} = [\dot{e}_1, \dot{e}_2, \dot{e}_3]^T$  y  $\zeta = e_4$ . El sistema anterior 2.45 es entrada-acotada estado-acotado estable. Este será el caso para alguna entrada acotada  $\zeta$  y algún estado inicial  $v(0)$  si el sistema  $\dot{v} = f_0(v, \zeta)$  es entrada a estado estable. Por lo tanto podemos concentrarnos en mostrar la convergencia de  $e(t_0)$  y que a su vez esté acotado. Por lo que podemos comenzar con el sistema:

$$\begin{aligned}\dot{e}_1 &= e_2 \\ \dot{e}_2 &= e_3 \\ \dot{e}_3 &= e_4,\end{aligned}\tag{2.46}$$

donde  $e_4$  es visto como la entrada de control. Se requiere diseñar  $e_4$  que estabilice el origen. Para el sistema lineal en su forma canónica controlable, se propone el control lineal,

$$e_4 = -(k_1 e_1 + k_2 e_2 + k_3 e_3),\tag{2.47}$$

tal que

$$e_4 = -[k_1 (y - y_d) + k_2 (\dot{y} - \dot{y}_d) + k_3 (\varphi - \ddot{y}_d)],\tag{2.48}$$

donde el diseño de los coeficientes  $k_1$  a  $k_3$  se eligen de modo que el polinomio característico dominante de la dinámica del error sea Hurwitz, es decir  $p^3 + k_3 p^2 + k_2 p + k_1 = 0$ . Entonces, para diseñar el control por modos deslizantes, se empieza por diseñar la superficie de deslizamiento. Para este problema de seguimiento, la superficie de deslizamiento queda denotado por  $s$ . Note que  $s$  está definido por los errores de seguimiento de la manera siguiente:

$$s = (k_1 e_1 + k_2 e_2 + k_3 e_3) + e_4,\tag{2.49}$$

y

$$\begin{aligned}\dot{s} &= (k_1 \dot{e}_1 + k_2 \dot{e}_2 + k_3 \dot{e}_3) + \dot{e}_4 \\ \dot{s} &= (k_1 \dot{e}_1 + k_2 \dot{e}_2 + k_3 \dot{e}_3) + \ddot{\varphi} \\ \dot{s} &= (k_1 e_2 + k_2 e_3 + k_3 e_4) + (\tilde{\tau}_\varphi \kappa_t + f_1(\varphi, \dot{\varphi}) + \gamma_{ext}).\end{aligned}\tag{2.50}$$

Si la ley de control puede llevar las trayectorias al espacio de fase tal que  $s = 0$  en 2.49, los errores de seguimiento convergen debido a que:

$$s = (k_1 e_1 + k_2 e_2 + k_3 e_3) + e_4 = 0,\tag{2.51}$$

donde la superficie  $s$  es alcanzada en tiempo finito. Note que esta definición se trata de un caso ideal. Por lo tanto, se puede diseñar la entrada de control ( $\tilde{\tau}_\varphi$ ) de la siguiente manera,

$$\tilde{\tau}_\varphi = -\frac{1}{\kappa_t} (k_1 e_2 + k_2 e_3 + k_3 e_4) + \nu, \quad (2.52)$$

con

$$\nu = -g_1 |s|^{1/2} \text{sign}(s) - g_2 \int_0^t \text{sign}(s(\tau)) d\tau, \quad (2.53)$$

donde la variable  $\nu$  representa el componente usual del algoritmo super-twisting, este se elije de la siguiente manera:

$$g_1 = 1.5\sqrt{L} \quad , \quad g_2 = 1.1L$$

y  $L$  es la suma de las perturbaciones que afectan directamente al vehículo. Esto es, ( $L = \delta_1 + \delta_2$ ).

## 2.7 Algoritmo de control Backstepping

En esta sección se desarrolla un algoritmo de control basado en el resultado clásico de Backstepping [17] [19], este enfoque hace uso de la función  $\sin \zeta$  como la entrada virtual. Con la condición de  $\zeta \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$ , se garantiza que sólo hay una raíz que cumple con  $\sin \zeta = 0$  y es cero. Esta suposición es válida para el sistema del cuadricóptero. Note que en el algoritmo clásico la entrada virtual sólo es  $\zeta$ .

Sea el siguiente sistema:

$$\begin{aligned} \dot{\eta}(t) &= f(\eta(t)) + g(\eta(t)) \sin(\zeta(t)) \\ \dot{\zeta}(t) &= u, \end{aligned} \quad (2.54)$$

donde  $\eta \in \mathfrak{R}^n$ ,  $\zeta \in \mathfrak{R}$ ,  $u \in \mathfrak{R}$ . Se asume lo siguiente:

1. Las funciones  $f : D \subseteq \mathfrak{R}^n \longrightarrow \mathfrak{R}^n$  y  $g : D \subseteq \mathfrak{R}^n \longrightarrow \mathfrak{R}^n$  son continuamente diferenciables,

2.  $f(0) = 0$  y  $\eta = 0 \in D$ ,
3.  $\zeta \in (-\frac{\pi}{2}, \frac{\pi}{2})$ .

El problema clásico es encontrar la entrada  $u$  tal que  $\eta = 0$  y  $\zeta = 0$  sean asintóticamente estables, sin embargo, para la estructura dada por la ecuación 2.56, la función  $\sin(\cdot)$  garantiza que esta entrada virtual siempre esté acotada. A diferencia de [2], donde se usó una transformación diferente de variables de estado, en este artículo también se utiliza el algoritmo Backstepping de una manera re-cursiva, pero cuando se usa la estructura dada en (2.56), permite garantizar que la entrada virtual para las ecuaciones de posición traslacional y su velocidad estén siempre acotadas.

Considere la siguiente Proposición:

Sea el sistema dado por la ecuación 2.56, bajo las suposiciones descritas anteriormente, un control  $u$  el cual estabiliza localmente al sistema 2.56, está dado de la siguiente manera:

$$u = \frac{-k(\sin(\zeta) - \varphi(\eta)) - \frac{\partial V(\eta)}{\partial \eta} g(\eta) + \frac{\partial \varphi}{\partial \eta} [f(\eta) + g(\eta) \sin(\zeta)]}{\cos(\zeta)}. \quad (2.55)$$

Donde  $k > 0$ ,  $\varphi(\eta)$  y  $V(\eta) > 0$  son funciones conocidas.

**Prueba:**

La prueba está basada en el algoritmo clásico de Backstepping, se asume que existe una función  $\varphi(\eta)$  tal que  $\varphi(0) = 0$  y el sistema:

$$\dot{\eta} = f(\eta) + g(\eta)\varphi(\eta), \quad (2.56)$$

tiene solución trivial estable. Entonces, se asume que existe una función suave, definida positiva  $V(\eta)$  el cual satisface:

$$\left. \frac{dV(\eta)}{dt} \right|_{(2.56)} = \frac{\partial V}{\partial \eta} [f(\eta) + g(\eta)\varphi(\eta)] \leq -W(\eta), \quad \forall \eta \in D,$$



donde  $W(\eta)$  es una función conocida y definida positiva. La ecuación en espacio estado 2.56 se puede reescribir de la siguiente manera:

$$\begin{aligned}\dot{\eta} &= f(\eta) + g(\eta)\varphi(\eta) + g(\eta)[\sin(\varphi(\eta)) - \varphi(\eta)] \\ \dot{\zeta} &= u.\end{aligned}\tag{2.57}$$

Sea  $z$  una variable de desviación, definida como  $z \triangleq \sin(\zeta) - \varphi(\eta)$  donde su derivada puede ser calculada como  $\dot{z} = \cos(\zeta)\dot{\zeta} - \dot{\varphi} = \cos(\zeta)u - \dot{\varphi}$ . Entonces el sistema 2.57 puede ser expresada como una función de  $z$ :

$$\begin{aligned}\dot{\eta} &= \overbrace{f(\eta) + g(\eta)\varphi(\eta)}^{\bar{f}(\cdot)} + g(\eta)z \\ \dot{z} &= \cos(\zeta)u - \dot{\varphi},\end{aligned}\tag{2.58}$$

como en el backstepping clásico, cuando  $z = 0$ , es posible garantizar que 2.58 tiene punto de equilibrio cero estable.

$$\dot{\varphi}(\eta) = \frac{\partial \varphi}{\partial \eta} \dot{\eta} = \frac{\partial \varphi(\eta)}{\partial \eta} [f(\eta) + g(\eta) \sin(\zeta)].\tag{2.59}$$

Defina  $\nu \triangleq \cos(\zeta)u - \dot{\varphi}$ , con esta definición el sistema puede ser representado de la siguiente manera:

$$\begin{aligned}\dot{\eta} &= \bar{f}(\eta) + g(\eta)z \\ \dot{z} &= \nu,\end{aligned}\tag{2.60}$$

y por otro lado, sea la siguiente función:

$$V_a(\eta, z) = V(\eta) + \frac{1}{2}z^2,\tag{2.61}$$

como  $V_a$  es definida positiva, podría ser una función candidata de Lyapunov para el sistema (2.60), la derivada temporal de esta función es:

$$\left. \frac{dV_a(\eta, z)}{dt} \right|_{(2.60)} = \frac{\partial V(\eta)}{\partial \eta} \bar{f}(\eta) + \frac{\partial V(\eta)}{\partial \eta} g(\eta)z + z\nu \leq -W(\eta) + \frac{\partial V(\eta)}{\partial \eta} g(\eta)z + z\nu.$$

Se elige la variable  $\nu$  como  $\nu \triangleq -kz - \frac{\partial V(\eta)}{\partial \eta} g(\eta)$ , donde  $k > 0$ , y sustituyendo en la ecuación anterior, se obtiene:

$$V_a(\dot{\eta}, z) \Big|_{(2.60)} \leq -W(\eta) - kz^2 < 0. \quad (2.62)$$

Por lo que el sistema dado por 2.60 tiene solución trivial, asintóticamente estable ( $\eta = 0$  y  $z = 0$ ). La entrada  $u$  se puede obtener de (2.58):

$$\begin{aligned} u &= \frac{\dot{z} + \dot{\varphi}}{\cos(\zeta)} = \frac{\nu + \dot{\varphi}}{\cos(\zeta)} \\ &= \frac{-k(\sin(\zeta) - \varphi(\eta)) - \frac{\partial V(\eta)}{\partial \eta} g(\eta) + \frac{\partial \varphi}{\partial \eta} [f(\eta) + g(\eta) \sin(\zeta)]}{\cos(\zeta)}, \quad \cos(\zeta) \neq 0. \end{aligned} \quad (2.63)$$

lo cual concluye la prueba. ■

Esta ley de control estabiliza el sistema dado por 2.56. Siguiendo este algoritmo simple, se puede aplicar para controlar el vuelo de un cuadricóptero en el exterior.

### 2.7.1 Algoritmo de backstepping y modos deslizantes terminales aplicado a un Cuadricóptero

En esta sección, se aplica el algoritmo de Backstepping a la parte de orientación y también al desplazamiento sobre los ejes  $x$  y  $y$ . Para la altura se implementa un control por modos deslizantes terminales, este algoritmo fue presentado en [12]. Como se explicó anteriormente, este sistema puede dividirse en cuatro subsistemas: El subsistema de altura, el subsistema del ángulo de guiñada, el subsistema  $x - \theta$  y el subsistema de  $y - \varphi$ , los cuales se discutirán a detalle en las siguientes subsecciones. Se utilizará el modelo no lineal el cual se tomó de [9].

#### Subsistema $z$

Sea el subsistema  $z$  el cual se representa de la siguiente manera:

$$m\ddot{z} = u \cos(\theta) \cos(\varphi) - mg. \quad (2.64)$$

Consideramos el siguiente cambio de variable:  $\xi_1 = z$  y  $\xi_2 = \dot{z}$ . Al reescribir la ecuación anterior, se obtiene lo siguiente:

$$\begin{aligned}\dot{\xi}_1 &= \xi_2, \\ \dot{\xi}_2 &= \frac{u}{m} \cos(\theta) \cos(\varphi) - g + \delta(t),\end{aligned}\tag{2.65}$$

Para este sistema  $f(\xi, t) = -g + \delta(t)$ , y  $|\delta(t)| \leq \delta_0$  por lo que  $\delta_1 = g + \delta_0$ , donde  $\delta(t)$  representa una perturbación acotada. Se puede definir  $\bar{g}(\xi) = \frac{1}{m} \cos(\theta) \cos(\varphi)$  el cual también esta acotada (Para mayor detalle consulte [12]). De ésta manera podemos proponer la entrada de control como:

$$\begin{aligned}u &= \begin{cases} \bar{u}, & \xi_1 \neq 0 \\ -\frac{1}{\bar{g}(\xi)} \rho \text{Sign}(\xi_2), & \xi_1 = 0 \end{cases} \\ \bar{u} &= -\frac{1}{\bar{g}(\xi)} \left\{ \frac{2}{3} R \xi_2 |\xi_1|^{-\frac{1}{3}} + \rho \text{Sign} \left( \xi_2 + R \xi_1 |\xi_1|^{-\frac{1}{3}} \right) \right\}\end{aligned}\tag{2.66}$$

Bajo la suposición de vuelo estacionario donde  $\theta \approx 0$  y  $\varphi \approx 0$ , la entrada de control para esta dinámica queda de la siguiente manera.

$$u = \begin{cases} -\frac{2}{3} R \xi_2 |\xi_1|^{-\frac{1}{3}} - \rho \text{Sign}(\xi_2 + R \xi_1 |\xi_1|^{-\frac{1}{3}}), & \xi_1 \neq 0, \\ -\rho \text{Sign}(\xi_2), & \xi_1 = 0. \end{cases}$$

### Subsistema $\psi$

El subsistema dos, el cual describe la dinámica del ángulo de guiñada está representado como:

$$\ddot{\psi} = \dot{\varphi} \dot{\theta} k_1 + g_1 \tau_\psi,\tag{2.67}$$

donde  $k_1 = \frac{(I_y - I_x)}{I_z}$ ,  $g_1 = \frac{l}{I_z}$ .  $I_x$ ,  $I_y$ ,  $I_z$  son los elementos de la diagonal principal de la Matriz de Inercia.

Utilizando las siguientes definiciones  $x_3 = \psi$ ,  $x_4 = \dot{\psi}$ . La representación en espacio estado del sistema es:

$$\dot{x}_3 = x_4\tag{2.68}$$

$$\dot{x}_4 = \dot{\varphi} \dot{\theta} k_1 + g_1 \tau_\psi,\tag{2.69}$$

$\tau_\psi$  esta definido como:

$$\tau_\psi = \frac{1}{g_1}(-\dot{\varphi}\dot{\theta}k_1 + u_{\alpha\psi}), \quad (2.70)$$

sustituyendo 2.70, la representación en espacio estado es:

$$\dot{x}_3 = x_4 \quad (2.71)$$

$$\dot{x}_4 = u_{\alpha\psi}. \quad (2.72)$$

Aplicando la estructura de la ecuación 2.56,  $x_4 = \varphi(x_3) = -\alpha_\psi x_3$ , y reescribiendo el sistema, se obtiene la siguiente expresión:

$$\dot{x}_3 = -\alpha_\psi x_3 \quad (2.73)$$

$$\dot{x}_4 = u_{\alpha\psi}. \quad (2.74)$$

Se elige una función de Lyapunov:

$$V_\psi(x_3) = \frac{1}{2}x_3^2, \quad (2.75)$$

y la entrada  $u_{\alpha\psi}$  tiene la siguiente estructura:

$$u_{\alpha\psi} = -k_\psi z - \frac{\partial V_\psi(\eta)}{\partial \eta} g(\eta) + \frac{\partial(\varphi)}{\partial \eta} [f(\eta) + g(\eta)\zeta], \quad (2.76)$$

donde  $\zeta = x_4$ ,  $\eta = x_3$ ,  $f(\eta) = 0$ ,  $g(\eta) = 1$ .

Sustituyendo las últimas consideraciones en 2.76 se obtiene la entrada  $u_{\alpha\psi}$ :

$$u_{\alpha\psi} = -\alpha_\psi x_4 - x_3 - k_\psi (x_4 + \alpha_\psi x_3). \quad (2.77)$$

### Subsistema $x-\theta$

Sea el siguiente subsistema el cual describe el movimiento traslacional en el eje  $x$  y el ángulo de cabeceo. Note que éste ya no puede ser subdividido ya que se trata de un sistema subactuado,

es decir, el movimiento de traslación en el eje  $x$  es el resultado de modificar el ángulo de cabeceo.

$$\begin{aligned}\ddot{x} &= -\frac{1}{m}\sin(\theta)u(.) \\ \ddot{\theta} &= \dot{\varphi}\dot{\psi}k_2 + g_2\tau_\theta,\end{aligned}\tag{2.78}$$

donde  $k_2 = \frac{(I_x - I_z)}{I_y}$  y  $g_2 = \frac{l}{I_y}$ . Note que  $u$  depende de  $x_7$  y  $x_{11}$  ( $u(x_7, x_{11})$ ) como se observa en 2.7.1. El sistema  $x - \theta$ , tiene las siguientes consideraciones  $x_5 = x$ ,  $x_6 = \dot{x}$ ,  $x_7 = \theta$ ,  $x_8 = \dot{\theta}$ .

La representación en espacio estado es:

$$\begin{aligned}\dot{x}_5 &= x_6 \\ \dot{x}_6 &= -\frac{1}{m}\sin(x_7)u(.) \\ \dot{x}_7 &= x_8 \\ \dot{x}_8 &= \dot{\varphi}\dot{\psi}k_2 + g_2\tau_\theta.\end{aligned}$$

Tomando la primera y la segunda ecuación de la representación en espacio estado:

$$\begin{aligned}\dot{x}_5 &= x_6 \\ \dot{x}_6 &= -\frac{1}{m}\sin(x_7)u(.),\end{aligned}\tag{2.79}$$

y eligiendo una entrada  $u_{1x}$

$$u_{1x} = \sin(x_7) = -\frac{m}{u(.)} [u_{ax}],\tag{2.80}$$

podemos sustituir 2.80, en 2.79:

$$\begin{aligned}\dot{x}_5 &= x_6 \\ \dot{x}_6 &= u_{ax}.\end{aligned}\tag{2.81}$$

por lo tanto ahora el objetivo es hallar la entrada  $u_{ax}$ . De acuerdo al sistema 2.81 y utilizando la estructura de 2.56,  $x_6 = \varphi(x_5) = -\alpha_{1x}x_5$ , se obtiene la siguiente representación:

$$\begin{aligned}\dot{x}_5 &= \alpha_{1x}x_5 \\ \dot{x}_6 &= u_{ax}.\end{aligned}\tag{2.82}$$

Se propone la siguiente función de Lyapunov:

$$V_{1x}(x_5) = \frac{1}{2}x_5^2, \quad (2.83)$$

así que la entrada  $u_{ax}$  es,

$$u_{ax} = -k_{1x}z - \frac{\partial V_{1x}(\eta)}{\partial \eta}g(\eta) + \frac{\partial(\varphi)}{\partial \eta} [f(\eta) + g(\eta)\zeta], \quad (2.84)$$

donde  $\zeta = x_6$ ,  $\eta = x_5$ ,  $z = x_6 + \alpha_{1x}x_5$ ,  $f(\eta) = 0$ ,  $g(\eta) = 1$ . Sustituyendo las últimas consideraciones en 2.84, obtenemos:

$$u_{ax} = (-x_5 - \alpha_{1x}x_6 - k_{1x}(x_6 + \alpha_{1x}x_5)), \quad (2.85)$$

y sustituyendo en 2.80

$$u_{1x} = -\frac{m}{u(\cdot)} (-x_5 - \alpha_{1x}x_6 - k_{1x}(x_6 + \alpha_{1x}x_5)). \quad (2.86)$$

La variable  $\dot{x}_7$  se agrega, resultando así en la siguiente representación de espacio estado:

$$\begin{aligned} \dot{x}_5 &= x_6 \\ \dot{x}_6 &= -\frac{1}{m}\sin(x_7)u(\cdot) \\ \dot{x}_7 &= x_8. \end{aligned} \quad (2.87)$$

Utilizando la estructura 2.56, el sistema 2.87 puede representarse como a continuación se muestra:

$$\dot{\eta} = \begin{bmatrix} x_6 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{u(\cdot)}{m} \end{bmatrix} \sin(x_7) \quad (2.88)$$

$$\dot{\zeta} = x_8 = u_{2x}, \quad (2.89)$$

donde  $\eta = [x_5 \ x_6]^T$ ,  $f(\eta) = [x_6 \ 0]^T$ ,  $g(\eta) = \left[0 \ -\frac{u(\cdot)}{m}\right]^T \sin(x_7)$ ,  $\zeta = x_7$ .

Una nueva  $V$  se propone, la función de Lyapunov aumentada considera el  $V$  descrito en 2.83:

$$V_{2x}(x_5, x_6) = \frac{1}{2}x_5^2 + \frac{\alpha_{2x}}{2}(x_6 + \alpha_{1x}x_5)^2. \quad (2.90)$$

Nótese que  $x_6 + \alpha_2 x_5$ , es el valor de  $z$ , donde  $\alpha_{2x}$  es un parámetro propuesto para establecer la velocidad de convergencia. La entrada virtual  $\varphi_{1x}$  está definida por  $u_{1x}$

$$\varphi_{1x}(x_5, x_6) = -\frac{m}{u(\cdot)}(-x_5 - \alpha_{1x}x_6 - k_{1x}(x_6 + \alpha_{1x}x_5)), \quad (2.91)$$

entonces las derivadas parciales de  $\varphi_{1x}$  y  $V_{2x}$  se obtienen de la siguiente manera:

$$\begin{aligned} \frac{\partial(\varphi_{1x}(x_5, x_6))}{\partial x_5} &= \frac{m}{u(\cdot)}(\alpha_{1x}k_{1x} + 1) \\ \frac{\partial(\varphi_{1x}(x_5, x_6))}{\partial x_6} &= \frac{m}{u(\cdot)}(\alpha_{1x} + k_{1x}) \\ \frac{\partial(V_{2x}(x_5, x_6))}{\partial x_5} &= \alpha_2 x_5 \alpha_{1x}^2 + \alpha_{2x} x_6 \alpha_{1x} + x_5 \\ \frac{\partial(V_{2x}(x_5, x_6))}{\partial x_6} &= \alpha_2 x_6 + \alpha_{1x} \alpha_{2x} x_5. \end{aligned}$$

De acuerdo con 2.63 la entrada es:

$$\begin{aligned} u_{2x} &= \begin{bmatrix} \frac{\partial(\varphi_{1x}(x_5, x_6))}{\partial x_5} & \frac{\partial(\varphi_{1x}(x_5, x_6))}{\partial x_6} \end{bmatrix} \begin{bmatrix} \frac{x_6}{\cos(x_7)} \\ \frac{u(\cdot)}{m} \sin(x_7) \\ \frac{1}{\cos(x_7)} \end{bmatrix} \\ &\quad - \begin{bmatrix} \frac{\partial(V_{2x}(x_5, x_6))}{\partial x_5} & \frac{\partial(V_{2x}(x_5, x_6))}{\partial x_6} \end{bmatrix} \begin{bmatrix} 0 \\ \frac{u(\cdot)}{m} \\ \frac{1}{\cos(x_7)} \end{bmatrix} \\ &\quad - \frac{k_{2x}}{\cos(x_7)} (\sin(x_7) - \varphi_{1x}(x_5, x_6)). \\ &= \frac{1}{m} \frac{u(\cdot)}{\cos(x_7)} (\alpha_{2x} x_6 + \alpha_{1x} \alpha_{2x} x_5) - \frac{k_{2x}}{\cos(x_7)} \left( \sin(x_7) - \frac{m}{u(\cdot)} (x_5 + \alpha_{1x} x_6 + k_{1x} (x_6 + \alpha_{1x} x_5)) \right) \\ &\quad - \frac{1}{\cos(x_7)} (\sin(x_7)) (\alpha_1 + k_{1x}) + \frac{m}{u(\cdot)} \frac{x_6}{\cos(x_7)} (\alpha_{1x} k_{1x} + 1). \end{aligned} \quad (2.92)$$

Finalmente agregamos la última variable de estado

$$\begin{aligned} \dot{x}_5 &= x_6 \\ \dot{x}_6 &= -\frac{1}{m} \sin(x_7) u(\cdot) \\ \dot{x}_7 &= x_8 \\ \dot{x}_8 &= \dot{\varphi} \psi k_2 + g_2 \tau_\theta, \end{aligned} \quad (2.93)$$

definimos  $\tau_\theta$

$$\tau_\theta = \frac{1}{g_2}(-\dot{\varphi}\dot{\psi}k_2 + \tau_{\theta a}), \quad (2.94)$$

sustituyendo el valor de  $\tau_\theta$  en 2.93

$$\begin{aligned} \dot{x}_5 &= x_6 \\ \dot{x}_6 &= -\frac{1}{m}\sin(x_7)u(.) \\ \dot{x}_7 &= x_8 \\ \dot{x}_8 &= \tau_{\theta a}. \end{aligned}$$

Similar a los previos subsistemas, el último subsistema puede ser representado como:

$$\dot{\eta} = \begin{bmatrix} x_6 \\ -u(.)\sin(x_7) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \zeta \quad (2.95)$$

$$\dot{\zeta} = \tau_{\theta a}, \quad (2.96)$$

en donde  $\eta = [x_5 \ x_6 \ x_7]^T$ ,  $f(\eta) = [x_6 \ -u(.)\sin(x_7) \ 0]^T$ ,  $g(\eta) = [0 \ 0 \ 1]^T$ ,  $\zeta = x_8$ .

El  $V$  de este tercer paso necesita considerar el  $V$  del paso anterior,

$$V_{3x}(x_5, x_6, x_7) = \frac{1}{2}x_5^2 + \frac{\alpha_{2x}}{2}(x_6 + \alpha_{2x}x_5)^2 + \frac{\alpha_{3x}}{2}\left(\sin(x_7) + \frac{m}{u(.)}(-x_5 - \alpha_{1x}x_6 - k_{1x}(x_6 + \alpha_{1x}x_5))\right)^2 \quad (2.97)$$

La entrada virtual  $\varphi_{2x}$  está definida por  $u_{2x}$

$$\begin{aligned} \varphi_{2x}(x_5, x_6, x_7) &= \frac{1}{m} \frac{u(.)}{\cos(x_7)} (\alpha_{2x}x_6 + \alpha_{1x}\alpha_{2x}x_5) \\ &- \frac{k_{2x}}{\cos(x_7)} \left( \sin(x_7) - \frac{m}{u(.)} (x_5 + \alpha_{1x}x_6 + k_{1x}(x_6 + \alpha_{1x}x_5)) \right) \\ &- \frac{1}{\cos(x_7)} (\sin(x_7))(\alpha_{1x} + k_{1x}) + \frac{m}{u(.)} \frac{x_6}{\cos(x_7)} (\alpha_{1x}k_{1x} + 1). \end{aligned} \quad (2.98)$$



Finalmente, la entrada  $\tau_{\theta_a}$  es

$$\begin{aligned} \tau_{\theta_a} = & \left[ \frac{\partial(\varphi_{2x}(x_5, x_6, x_7))}{\partial x_5} \quad \frac{\partial(\varphi_{2x}(x_5, x_6, x_7))}{\partial x_6} \quad \frac{\partial(\varphi_{2x}(x_5, x_6, x_7))}{\partial x_7} \right] \begin{bmatrix} x_6 \\ -\frac{u(\cdot)}{m} \sin(x_7) \\ x_8 \end{bmatrix} \\ & - \left[ \frac{\partial(V_{3x}(x_5, x_6, x_7))}{\partial x_5} \quad \frac{\partial(V_{3x}(x_5, x_6, x_7))}{\partial x_6} \quad \frac{\partial(V_{3x}(x_5, x_6, x_7))}{\partial x_7} \right] \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\ & - k_{3x}(x_8 - \varphi_{2x}(x_5, x_6, x_7)). \end{aligned} \quad (2.99)$$

### Subsistema $y$ - $\varphi$

Utilizando el siguiente subsistema, el cual describe el movimiento en el eje  $Y$  y el ángulo de alabeo:

$$\begin{aligned} \ddot{y} &= \frac{u}{m} \cos(\theta) \sin(\varphi) \\ \ddot{\varphi} &= \dot{\theta} \dot{\psi} k_3 + g_3 \tau_\varphi, \end{aligned} \quad (2.100)$$

donde  $k_3 = \frac{(I_y - I_z)}{I_x}$  y  $g_3 = \frac{l}{I_x}$ . Para este subsistema se harán las siguientes consideraciones:  $x_9 = y$ ,  $x_{10} = \dot{y}$ ,  $x_{11} = \varphi$ ,  $x_{12} = \dot{\varphi}$ . La representación en espacio estado es la siguiente:

$$\begin{aligned} \dot{x}_9 &= x_{10} \\ \dot{x}_{10} &= -\frac{u(\cdot)}{m} \cos(x_7) \sin(x_{11}) \\ \dot{x}_{11} &= x_{12} \\ \dot{x}_{12} &= \dot{\theta} \dot{\psi} k_3 + g_3 \tau_\varphi. \end{aligned}$$

Para el primer paso, se tomarán las primeras dos variables de estado:

$$\begin{aligned} \dot{x}_9 &= x_{10} \\ \dot{x}_{10} &= -\frac{u(\cdot)}{m} \cos(x_7) \sin(x_{11}), \end{aligned} \quad (2.101)$$

se toma  $u_{1y}$  como una entrada:

$$u_{1y} = \sin(x_{11}) = \frac{m}{u(\cdot) \cos(x_7)} (u_{\alpha y}). \quad (2.102)$$

Sustituyendo 2.102 y reescribiendo el conjunto de ecuaciones se obtiene la siguiente representación:

$$\begin{aligned} \dot{x}_9 &= x_{10} \\ \dot{x}_{10} &= u_{\alpha y}, \end{aligned} \quad (2.103)$$

Como en los subsistemas previos, consideramos  $x_{10} = \varphi(x_9) = -\alpha_{1y}x_9$ . Reescribiendo el sistema 2.103, se obtiene:

$$\begin{aligned} \dot{x}_9 &= \alpha_{1y}x_9 \\ \dot{x}_{10} &= u_{\alpha y}. \end{aligned} \quad (2.104)$$

Para el subsistema  $y - \varphi$ , la función de Lyapunov es:

$$V_{1y}(x_9) = \frac{1}{2}x_9^2 \quad (2.105)$$

La entrada de  $u_{\alpha y}$  es

$$u_{\alpha y} = -k_{1y}z - \frac{\partial V_{1y}(\eta)}{\partial \eta} g(\eta) + \frac{\partial(\varphi)}{\partial \eta} (f(\eta) + g(\eta)\zeta) \quad (2.106)$$

en donde  $\zeta = x_{10}$ ,  $\eta = x_9$ ,  $z = x_{10} + \alpha_{3y}x_9$ ,  $f(\eta) = 0$ ,  $g(\eta) = 1$ .

Sustituyendo 2.106 con las consideraciones, obtenemos:

$$u_{\alpha y} = -k_{1y} (x_{10} - \alpha_{3y}x_9) - x_9 - \alpha_{3y} (x_{10}). \quad (2.107)$$

Mediante la sustitución de 2.107 en 2.102, la entrada  $u_{1y}$  queda como:

$$u_{1y} = \frac{m}{u(\cdot) \cos(x_7)} (-k_{1y} (x_{10} - \alpha_{1y}x_9) - x_9 - \alpha_{1y} (x_{10})). \quad (2.108)$$

En el segundo paso, la variable  $\dot{x}_{11}$  se agrega, por lo cual, la representación en espacio de estados es:

$$\dot{x}_9 = x_{10} \quad (2.109)$$

$$\dot{x}_{10} = -\frac{u(\cdot)}{m} \cos(x_7) \sin(x_{11})$$

$$\dot{x}_{11} = x_{12},$$

$$(2.110)$$

utilizando 2.56, el sistema 2.109 puede ser reescrito como:

$$\dot{\eta} = \begin{bmatrix} x_{10} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{u(\cdot) \cos(x_7)}{m} \end{bmatrix} \sin(x_{11}) \quad (2.111)$$

$$\dot{\zeta} = x_{12} = u_{2y}, \quad (2.112)$$

en donde  $\eta = [x_9 \ x_{10}]^T$ ,  $f(\eta) = [x_{10} \ 0]^T$ ,  $g(\eta) = \left[0 \ -\frac{u(\cdot) \cos(x_7)}{m}\right]^T \sin(x_{11})$ ,  $\zeta = x_{11}$ .

Se propone una nueva  $V$ , la nueva función de Lyapunov aumentada, considera el valor de 2.83,

$$V_{2y}(x_9, x_{10}) = \frac{1}{2}x_9^2 + \frac{\alpha_{2y}}{2}(x_{10} + \alpha_{1y}x_9)^2, \quad (2.113)$$

$\alpha_{2y}$  es el parámetro propuesto, como en el subsistema previo. La entrada virtual  $\varphi_{1y}$  esta definida por  $u_{1y}$

$$\varphi_{1y}(x_9, x_{10}) = \frac{m}{u(\cdot) \cos(x_7)} (-k_{1y}(x_{10} - \alpha_{1y}x_9) - x_9 - \alpha_{1y}(x_{10})), \quad (2.114)$$

en donde la evaluación de las derivadas parciales  $\varphi_{1y}$  y  $V_{2y}$

$$\begin{aligned} \frac{\partial (\varphi_{1x}(x_9, x_{10}))}{\partial x_9} &= \frac{m}{u(\cdot) \cos(x_7)} (\alpha_{1y}k_{1y} + 1) \\ \frac{\partial (\varphi_{1y}(x_9, x_{10}))}{\partial x_6} &= \frac{m}{u(\cdot)} (\alpha_{1y} + k_{1y}) \\ \frac{\partial (V_{2y}(x_9, x_{10}))}{\partial x_9} &= \alpha_{2y}x_9\alpha_{1y}^2 + \alpha_{2y}x_{10}\alpha_{1y} + x_9 \\ \frac{\partial (V_{2y}(x_9, x_{10}))}{\partial x_{10}} &= \alpha_{2y}x_{10} + \alpha_{1y}\alpha_{2y}x_9. \end{aligned}$$

La entrada queda como:

$$\begin{aligned}
u_{2y} = & \left[ \frac{\partial (\varphi_{1y}(x_9, x_{10}))}{\partial x_9} \quad \frac{\partial (\varphi_{1y}(x_9, x_{10}))}{\partial x_{10}} \right] \begin{bmatrix} \frac{x_{10}}{\cos(x_{11})} \\ -\frac{u(\cdot)}{m} \cos(x_7) \sin(x_{11}) \\ \frac{u(\cdot)}{m} \cos(x_7) \end{bmatrix} \\
& - \left[ \frac{\partial (V_{1y}(x_9, x_{10}))}{\partial x_9} \quad \frac{\partial (V_{1y}(x_9, x_{10}))}{\partial x_{10}} \right] \begin{bmatrix} 0 \\ \frac{u(\cdot)}{m} \cos(x_7) \\ \frac{u(\cdot)}{m} \cos(x_7) \end{bmatrix} \\
& - \frac{k_{2y}}{\cos(x_{11})} (\sin(x_{11}) - \varphi_1(x_9, x_{10})).
\end{aligned} \tag{2.115}$$

$$\begin{aligned}
u_{2y} = & -\frac{1}{\cos(x_{11})} (\sin(x_{11})) (\alpha_{1y} + k_{1y}) - \frac{1}{m} u(\cdot) \frac{\cos(x_7)}{\cos(x_{11})} (\alpha_{2y} x_{10} + \alpha_{1y} \alpha_{2y} x_9) \\
& - \frac{k_{2y}}{\cos(x_{11})} \left( \sin(x_{11}) + \frac{m}{u(\cdot) \cos(x_7)} (x_9 + \alpha_{1y} x_{10} + k_{1y} (x_{10} + \alpha_{1y} x_9)) \right) \\
& - \frac{m}{u(\cdot) \cos(x_7) \cos(x_{11})} (\alpha_{1y} k_{1y} + 1).
\end{aligned} \tag{2.116}$$

Para el último paso se tiene,

$$\begin{aligned}
\dot{x}_9 &= x_{10} \\
\dot{x}_{10} &= -\frac{u(\cdot)}{m} \cos(x_7) \sin(x_{11}) \\
\dot{x}_{11} &= x_{12} \\
\dot{x}_{12} &= \dot{\theta} \dot{\psi} k_3 + g_3 \tau_\varphi,
\end{aligned} \tag{2.117}$$

definimos  $\tau_\varphi$

$$\tau_\varphi = \frac{1}{g_3} (-\dot{\theta} \dot{\psi} k_2 + \tau_{\varphi a}). \tag{2.118}$$

Sustituyendo 2.118 en el sistema descrito en 2.117 se llega a:

$$\begin{aligned}
\dot{x}_9 &= x_{10} \\
\dot{x}_{10} &= -\frac{u(\cdot)}{m} \cos(x_7) \sin(x_{11}) \\
\dot{x}_{11} &= x_{12} \\
\dot{x}_{12} &= \tau_{\varphi a}.
\end{aligned} \tag{2.119}$$

Utilizando la estructura de 2.56, obtenemos el último sistema como a continuación se muestra:

$$\dot{\eta} = \begin{bmatrix} x_{10} \\ -u(\cdot) \cos(x_7) \sin(x_{11}) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \zeta \quad (2.120)$$

$$\dot{\zeta} = \tau_{\theta a}, \quad (2.121)$$

en donde  $\eta = [x_9 \ x_{10} \ x_{11}]^T$ ,  $f(\eta) = [x_{10} \ u(\cdot) \cos(x_7) \sin(x_{11}) \ 0]^T$ ,  $g(\eta) = [0 \ 0 \ 1]^T$ ,  $\zeta = x_{12}$ . La  $V$  para el tercer paso incluye el valor de  $V_2$

$$\begin{aligned} V_{3y}(x_9, x_{10}, x_{11}) &= \frac{1}{2}x_9^2 + \frac{\alpha_{2y}}{2}(x_{10} + \alpha_{1y}x_9)^2 \\ &+ \frac{\alpha_{3y}}{2} \left( \sin(x_{11}) - \frac{m}{u(\cdot) \cos(x_7)} (-x_9 - \alpha_{1y}x_{10} - k_{1y}(x_{10} + \alpha_{1y}x_9)) \right)^2. \end{aligned} \quad (2.122)$$

$\varphi_{2y}$  es definida por  $u_{2y}$

$$\begin{aligned} &= -\frac{1}{\cos(x_{11})} (\sin(x_{11})) (\alpha_{1y} + k_1) - \frac{1}{m} u(\cdot) \frac{\cos(x_7)}{\cos(x_{11})} (\alpha_{5y}x_{10} + \alpha_{1y}\alpha_{2y}x_9) \\ &- \frac{k_{2y}}{\cos(x_{11})} \left( \sin(x_{11}) + \frac{m}{u(\cdot) \cos(x_7)} (x_9 + \alpha_{1y}x_{10} + k_{1y}(x_{10} + \alpha_{1y}x_9)) \right) \\ &- \frac{m}{u(\cdot) \cos(x_7) \cos(x_{11})} \frac{x_{10}}{(\alpha_{1y}k_{1y} + 1)}. \end{aligned} \quad (2.123)$$

Finalmente, la entrada  $\tau_{\varphi a}$  está descrita como a continuación se muestra:

$$\begin{aligned} \tau_{\varphi a} &= \begin{bmatrix} \frac{\partial(\varphi_{2y}(x_9, x_{10}, x_{11}))}{\partial x_9} & \frac{\partial(\varphi_{2y}(x_9, x_{10}, x_{11}))}{\partial x_{10}} & \frac{\partial(\varphi_{2y}(x_9, x_{10}, x_{11}))}{\partial x_{11}} \end{bmatrix} \begin{bmatrix} x_{10} \\ -\frac{u(\cdot)}{m} \cos(x_7) \sin(x_7) \\ x_{12} \end{bmatrix} \\ &- \begin{bmatrix} \frac{\partial(v_{2y}(x_9, x_{10}, x_{11}))}{\partial x_9} & \frac{\partial(v_{2y}(x_9, x_{10}, x_{11}))}{\partial x_{10}} & \frac{\partial(v_{2y}(x_9, x_{10}, x_{11}))}{\partial x_{11}} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\ &- k(x_{12} - \varphi_{2y}(x_9, x_{10}, x_{11})). \end{aligned} \quad (2.124)$$

## 2.8 Conclusiones

En este Capítulo se desarrollaron varios algoritmos de control de manera satisfactoria, las cuales se implementaron al modelo del cuadricóptero. EL objetivo es poder encontrar una técnica que permita hacer seguimiento de trayectorias, por lo que se desarrollaron controles tanto robustos como óptimos. Los resultados se mostrarán de manera experimental en el Capítulo 4.

# CAPÍTULO 3

## DISEÑO Y DESARROLLO DEL PROTOTIPO EXPERIMENTAL

### 3.1 Introducción

El prototipo utilizado para los experimentos, está basado en el marco ó “chasis”, HobbyKing versión “X666”, este chasis está construido en fibra de vidrio y aluminio con una longitud de 666 mm. En este chasis va montado un autopiloto Pixhawk, se agregaron también diversos sensores externos de acuerdo al control que se estaba implementando, tales como: GPS, Px4 Flow, Sensor láser Swep 360 y la Cámara monocular Pixy. Estos sensores se explicarán a detalle en las siguientes secciones. La figura 3.1 muestra la versión utilizada en los experimentos de la navegación en presencia de obstáculos.

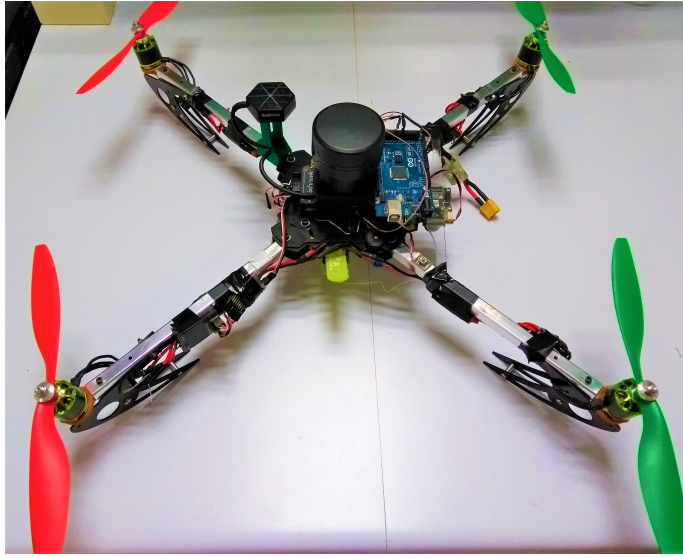


Figura 3.1: Versión final de la plataforma experimental utilizada en la navegación en presencia de obstáculos

La siguiente gráfica 3.2 muestra el esquema general de conexión de la plataforma experimental. Los sensores principales se explican más adelante sin embargo podemos notar que también se montó un Arduino mega, este microcontrolador se utilizó para leer el sensor láser Sweep v1. También se utilizó un radio control Futaba de 7 canales para poder controlar de forma manual al vehículo. Es importante mencionar que la batería no siempre fue la misma, durante todo el proyecto de tesis se utilizaron más de 5 tipos de baterías. Se utilizaron también controladores de velocidad de 30A y el sensor sweep se alimentó con un regulador BEC de 5 volts.



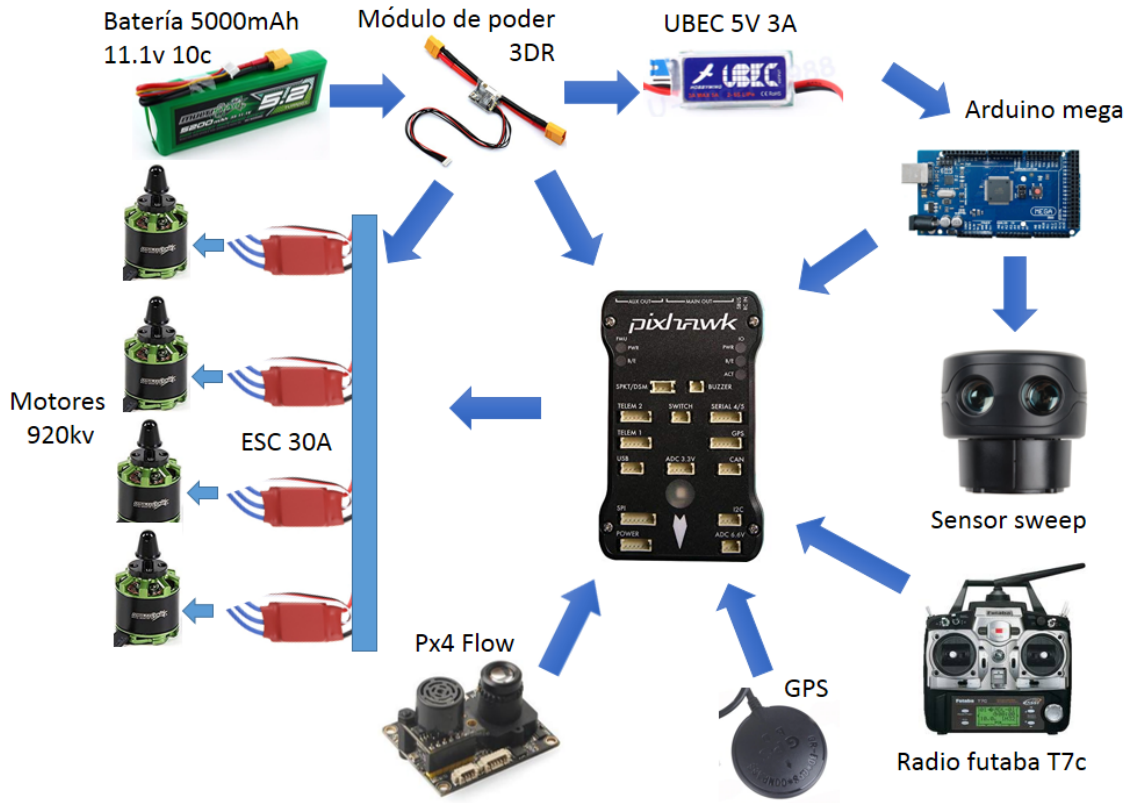


Figura 3.2: Esquema de conexiones de la plataforma experimental basada en un marco de 666 mm.

## 3.2 Autopiloto Pixhawk

El autopiloto Pixhawk es un proyecto independiente de hardware abierto cuyo objetivo es proveer una plataforma estándar, fácilmente disponible, de alta calidad y de bajo costo [23]. Esta diseñado para propósitos académicos, aficionados y comunidades de desarrolladores. Existen muchas plataformas que surgieron a partir del proyecto estudiantil del Pixhawk, tales como:

- mRo Pixhawk,
- HKPilot32,

- Pixfalcon,
- DroPix,
- Pixracer,
- MindPX,
- MindRacer,
- Pixhack v3,
- Pixhawk 2,
- Pixhawk 3 Pro,
- Pixhawk 4,
- Pixhack v5.

En esta tesis se hizo uso de la primera versión, el Pixhawk 1 3.3, el cual es un controlador de vuelo muy popular para propósitos generales y está basado en el proyecto FMUv2, el cual es un diseño de hardware abierto y combina la funcionalidad del px4FMU. Originalmente fue manufacturado por 3DR, esta placa era la plataforma de microcontrolador estándar original para PX4.



Figura 3.3: Controlador de vuelo Pixhawk 1 manufacturado por 3DR.

Las principales características de este autopiloto son las siguientes:

- Microcontrolador principal: STM32F427,
- CPU: ARM® Cortex® M4 corriendo a 180 MHz con una FPU de precisión simple,
- RAM: 256 KB SRAM tipo (L1),
- Co-procesador para fallos: STM32F100,
  - CPU: ARM Cortex M3 a 24 MHz,
  - RAM: 8 KB SRAM,
- Wifi: ESP8266, externo,
- GPS: Compatible con U-Blox® 7/8 (Hobbyking®) y U-Blox 6 (3D Robotics),
- Flujo óptico: Compatible con *PX4 Flow*.
- Entradas de energía redundantes con corte automático por error,
- Botón de seguridad externo,
- Indicador Led multicolor,
- Indicador auditivo multitono,
- Memoria Micro SD de alta tasa de escritura y periodos prolongados.

### Opciones de conectividad

- 1 x I2C
- 1 x CAN (2 opcional)
- 1 x ADC

- 4 x UART (2 con control de flujo)
- 1 x Consola de salida
- 8 x Salidas PWM
- 6 x Salidas PWM, GPIO, o entrada PWM
- Puerto S.BUS, PPM y entrada para spektrum
- Puerto S.BUS de salida.

Este auto-piloto también tiene integrado varios sensores [23], lo cual le permite tener mediciones de sus ángulos y su posición:

- Giroscopio de 16 bits ST Micro L3GD20H
- Acelerómetro y magnetómetro de 14 bits ST Micro LSM303D
- Acelerómetro y magnetómetro de 3 ejes Invensense MPU 6000
- Barómetro MEAS MS5611

Este autopiloto no está limitado a un vehículo de cuatro rotores sino que se puede utilizar para controlar hasta un vehículo de 8 rotores, también se puede implementar en un vehículo terrestre o en un avión de ala fija. Se decidió utilizar esta tarjeta debido a que es de código abierto y el amplio soporte que existe en la red.

### **3.3 GPS**

La mayoría de las pruebas se realizaron en el exterior por lo que fue necesario contar con un sensor de posicionamiento respecto de la tierra, así que dependiendo el enfoque se utilizó un GPS o un sensor por flujo óptico. En esta sección se documentará a detalle el GPS utilizado.

La siguiente figura muestra el GPS 3.4, Ublox Neo-M8N el cual está diseñado con un conector específico para el Pixhawk y se comunica a él mediante un protocolo serial.



Figura 3.4: GPS Ublox Neo-M8N.

Este sensor tiene características que lo hacen ideal para este tipo de aplicaciones, entre ellos: una rápida búsqueda de satélites, únicamente necesita 10 segundos para encontrar 6 satélites en espacio abierto. Trae incluido una brújula el cual provee la señal de guiñada al vehículo, tiene una velocidad de actualización de más de 18Hz. Uno de los factores más importantes es su precisión, que a diferencia de las versiones anteriores NEO-7N que tenían una precisión entre 1.4 y 1.7 metros, éste tiene una precisión de 0.6 metros. Tiene también soporte de GPS+BD+SBAS, GPS+GLONASS+SBAS. La siguiente lista abarca otras características también importantes:

- Tipo de receptor: 72-canales, u-blox M8
- Soporte de: GPS/QZSS L1 C/A, GLONASS L10F, BeiDou B1, SBAS L1 C/A: WAAS, EGNOS, MSAS Galileo-ready E1B/C (NEO-M8N)
- Velocidad de actualización GNSS: mas de 18 HZ
- Concurrencia GNSS: arriba de 10 Hz

- Inicio en frío: 26s
- Re-adquisición de señal: 1.5 s
- Temperatura de operación:  $-40^{\circ}$  C a  $85^{\circ}$  C
- Temperatura de almacenamiento:  $-40^{\circ}$  C a  $105^{\circ}$  C (NEO-M8M)
- Certificación: ISO 16750
- Voltaje de entrada: 1.65 V a 3.6 V (NEO-M8M),
- Consumo eléctrico: 23 mA-3.0 V (continuo), 5mA-3.0 V, en modo ahorro(1 Hz, únicamente modo GPS)

### 3.4 Px4 Flow

El px4 Flow 3.5 es una cámara inteligente con una resolución de  $752 \times 480$  el cual calcula el flujo óptico en una área recortada(4x) a 400HZ, con alta sensibilidad a la luz natural [25]. Desde la perspectiva de hardware, tiene las siguientes características:

- Tiene un CPU Cortex M4F a 168MHZ (168 + 64 KB RAM),
- Tiene un sensor de imagen de  $752 \times 480$  MT9V034 y un giroscopio de 3 ejes L3GD20.
- Tiene un lente de 16mm M12 con filtro de IR,
- Las dimensiones son de 45mm $\times$ 35mm,
- El consumo de energía es de 115mA a 5 volts.

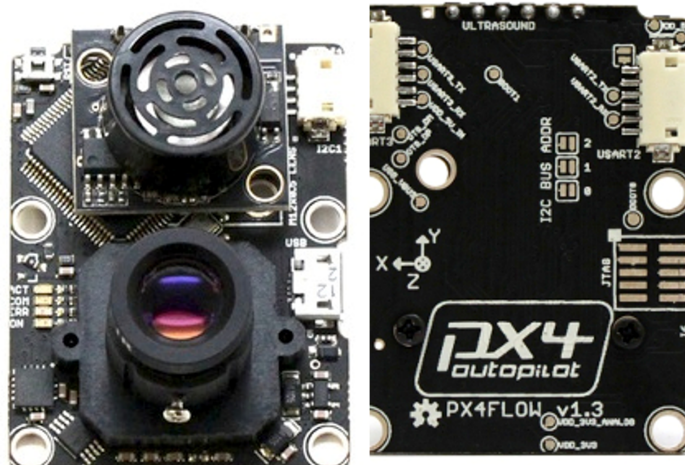


Figura 3.5: Sensor de flujo óptico px4-flow utilizada para estimar la posición del vehículo en el exterior

Los datos se obtienen través del un puerto 12C; se debe hacer una calibración previa para que las mediciones sean precisas. Este sensor devuelve la velocidad en pixeles, sin embargo no es difícil obtener las mediciones de velocidad en  $m/s$  ya que tiene integrado un sensor ultrasónico.

### 3.5 Sensor láser Sweep 360

El escáner láser Sweep V1 360° 3.6 es un sensor tipo LiDAR el cual ofrece una potente capacidad de detección de 360 grados. Es un sensor ideal para detectar y evitar obstáculos en robots móviles. Puede ser implementado en vehículos terrestres o aéreos [30]. Este sensor Sweep proporciona mediciones de largo alcance (hasta 40 metros) y rango preciso (1cm de resolución) a objetos, incluso en entornos soleados y ruidosos.



Figura 3.6: Sensor de barrido láser, utilizado para detectar los obstáculos durante el vuelo

Las características principales desde el punto de vista del hardware son las siguientes:

- Rango de hasta 40 metros,
- Campo de visión de 360 grados,
- Frecuencia de rotación de 2 a 10 HZ ajustable,
- UART de 3.3v a 5v,
- Conexión USB, i2c,
- Entrada de voltaje de 5 volts,
- Consumo de corriente de 400mA.

Este sensor provee 3 mediciones por cada punto encontrado: la distancia al objeto, el ángulo de detección y la salud de la medición. La resolución del ángulo es de 1 grado, es decir puede entregar como máximo 360 mediciones por vuelta. Tiene una apertura vertical de 0.5 grados. Pesa únicamente 120 g lo cual lo hace ideal para aplicaciones en vehículos aéreos respecto a otros que son más pesados.

### 3.6 Cámara Pixy

La cámara Pixy CMUCam 5 3.7 es un sensor de imagen muy poderoso el cual puede identificar tono y saturación en imágenes. Es robusto ante cambios en la exposición e iluminación. Tiene



varias interfaces para entregar la información leída tales como puerto serial, SPI, I2c, salidas digitales y analógicas [24].

Esta cámara puede procesar hasta 50 fotogramas por segundo. Tiene la capacidad de aislar patrones de colores con mucha facilidad, basta con hacer un entrenamiento oprimiendo un botón con el patrón que se desea detectar. Es de código abierto lo cual permite hacerle modificaciones para tareas específicas.



Figura 3.7: Camara Pixy el cual fue utilizado para aproximar la posición del vehículo respecto de un objetivo en tierra

Las características más importantes de su hardware son las siguientes:

- Tiene un Procesador dual core NXP LPC4330 a 204 MHZ
- Memoria RAM de 264Kb,
- Consumo de corriente de 140mA,
- Sensor de imagen Omnivision OV9715, 1/4" de 1280x800,
- Botón de entrenamiento,
- Puertos de comunicación: SPI, I2C, UART, USB o salida analoga/digital.
- Dimensiones: 50 mm x 54 mm x 2 mm,
- Peso de 25,5 g.

## **3.7 Conclusiones**

La plataforma experimental se realizó sobre un chasis comercial, es decir no se realizó ningún diseño. Sin embargo todos los componentes fueron seleccionados para cumplir una tarea específica. La plataforma experimental no sólo fue una, sino que hubieron muchas versiones, esto dependió en gran medida de la técnica de control que se utilizó así mismo de la tarea que se quería ejecutar. En general se puede concluir que la parte del montaje de los componentes de la plataforma experimental no tomó mucho tiempo. Sin embargo el aporte tecnológico fue la integración de todos los componentes para que se comportara como un único sistema.

# CAPÍTULO 4

## RESULTADOS

### 4.1 Introducción

En esta sección se presentan los resultados experimentales de cada algoritmo de control que se presentó en el Capítulo 2 e implementado en la plataforma que se describe en el Capítulo 3. Cada algoritmo se probó en diferentes condiciones debido a los requerimientos de cada técnica, por lo tanto los resultados pueden ser diferentes entre cada técnica debido al conjunto de sensores que se utilizaron por cada caso. Adicionalmente en la sección 4.6 se presentan resultados experimentales en las tareas de despegue y aterrizaje con el control subóptimo no lineal.

### 4.2 Control óptimo de horizonte infinito en tiempo discreto

#### 4.2.1 Prueba 1: Control en altura y posición

El sensor de flujo óptico Pxf4flow, provee mediciones de velocidad en pixeles por segundo, sin embargo también incluye un sensor ultrasónico lo que permite obtener la medición correspondiente a metros por segundo, estas mediciones son leídas en la tarjeta Pixhawk a una

velocidad de 200Hz. Las señales de velocidad obtenidas correspondientes a los ejes  $x$  y  $y$  son sometidas a un proceso de integración discreta para aproximar las posiciones longitudinales correspondientes.

La implementación de un control en modo estacionario requirió previamente de un control para la altura. Las pruebas realizadas se llevaron a cabo en el exterior con condiciones de viento moderado. Las matrices de penalización son las siguientes:

$$Q_{x,y} = 100 * \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 1800 \end{bmatrix}, R_{x,y} = 20. \quad (4.1)$$

$$Q_{z,\psi} = 1000 * \begin{bmatrix} 120 & 0 & 0 & 0 \\ 0 & 1800 & 0 & 0 \\ 0 & 0 & 1600 & 0 \\ 0 & 0 & 0 & 800 \end{bmatrix}, R_{z,\psi} = \begin{bmatrix} 30 & 0 \\ 0 & 30 \end{bmatrix}. \quad (4.2)$$

En la gráfica 4.1 se muestran las posiciones de los ejes cartesianos  $x$ ,  $y$  y  $z$ , aplicando un control óptimo de horizonte infinito utilizando la teoría del Regulador Cuadrático Lineal(LQR), para ello se utilizaron las matrices de penalización 4.1, 4.2, para la dinámica de  $x - \theta$ ,  $y - \varphi$  y  $z - \psi$  respectivamente.

Las consignas para vuelo estacionario son: para la posición  $x$  y  $y$  cero, para la altura  $z$ , es una constante mayor a cero, para estas pruebas se elevó el vehículo a 3.5 metros aproximadamente. La referencia para todos los ángulos son cero.

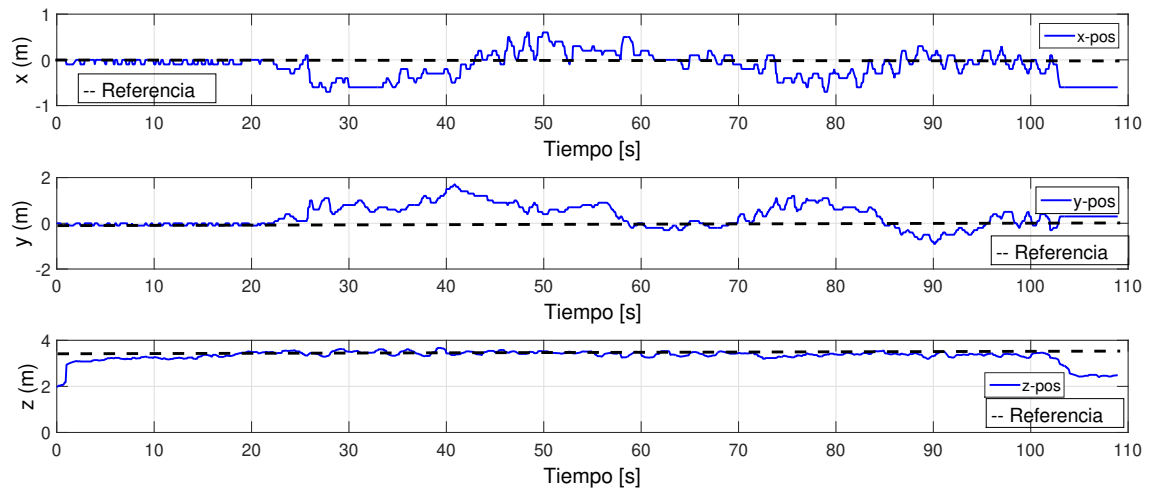


Figura 4.1: Señales de posición aplicando control óptimo de horizonte infinito para vuelo estacionario.

En la gráfica anterior se observa que las posiciones longitudinales se encuentran cercanas a la referencia, ésto es, los errores promedio son: 0.22, 0.44, 0.13 para los ejes  $x$ ,  $y$  y  $z$  respectivamente, podemos notar que el error promedio en el eje  $y$  es mayor sin embargo como se mencionó arriba estas pruebas se hicieron en el exterior, en presencia de perturbaciones externas tales como el viento.

La gráfica 4.2 muestra el comportamiento de las señales de orientación para los ángulos de *roll*, *pitch* y *yaw* respectivamente, aplicando la estrategia de control óptima descrita anteriormente. Se puede notar que el ángulo de *yaw* se mueve sobre la referencia de  $0^\circ$  ésto se debe a varios fenómenos, entre los más importantes: la planta, puesto que la medición del ángulo siempre tiene una deriva en un sentido ó en otro, también se debe a que los actuadores son motores, éstos idealmente deberían tener el mismo comportamiento pero al incluir partes mecánicas ésto ya no es del todo cierto.

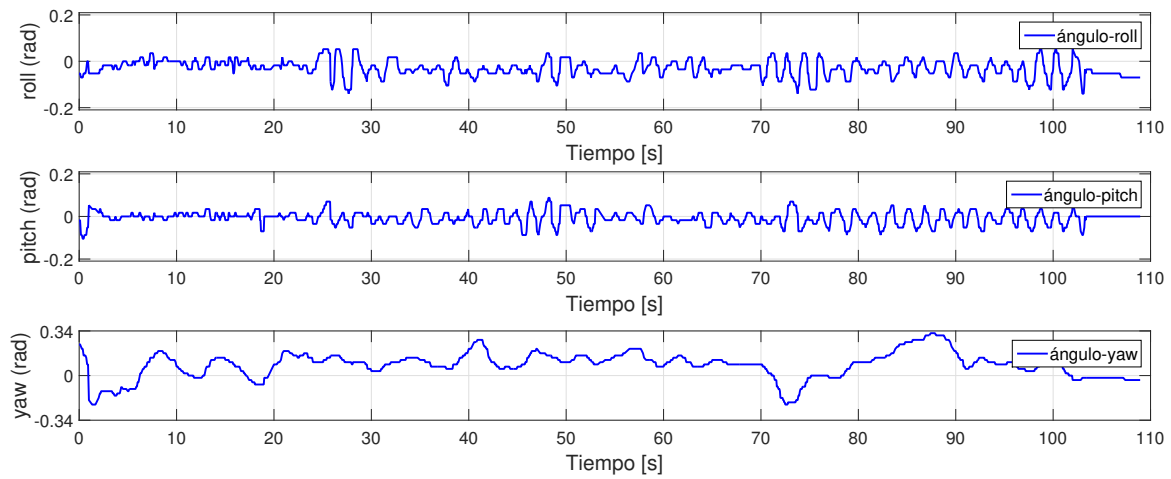


Figura 4.2: Señales de orientación aplicando control óptimo de horizonte infinito para vuelo estacionario.

Las velocidades longitudinales y angulares para este mismo experimento se muestran en la figura 4.3

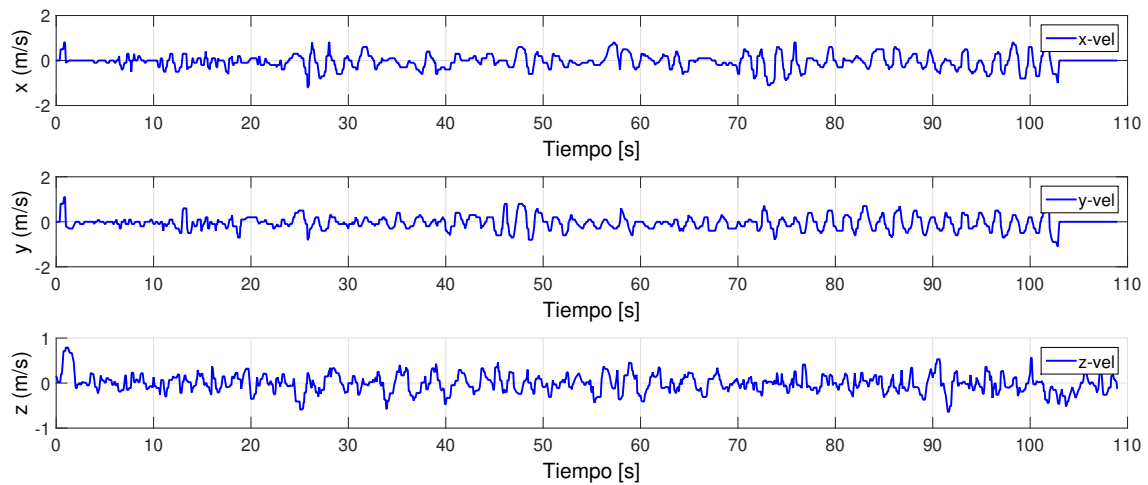


Figura 4.3: Señales de velocidad en los ejes  $x$ ,  $y$  y  $z$  aplicando control óptimo de horizonte infinito para vuelo estacionario.

Se puede observar que las velocidades longitudinales oscilan más respecto a los experi-

mentos posteriores (sección 4.5) donde se utiliza el GPS para las mediciones de posición  $x$  y  $y$ . Esto se debe a que se está utilizando un sensor de flujo óptico y a pesar de que se hace un post-procesamiento de filtrado a la señal existen aún variaciones en las velocidades que se deben al movimiento angular y no al movimiento de traslación. Estas variaciones también se pueden ver en las velocidades angulares 4.4.

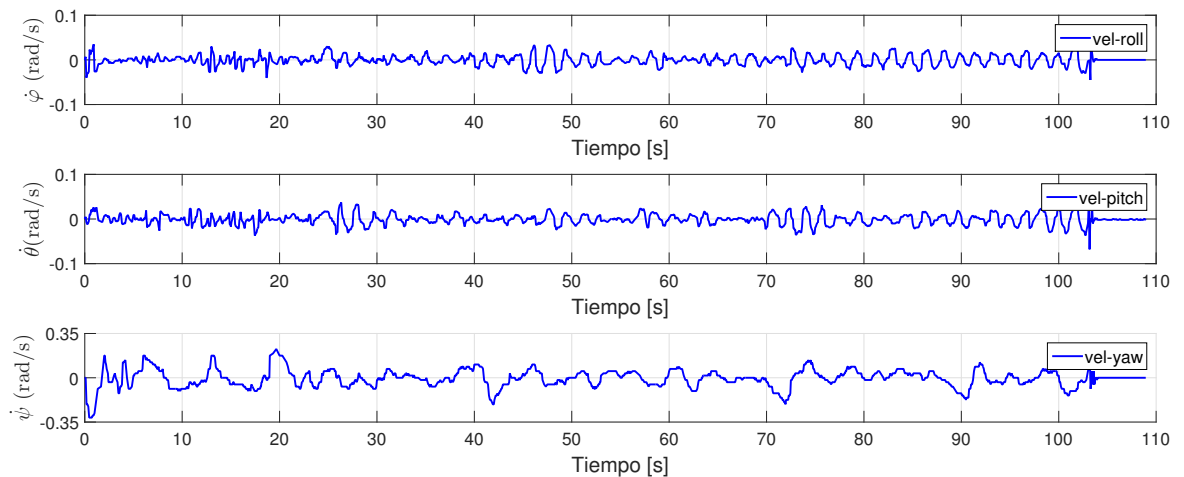


Figura 4.4: Señales de velocidad angular aplicando control óptimo de horizonte infinito para vuelo estacionario.

La gráfica siguiente 4.5 muestra la vista en  $2D$  para las posiciones en los ejes  $x$  y  $y$ . Esta vista nos permite corroborar la posición actual del vehículo en un plano cartesiano, se puede observar que el vehículo se encontraba entre  $-0.6\text{m}$  y  $0.6\text{m}$  en el eje  $x$  y entre  $-1\text{m}$  y  $1.6\text{m}$  para el eje  $y$ , lo cual comparado con las mediciones obtenidas de un GPS, son más precisas.

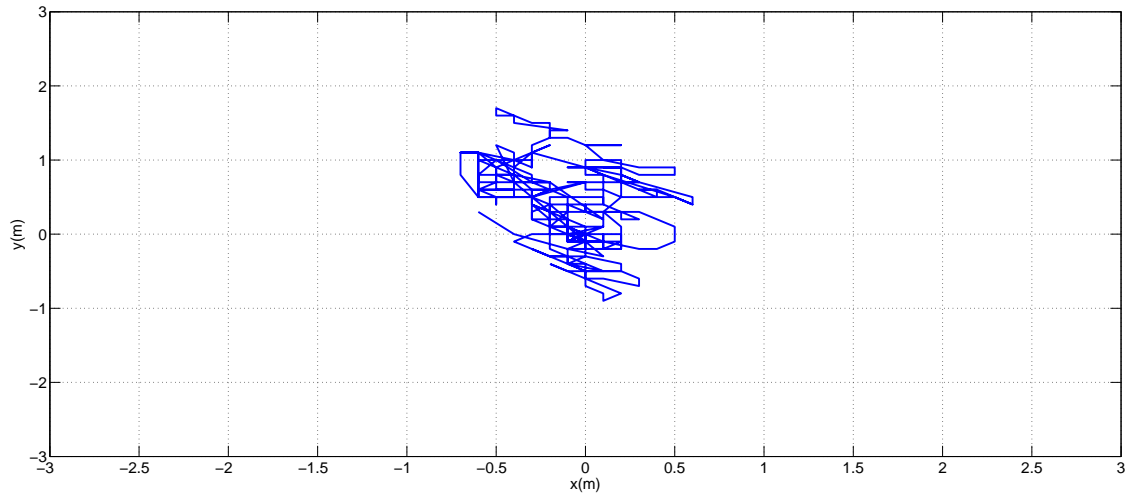


Figura 4.5: Posición en 2D aplicando control óptimo de horizonte infinito para vuelo estacionario.

Las señales de control se muestran en la figura 4.6:

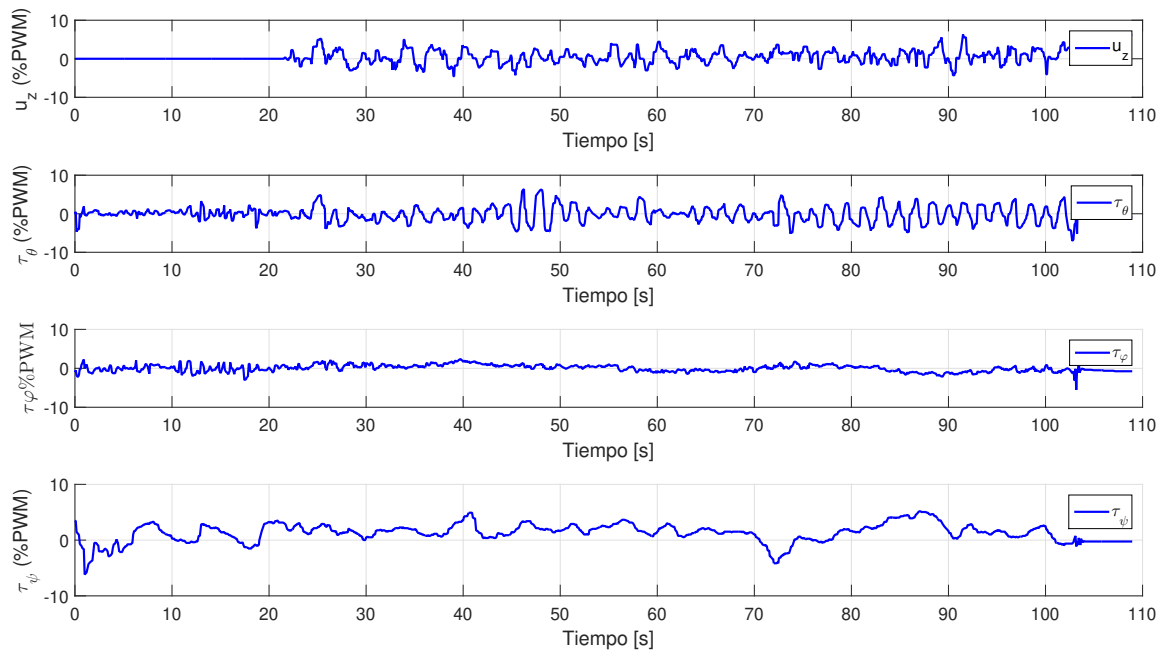


Figura 4.6: Señales de control óptimo de horizonte infinito para vuelo estacionario.



Las señales de control anteriores están interpretados en un porcentaje de *PWM*, donde el mínimo valor es 1 y el máximo valor que podemos enviar es 792, en los controles  $\tau_\varphi$ ,  $\tau_\theta$  y  $\tau_\psi$ , las señales tienen variaciones en el lado negativo sin embargo esto únicamente nos indica en que sentido se debe hacer la corrección.

### 4.2.2 Prueba 2: Seguimiento de trayectoria

Para este enfoque también se realizaron experimentos de seguimiento de una trayectoria. Se eligió una media circunferencia ya que se quería observar el comportamiento de los controles cuando el vehículo debía moverse a lo largo de los dos ejes cartesianos  $x$  y  $y$ . Esta prueba se realizó durante la tesis de maestría [14], el experimento se realizó utilizando mediciones del sistema de visión Optitrack Flex v3.

La siguiente gráfica 4.7 muestra el resultado de aplicar el control óptimo discreto de horizonte infinito, este experimento se incluyó en el artículo [29].

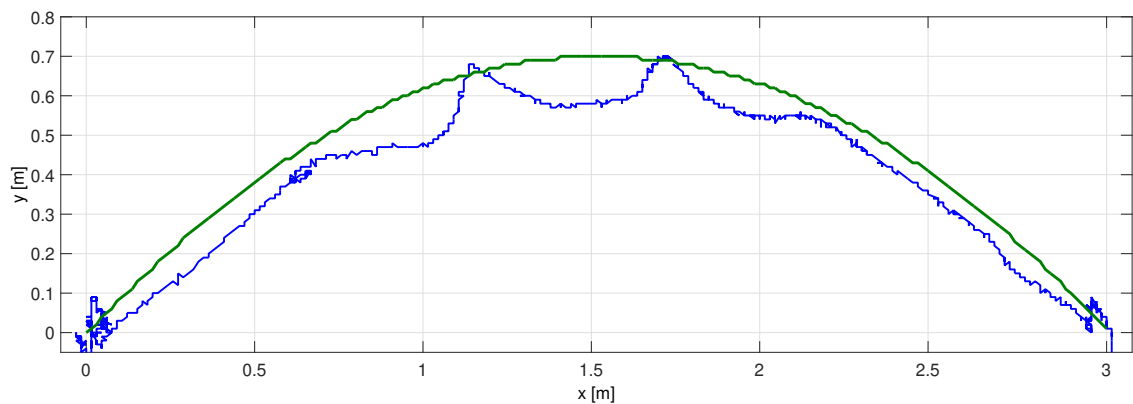


Figura 4.7: Seguimiento de una trayectoria semi-circular con un control óptimo discreto infinito.

La trayectoria anterior se realizó en interiores, y con una plataforma basada en el vehículo *Parrot*, cabe denotar que el UAV se encontraba energizado con una fuente en tierra, por lo que el cable de alimentación representó una perturbación externa. Se realizó un análisis

de consumo energético, estos resultados se compararon con un controlador Proporcional Derivativo. Los resultados indicaron un ahorro del 8.9% por parte del control óptimo infinito, [29].

## 4.3 Control óptimo de horizonte finito, caso discreto

### 4.3.1 Prueba 1: Seguimiento de trayectorias en exteriores

Este experimento se centra en el control de las dinámicas  $x - \theta$  y  $y - \varphi$ , sin embargo también la altura se estaba controlando haciendo uso de las matrices de penalización 4.3. Este experimento se llevó a cabo con la plataforma descrita en el Capítulo 3, al igual que la prueba anterior, se utilizó el sensor de flujo óptico Pxf4flow para estimar las posiciones  $x$  y  $y$ , la altura se obtuvo con el sensor barométrico MS5611 el cual viene incluido en el autopiloto Pixhawk.

$$\begin{aligned} Q_{z,\psi} &= \text{diag}\{9 \times 10^4, 6 \times 10^4, 1.6 \times 10^6, 1 \times 10^6\}, \\ R_{z\psi} &= \text{diag}\{32, 28\}, \\ Hz, \psi &= \text{diag}\{100, 150, 100, 150\}. \end{aligned} \tag{4.3}$$

Las matrices de penalización utilizadas para las dinámicas de  $x - \theta$  y  $y - \varphi$  son las siguientes:

$$\begin{aligned} Q_{y-\varphi} = Q_{x-\theta} &= \text{diag}\{1 \times 10^4, 9 \times 10^4, 1 \times 10^6, 1.8 \times 10^5\}, \\ R_\varphi = R_\theta = 20, \quad H_\varphi = H_\theta &= \text{diag}\{70, 32, 30, 30\}. \end{aligned} \tag{4.4}$$

La tarea que debe realizar el vehículo es: inicialmente debe despegar hasta llegar a la altura deseada, éste se fijó previamente en  $1.5m$ , después debe iniciar una trayectoria circular con un radio de 1 metro. Todo esto utilizando un control óptimo discreto de horizonte finito con una linealización exacta. Las posiciones angulares deseadas, las velocidades traslacionales

$\dot{x}$ ,  $\dot{y}$  y las velocidades angulares deseadas se fijaron en cero durante el despegue y aterrizaje. El horizonte para este control se fijó en 60s, este horizonte puede moverse de acuerdo a la velocidad de convergencia deseado, sin embargo para estas pruebas se buscaba un ahorro en el consumo por lo que se prefirió una baja velocidad. El tiempo de muestreo para todas las pruebas fue de  $T = 0.01s$

La primera gráfica 4.8 nos muestra la señal correspondiente al eje  $x$  con respecto del eje  $y$ , donde pude observarse la trayectoria circular deseada que se propuso previamente con un radio de 1m y el seguimiento en tiempo real, esta gráfica también involucra las fases de despegue y aterrizaje.

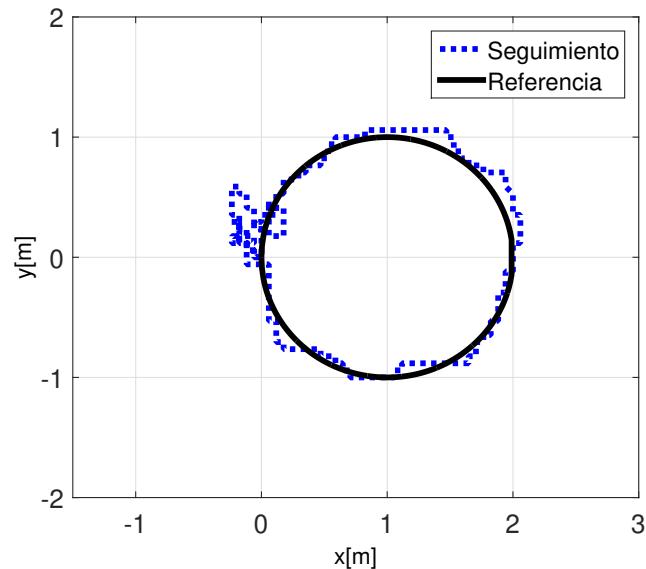


Figura 4.8: Gráfico  $x$  vs  $y$  de la trayectoria recorrida por el vehículo y la referencia.

Al inicio se pueden ver algunos errores en la gráfica 2D esto se debe a que el sensor Px4 flow, presentaba algunos errores cuando se encontraba a menos de 20cm. A continuación se muestra la gráfica 4.9 de los ejes traslacionales  $x$ ,  $y$  y  $z$  respecto del tiempo, el cual muestra el comportamiento del vehículo durante el tiempo que dura la misión.

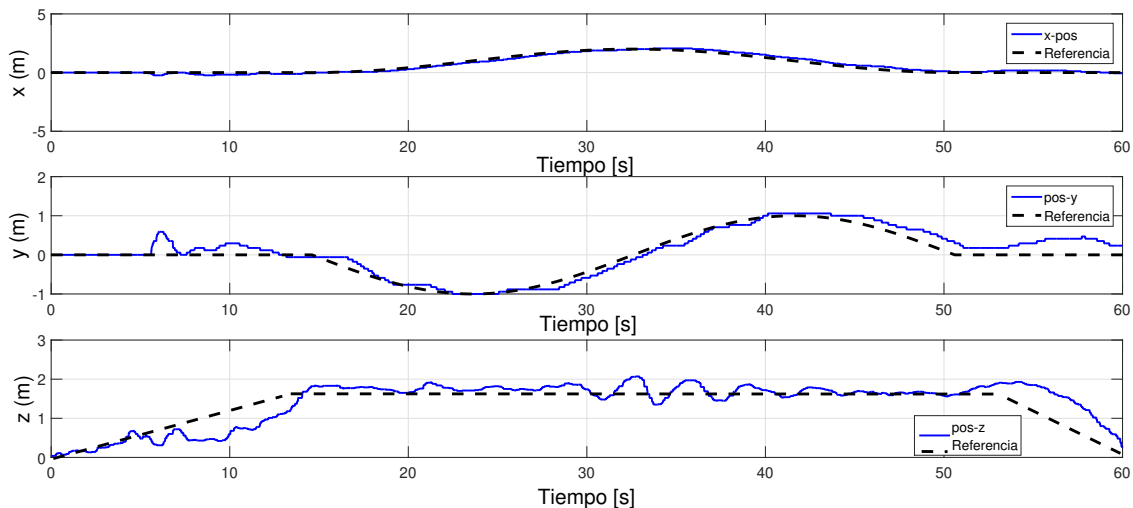


Figura 4.9: Gráfico de las señales  $x$ ,  $y$ ,  $z$  durante la trayectoria, aplicando el control óptimo lineal discreto de horizonte finito.

La figura 4.10 muestra las señales de los ángulos durante el seguimiento de la trayectoria. El ángulo deseado en yaw es 0 sin embargo debido a defectos mecánicos como se mencionó anteriormente, la planta se mantenía en otro valor, en otras pruebas este problema se corrigió hasta mantener un valor cercano a la referencia.

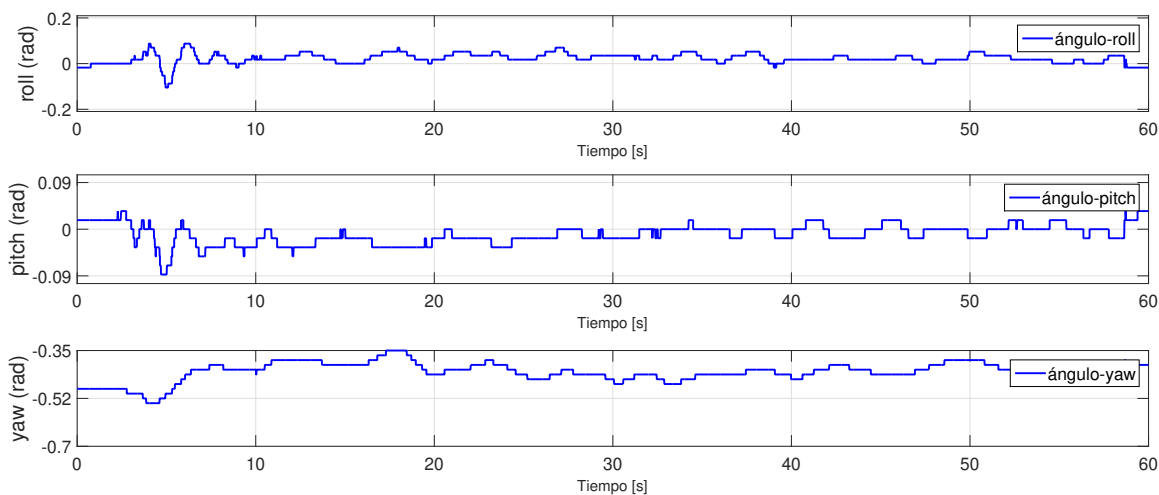


Figura 4.10: Gráfico de las posiciones angulares  $roll$ ,  $pitch$  y  $yaw$ .

También en la siguiente gráfica se muestran las velocidades traslacionales,  $x$ ,  $y$  y  $z$ .

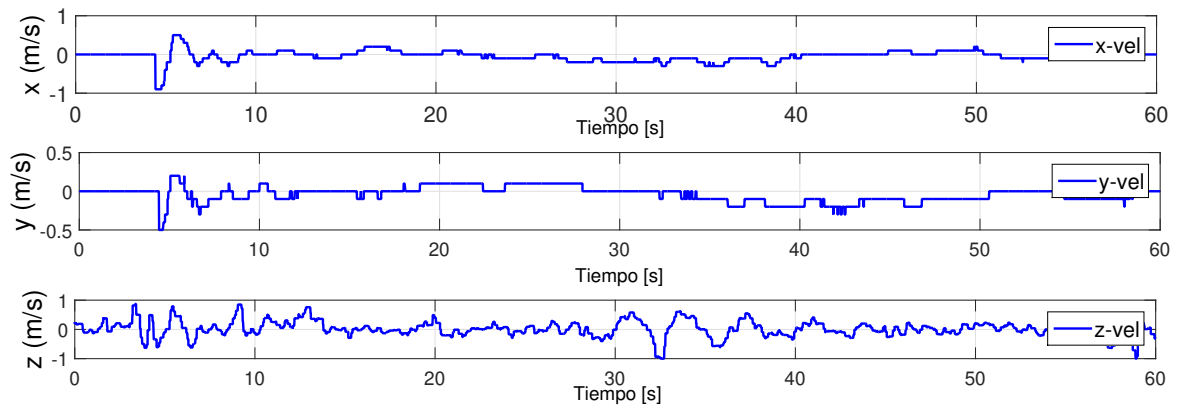


Figura 4.11: Gráfico de las velocidades traslacionales durante la trayectoria.

Finalmente se muestran las gráficas de las señales de control óptimas discretas. Estos resultados se documentaron en el artículo [28] el cual se encuentra bajo revisión.

Es importante resaltar que los resultados de estas pruebas se recopilaban en tiempo real a una transmisión menor a 10hz, sin embargo internamente todas las señales de control control se calcularon a 100 Hz.

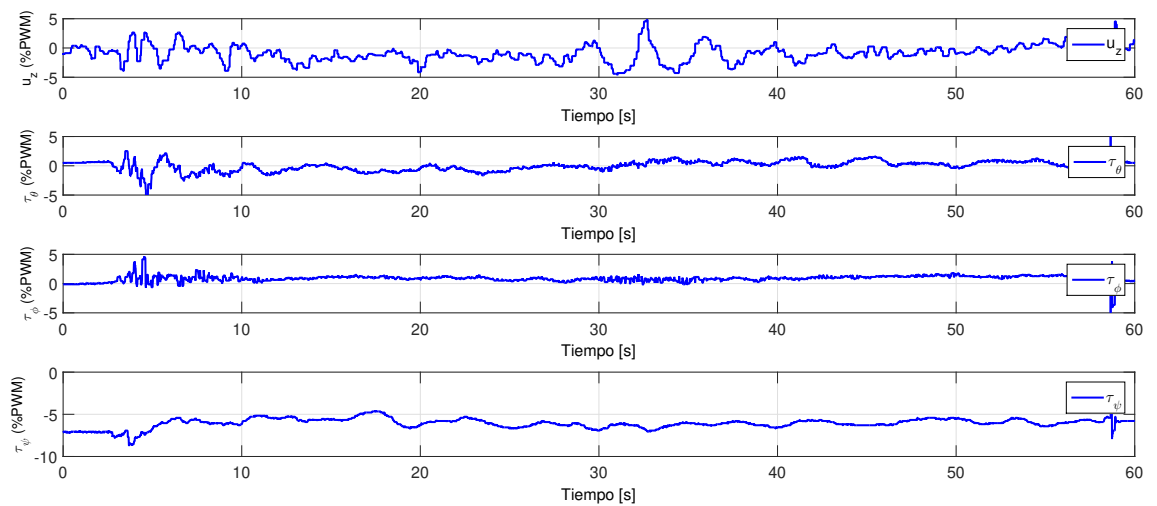


Figura 4.12: Señales de control óptimas discretas de horizonte finito.

Se realizaron 10 pruebas experimentales en el exterior con el objetivo de analizar el consumo energético y del desempeño. Estos resultados se muestran en la siguiente subsección.

## 4.4 Control subóptimo no lineal discreto de horizonte finito

### 4.4.1 Prueba 1: Seguimiento de una trayectoria

La misión de este experimento es seguir una trayectoria deseada circular en un ambiente no controlado bajo diferentes perturbaciones externas, principalmente el viento. Las mediciones de las posiciones para  $x$  y  $y$  fueron obtenidas con el sensor *Px4 Flow*, la altura con el barómetro. El tiempo de muestreo utilizado es  $T = 0.01s$ , el horizonte fue elegido de  $56s$ . En este experimento se utilizaron las siguientes matrices de penalización 4.5 para el control subóptimo discreto no lineal de horizonte finito:

$$H_0 = \begin{bmatrix} 4,200 & 3,800 & 0 & 0 \\ 3,800 & 4,300 & 0 & 0 \\ 0 & 0 & 17,500 & 16,000 \\ 0 & 0 & 16,000 & 17,500 \end{bmatrix}, \quad Q_{z,\psi} = \begin{bmatrix} 4,000 & 3,900 & 0 & 0 \\ 3,900 & 4,000 & 0 & 0 \\ 0 & 0 & 17,500 & 17,000 \\ 0 & 0 & 17,000 & 17,500 \end{bmatrix},$$

$$R_{z,\psi} = \text{diag}\{0.47, 0.5\},$$

$$Q_{x,\theta} = Q_{y,\varphi} = \begin{bmatrix} 8 \times 10^5 & 0 & 0 & 129 \\ 0 & 8 \times 10^5 & 0 & 8,000 \\ 0 & 0 & 8,500 & 8,450 \\ 129 & 8,000 & 8,450 & 8,500 \end{bmatrix},$$

$$H_{0,x} = H_{0,y} = \begin{bmatrix} 8 \times 10^5 & 0 & 0 & 129 \\ 0 & 8 \times 10^5 & 0 & 5,000 \\ 0 & 0 & 8,500 & 8,450 \\ 129 & 5,000 & 8,450 & 8,500 \end{bmatrix},$$

$$R_\varphi = R_\theta = 0.45.$$

(4.5)

La figura 4.13 muestra el resultado en 2D del seguimiento de la trayectoria con este enfoque subóptimo de control. Observe que en algunas partes de la señal parece que hay una discontinuidad, lo cual se debe a que el tiempo de muestreo al adquirir los datos era muy grande.

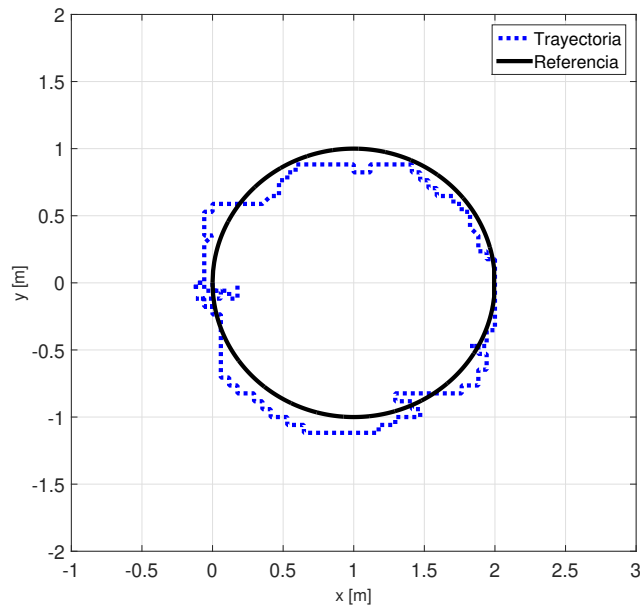


Figura 4.13: Gráfico de la trayectoria realizada por el vehículo en dos dimensiones.

En las siguientes gráficas 4.14, 4.15 se pueden apreciar las posiciones  $x$ ,  $y$  y  $z$  4.14 con respecto del tiempo y las señales de posición angular 4.15 respectivamente. En la señal de altura, se observan algunas oscilaciones, aunque éstas parecen grandes realmente no lo son, esto se debió a que la altura se obtenía únicamente de la señal del barómetro. En los experimentos posteriores este fenómeno desaparece al utilizar la medición que resulta de la fusión del acelerómetro y el barómetro.



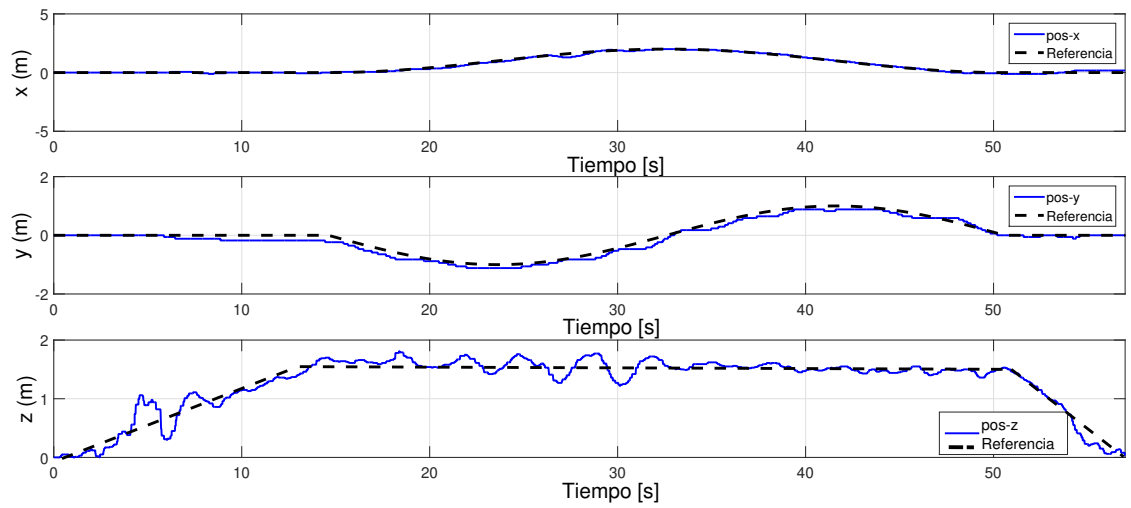


Figura 4.14: Señales de las posiciones traslacionales  $x$ ,  $y$  y  $z$  respecto del tiempo.

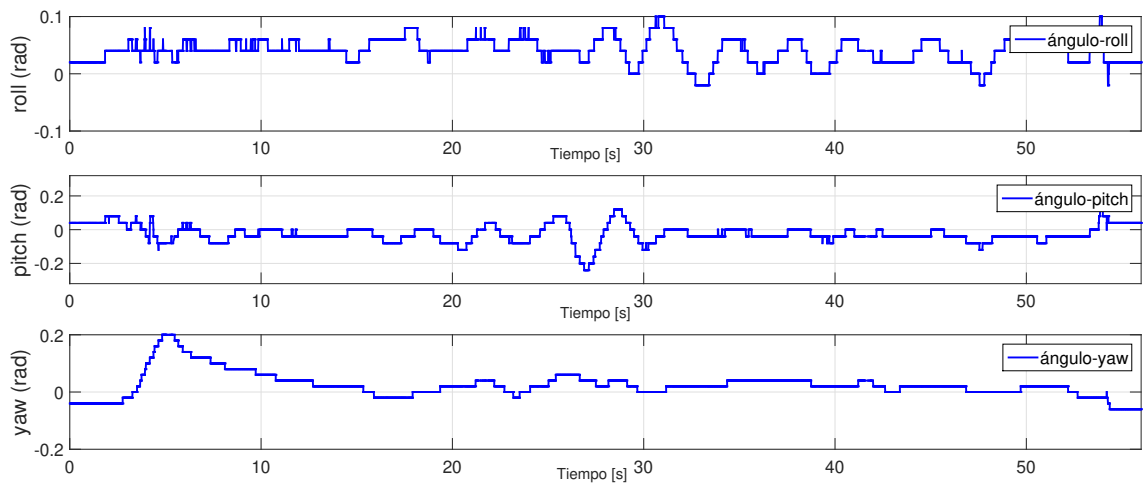


Figura 4.15: Gráfico de los ángulos  $roll$ ,  $pitch$  y  $yaw$  durante el seguimiento de la trayectoria.

En esta gráfica se muestra las señales independientes de velocidad para cada eje.

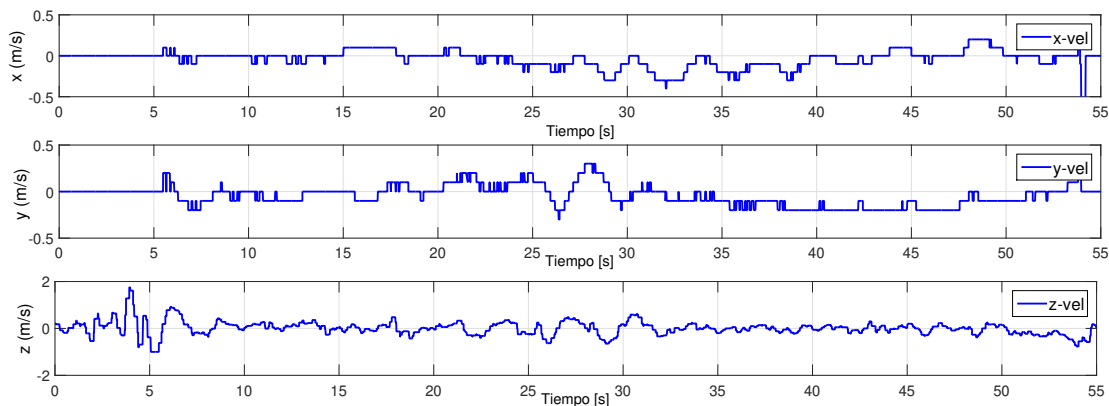


Figura 4.16: Señales de velocidad  $\dot{x}$ ,  $\dot{y}$  y  $\dot{z}$  durante el seguimiento de la trayectoria.

Esta última gráfica 4.17 muestra las señales de control aplicadas al cuadricóptero, las cuales cabe resaltar que fueron obtenidas en tiempo real y se calcularon a bordo del autopiloto.

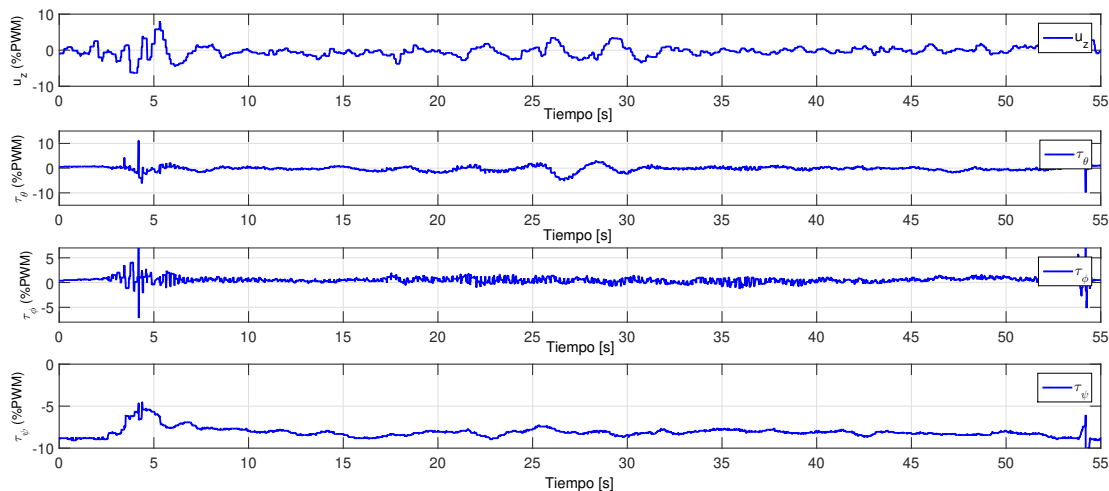


Figura 4.17: Señales de control sub-óptimas discretas no lineales (NLSOC):  $\tau_\theta$ ,  $\tau_\phi$ ,  $\tau_\psi$ ,  $u_z$ .

Se realizaron 10 pruebas con el objetivo de verificar la repetitividad entre cada experimento. La tabla 4.1 muestra los errores promedio y la desviación estándar de todas las pruebas los cuales se comparan con el control óptimo lineal de horizonte finito(LOC) y controlador PD el cual se sintonizó de manera experimental. La tabla muestra el análisis

del consumo energético y el desempeño, donde el índice de desempeño  $J$  está dado como  $J_{control} = \sum(abs(Control(k)))$ .

	<i>NLSOC</i>	<i>LOC</i>	<i>PD</i>
Error promedio	0.19 m	0.23 m	0.3 m
Desviación estándar	0.05	0.09	0.14

Tabla 4.1: Comparación de los errores promedio y la desviación estándar, basado en los experimentos de seguimiento de trayectoria en exteriores.

Índice de desempeño	<i>NLSOC</i>	<i>LOC</i>	<i>PD</i>
$J_{u,\tau_\varphi,\tau_\psi,\tau_\theta}$	5,415.2	5,374.5	6,072.6
$J_e$	22,811.6	27,233.01	35,575.6

Tabla 4.2: Comparación del desempeño y el ahorro energético basado en resultados experimentales en el exterior.

La tabla anterior 4.2 permite concluir que el controlador propuesto es una opción viable para aplicarse en este tipo de vehículos, el consumo energético del controlador subóptimo no-lineal (NLSOC) es similar al control óptimo lineal (LOC), pero el subóptimo no lineal mejora en el desempeño, véase la tabla 4.2, donde  $J_e = \sum(abs(error))$ . La tabla 4.1 muestra que los errores promedio fueron menores para el control subóptimo no-lineal comparados con el controlador óptimo lineal y el control Proporcional Derivativo, la desviación estándar indica que los errores no se alejan del promedio, es decir que hay repetitividad en las 10 pruebas realizadas. Estos resultados se presentaron en [28] el cual se encuentra sometido a una revista.

## 4.5 Algoritmo de control Super-twisting combinado con el control subóptimo no lineal

### 4.5.1 Prueba 1: Estimación de perturbaciones externas y seguimiento de trayectoria

Este experimento se realizó en el exterior en presencia de perturbaciones externas, las cuales este método trata de estimar con el algoritmo de mínimos cuadrados. En este caso se asume que la perturbación más grande medible al cual se enfrenta el vehículo es el viento.

La tarea que el vehículo debe realizar se divide en 3 fases: el primero consiste en llegar a una altura deseada el cual se fijó en 2 metros, durante esta fase sólo se aplica el control subóptimo, el cual ya se presentó anteriormente. En la segunda fase el cuadricóptero debe seguir una trayectoria circular descritas por las ecuaciones 4.6 y 4.7, durante esta fase se implementa el algoritmo de mínimos cuadrados para estimar la perturbación externa y con base a dicha estimación se decide entre aplicar el control subóptimo no lineal o el control basado en el algoritmo "super-twisting". La última fase es el aterrizaje, nuevamente aquí sólo se aplica el control subóptimo.

#### Parámetros utilizados para este experimento

Posiciones, velocidades y aceleraciones deseadas:

$$\begin{aligned}
 x &= -4 * \cos((\pi/180) * t) + 4, & (4.6) \\
 \dot{x} &= 0.069 * \sin((\pi/180) * t), \\
 \ddot{x} &= 0.0012 * \cos((\pi/180) * t)
 \end{aligned}$$

$$\begin{aligned}
 y &= -4 * \sin((\pi/180) * t), \\
 \dot{y} &= -0.069 * \cos((\pi/180) * t), \\
 \ddot{y} &= 0.0012 * \sin((\pi/180) * t).
 \end{aligned}
 \tag{4.7}$$

donde  $t = [t_1, t_2]$  que corresponde a la fase 2. Las matrices de penalización utilizadas para el control Subóptimo no lineal son las mismas que en 4.5.

#### Parámetros para el control Super-twisting

Tabla 4.3: Ganancias para el control super-twisting durante el seguimiento de la trayectoria.

Parámetros	Valor
Masa del vehículo, (m) [kg]	1
Constante positiva, ( $k_1$ )	27
Constante positiva, ( $k_2$ )	50
Constante Positiva, ( $k_3$ )	2.8
Ganancia 1, $g_1$	$1.5 * \sqrt{L}$
Ganancia 2, $g_2$	$1.1 * L$
Tiempo de muestreo, ( $T$ ) [s]	0.01

En esta figura 4.18 se muestra el esquema general que se siguió para poder implementar la estructura de control propuesta.

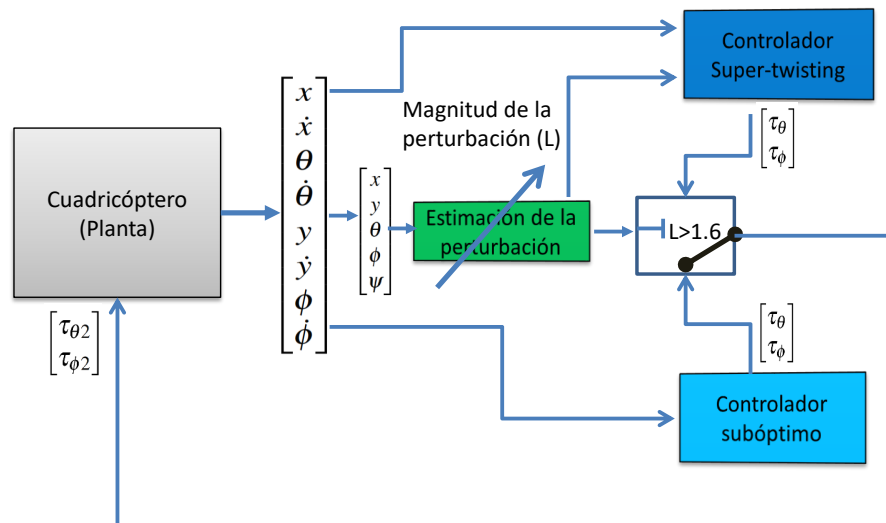


Figura 4.18: Descripción general de la implementación del control.

La figura anterior 4.18 nos muestra que se realiza una estimación de las perturbaciones externas y en base a ella se decide que control implementar.

Este experimento se centra principalmente en la fase de seguimiento, pues, es en esta fase donde se implementa la estructura de control “Óptimo-robusto” y la estimación de perturbaciones. La siguiente gráfica, muestra la posición en  $2D$  de la trayectoria que el vehículo realizó.

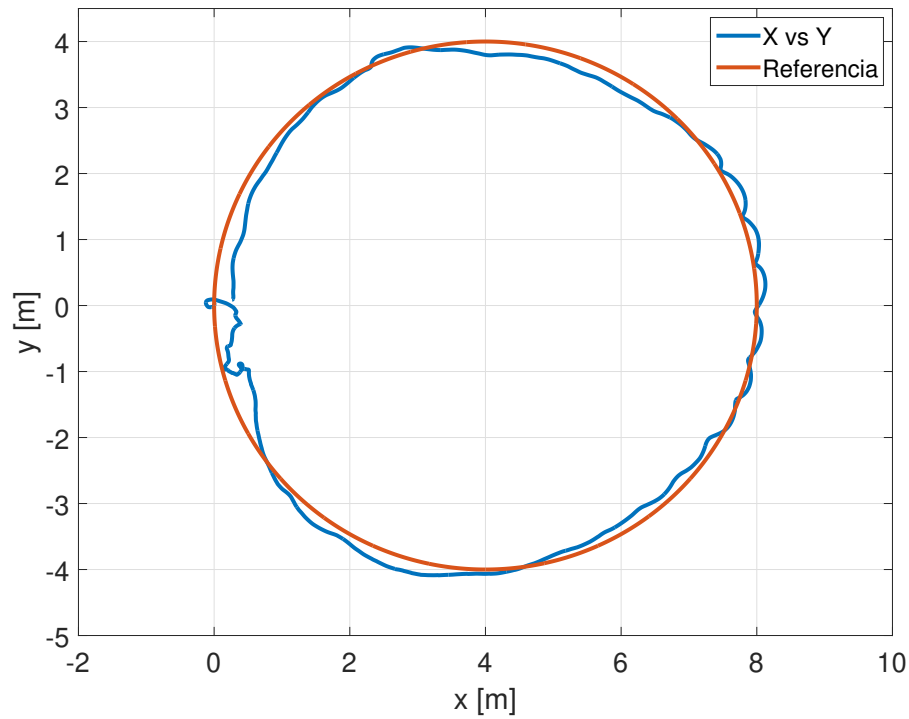


Figura 4.19: Gráfico  $x$  vs  $y$  de la trayectoria seguida y la referencia.

Las siguientes figuras muestran las posiciones respecto del tiempo, en los ejes  $x$  y  $y$ .

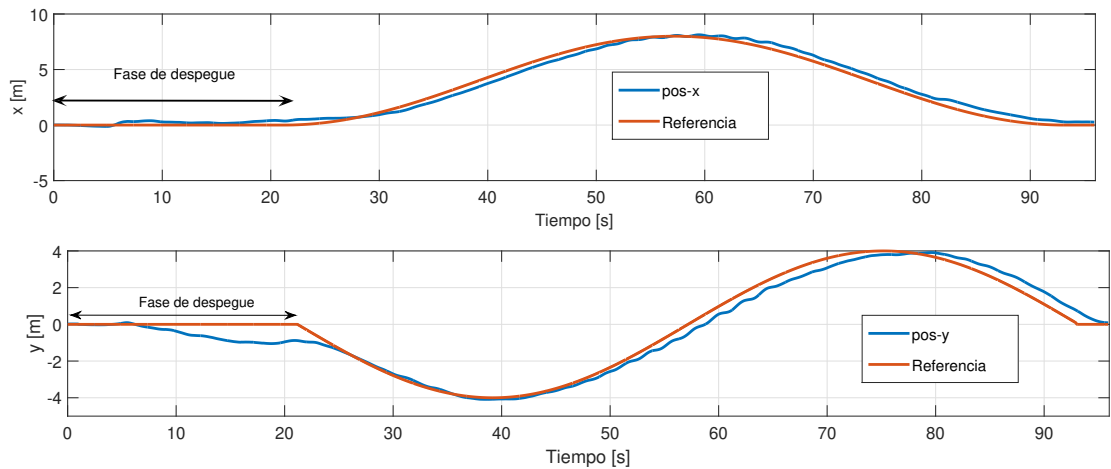


Figura 4.20: Gráficos  $x$  y  $y$  de la trayectoria durante el seguimiento, respecto del tiempo.

En este experimento se propusieron velocidades deseadas a diferencia de los demás al-

goritmos de control propuestos, por ello, se agregan las siguientes gráficas de las señales de velocidad junto con su referencia.

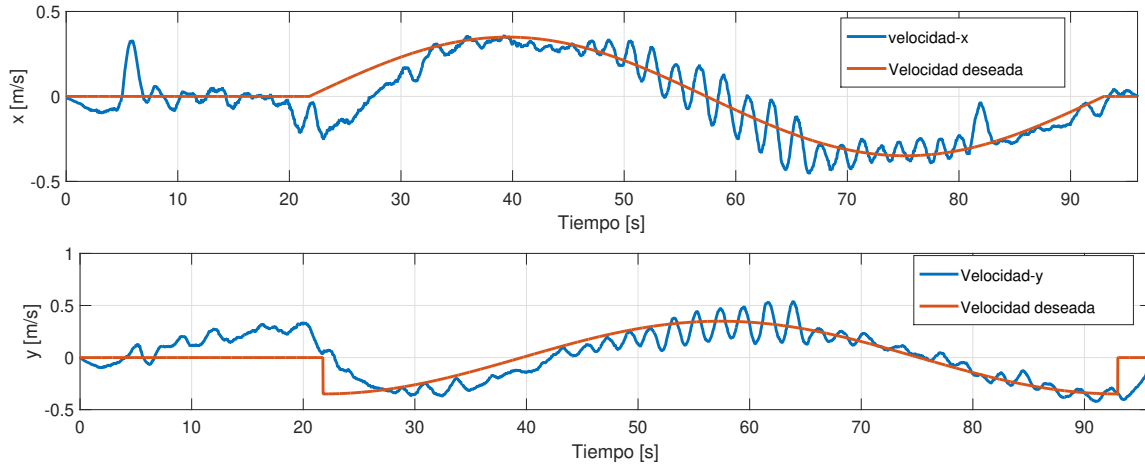


Figura 4.21: Gráficos  $\dot{x}$  y  $\dot{y}$  de la trayectoria durante el seguimiento, respecto del tiempo.

Las gráficas de los errores en ambos ejes cartesianos,  $x$  y  $y$  también se agregaron con el objetivo de poder verificar la diferencia entre la referencia y la posición actual del vehículo, respecto del tiempo.

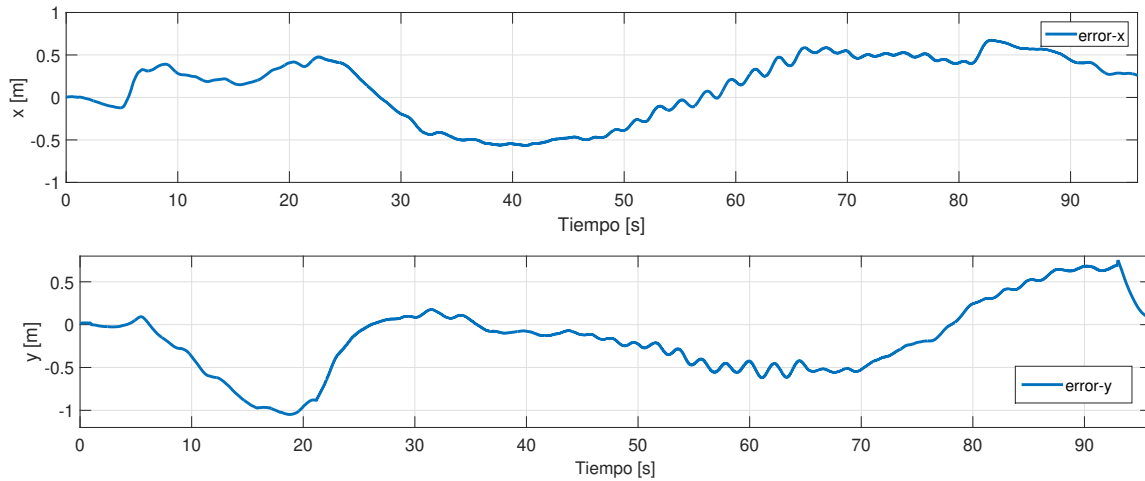


Figura 4.22: Gráficos de los errores de seguimiento durante la trayectoria en el tiempo.

La decisión de utilizar uno u otro control, se basó en la estimación de la perturbación externa con el algoritmo de mínimos cuadrados. Para ello se utilizó el siguiente criterio:



Si  $L =$  Estimación de la perturbación  $\geq 1.6N$  se aplica el algoritmo super-twisting, de lo contrario si la *Perturbación*  $< 1.6N$ , entonces se aplica el control subóptimo. Este criterio se obtuvo con base a varias pruebas que se analizaron.

Las siguientes figuras muestran las estimaciones de las posibles perturbaciones externas en cada eje respecto del tiempo.

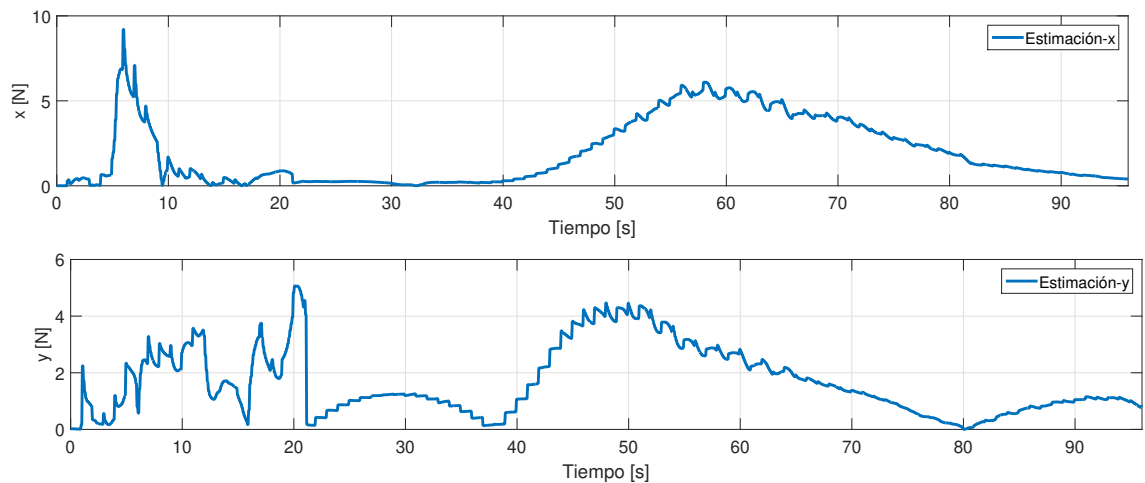


Figura 4.23: Gráficos de las estimaciones de perturbaciones externas en los ejes  $x$  y  $y$  respectivamente.

Las gráficas anteriores junto con el criterio anterior son las que definen la señal de control que se aplicará. Las señales resultantes se muestran en las siguientes gráficas. Se ha añadido también una señal, llamada “switch de control” el cual únicamente sirve para indicar visualmente que señal de control se está aplicando en determinado momento.

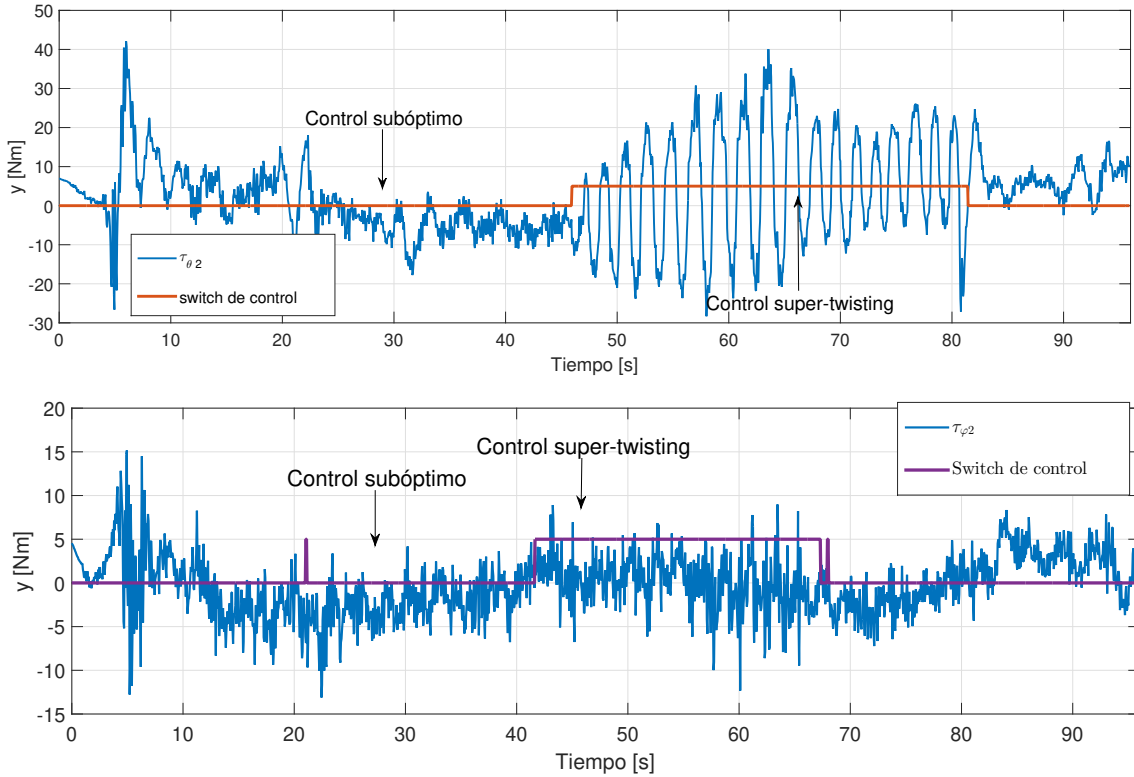


Figura 4.24: Señales de control  $\tau_{\theta 2}$  y  $\tau_{\varphi 2}$  respectivamente, que involucran al control subóptimo y el control super-twisting.

El objetivo de esta nueva estructura de control es demostrar que se puede obtener un ahorro en el consumo energético conservando un desempeño aceptable, es decir, aprovechar las cualidades del control subóptimo el cual nos ayuda a ahorrar energía y el control robusto que nos permite tener un mejor desempeño. Para ello se realizaron varias pruebas individuales, únicamente utilizando el control no lineal subóptimo (NLSOC) o el control basado en el algoritmo del Super-twisting (STA), estos resultados se muestran en la siguiente tabla 4.4. EL indicador de desempeño utilizado fue el IAE (Integral Absoluta del Error). También se analizaron las señales de control utilizando la siguiente función de costo  $J = \sum_{t_1}^{t_2} T(abs(Controlador(k)))$ .

<i>Indicador</i>	<i>NLSOC</i>	<i>STA</i>	<i>NLSOC + STA</i>
IAE $x$	862.40	303.72	578.51
IAE $y$	748.61	298.07	432.06
$J_{\tau\theta\tau\varphi}$	3,986.83	4,730.83	4,065.83

Tabla 4.4: Comparación del IAE y el consumo energético durante la trayectoria.

En la tabla anterior 4.4 podemos notar que el objetivo se cumplió, ya que en los indicadores, se puede ver que utilizando únicamente el control super-twisting, el error es menor respecto del control subóptimo, pero el consumo energía es mayor. Si sólo se utiliza el control subóptimo el error es mayor pero el consumo es menor. Por lo tanto al utilizar la combinación de ambos se logra un punto intermedio, lo que significa que tendremos aún así un ahorro en el consumo energético conservando un desempeño aceptable. Estos resultados se documentaron en el artículo [15] el cual está en revisión.

## 4.6 Despegue y aterrizaje autónomo utilizando visión artificial

### 4.6.1 Prueba 1: Simulación del despegue y aterrizaje

En este experimento se hizo una simulación en Simulink del despegue y aterrizaje de un cuadricóptero utilizando el control subóptimo no lineal discreto en horizonte finito el cual se presenta en el capítulo 2. Las matrices utilizadas fueron las mismas 4.5.

La tarea que debe realizar el vehículo únicamente es elevarse a una altura deseada e inmediatamente descender. La referencia de ascenso se fijó mediante la siguiente función:  $z = \frac{1}{3} * t$  y la de descenso con  $z = (T_f - t) * 0.25$ . Donde  $t$  representa el tiempo actual y el horizonte  $T_f = 21s$ . A esta simulación se le agregó una perturbación utilizando una función

sinusoidal con 0.4 de amplitud y con una frecuencia de  $15\text{rad/s}$ .

La siguiente gráfica muestra el resultado del despegue y aterrizaje.

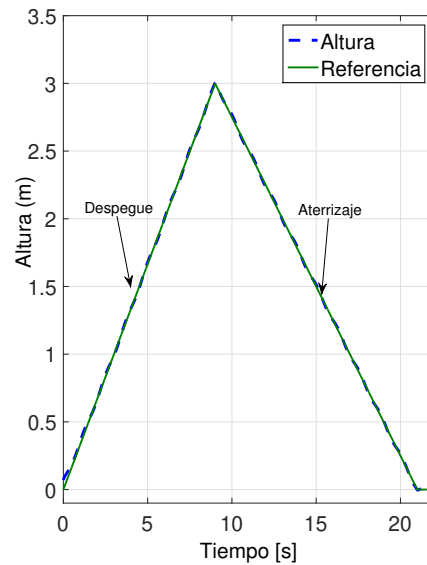


Figura 4.25: Simulación del despegue y aterrizaje mas una perturbación constante.

En la siguiente figura se muestra la señal de control para esta dinámica.

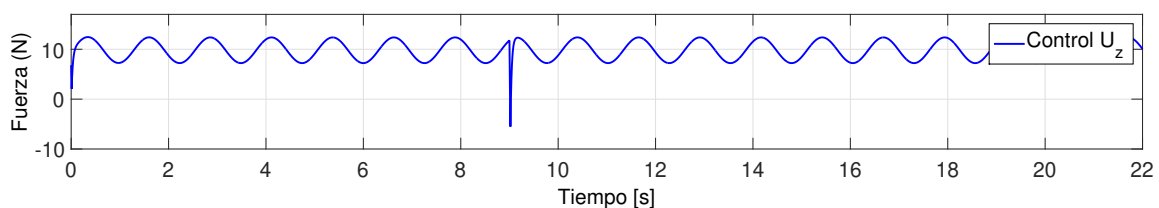


Figura 4.26: Señal de control  $u_z$  durante las fases de despegue y aterrizaje.

Esta simulación se realizó, con la finalidad de verificar que las funciones de despegue y aterrizaje funcionaran y que el control subóptimo no lineal discreto pudiera mantener al vehículo siguiendo la referencia a pesar de una pequeña perturbación. En la siguiente subsección se verán los resultados experimentales.

### 4.6.2 Prueba 2: Despegue y aterrizaje, experimental

Este experimento se realizó en tiempo real, en el exterior, bajo perturbaciones externas, los parámetros utilizados para la referencia, fueron las mismas que en la simulación.

El procedimiento de la corrección de la posición, se realizó mientras el vehículo descendía. Esto debido a que las mediciones de la cámara dependían en gran medida del tamaño del patrón que en este caso era la imagen de un heli-puerto en color rojo. De las pruebas realizadas se observó que las mediciones eran aceptables hasta en 2 metros de distancia. Las mediciones se promediaron para que se eliminara el efecto de la discontinuidad en las mediciones. Cabe destacar que esta discontinuidad aún así está acotada.

La siguiente gráfica muestra la rutina de despegue y aterrizaje, en exteriores utilizando el control NLSOC presentado en la subsección 4.4.

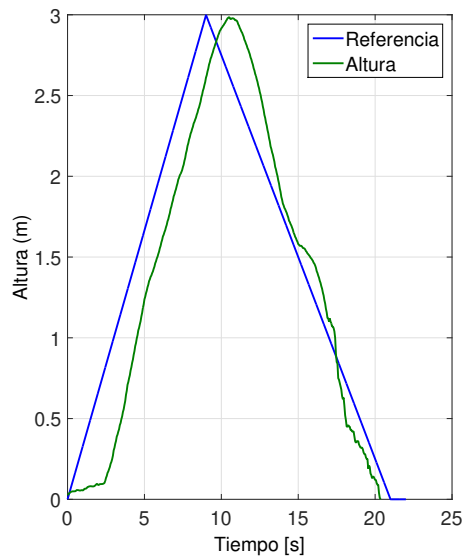


Figura 4.27: Señal de altura, el cual involucra las fases de despegue y aterrizaje.

La posición en 2D se muestra en la gráfica 4.28.

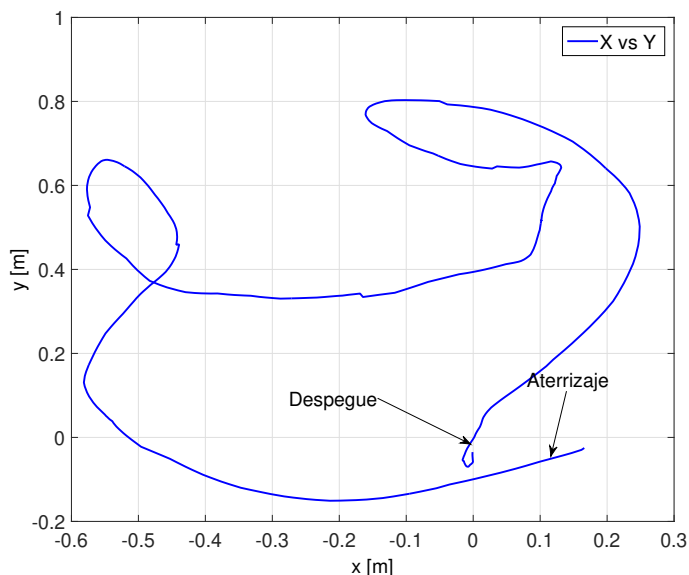


Figura 4.28: Señal  $x$  vs  $y$  durante el despegue y aterrizaje.

La siguiente gráfica muestra los ejes  $x$  y  $y$  respecto del tiempo, las posiciones deseadas  $x_{des}$  y  $y_{des}$  se fijaron en cero.

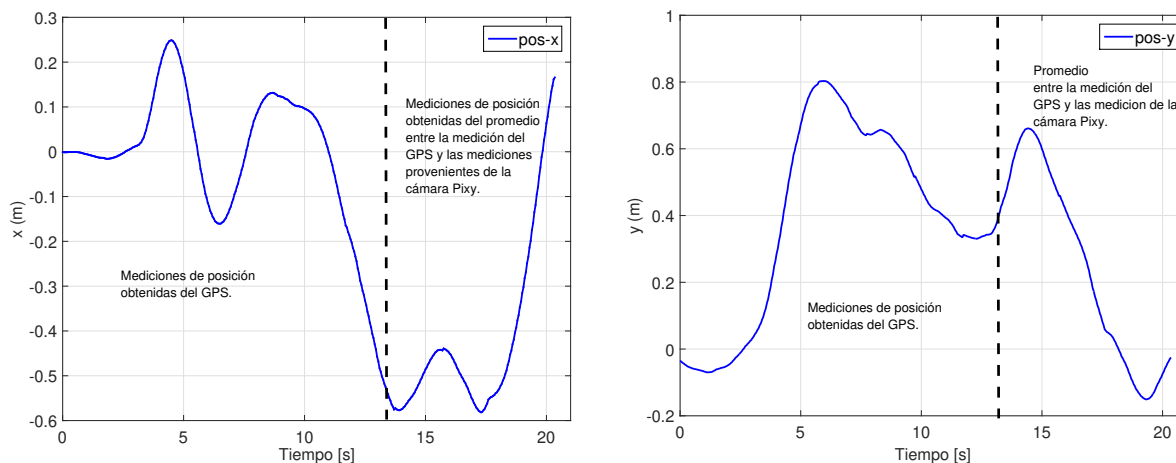


Figura 4.29: Señales  $x$  y  $y$  durante el despegue y aterrizaje respecto del tiempo.

En la figura anterior podemos observar el procedimiento de corrección, aproximadamente en  $t = 13$ s la posición empieza a corregirse en dirección al centro del patrón localizado en el

piso. Cabe resaltar también que el experimento se realizó en presencia de viento y aún así el control logró tener un buen desempeño.

Finalmente podemos observar en la gráfica 4.30 las señales de control subóptimas discretas no lineales que se aplicaron a las dinámicas correspondientes.

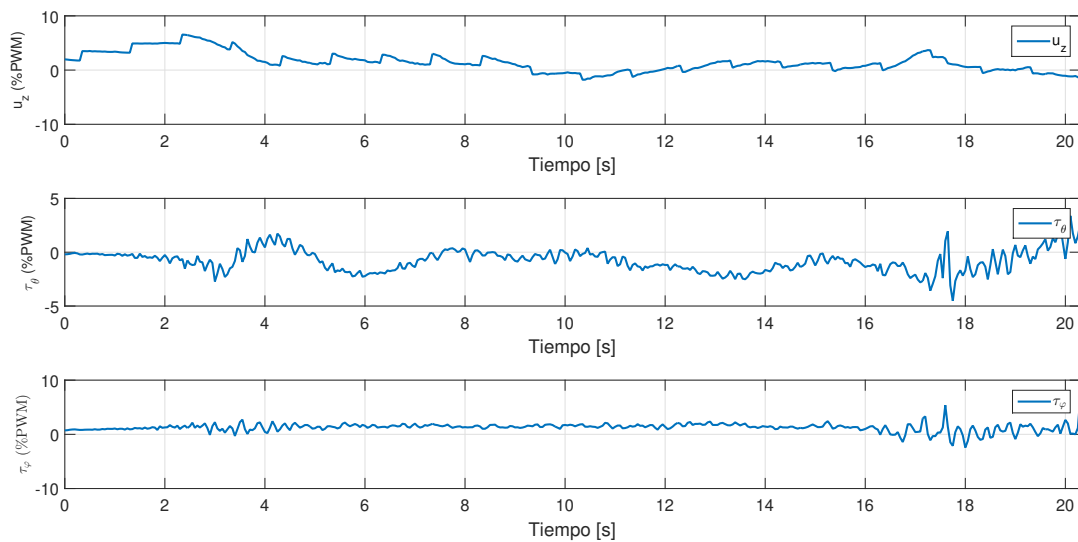


Figura 4.30: Señales de control, durante la misión aérea.

Para poder analizar si el desempeño era aceptable ó no, se realizaron 10 pruebas utilizando la corrección con visión artificial y 10 pruebas más únicamente con la señal del GPS. Los resultados se muestran en la tabla 4.5.

	<i>Con la corrección</i>	<i>Sin corrección</i>
Error promedio $x$	0.19	0.96
Desviación estándar	0.19	0.19
Error promedio $y$	0.361	0.48
Desviación estándar	0.27	0.34
Error promedio $z$	0.34	0.39
Desviación estándar	0.13	0.16

Tabla 4.5: Comparación de los errores promedio y de las desviaciones estándar durante las fases de despegue y aterrizaje con y sin la corrección de la posición.

La tabla 4.5 permite concluir que los errores promedio, son menores cuando utilizamos la corrección de posición con visión artificial comparando con sólo utilizar el GPS. Los errores promedio de la dinámica de altura son similares, lo cual es esperado ya que se utiliza el mismo controlador y los mismos sensores. Las desviaciones estándar son similares en ambos casos lo cual nos indica que existe repetitividad en los experimentos. Estos resultados fueron publicados en el congreso ICUAS 2017, [11].

## 4.7 Resultados experimentales utilizando el control no lineal basado en Backstepping

En esta sección se presentan los resultados experimentales de las misiones de despegue, seguimiento y aterrizaje autónomos, aplicando el algoritmo descrito en el Capítulo 2. Se realiza también un análisis estadístico para comparar la repetitividad de los resultados entre cada prueba.

La tarea que se debe realizar está dividida en 3 fases: la primera es el despegue, una vez que el vehículo alcanza una altura deseada comienza la segunda fase que es la de seguimiento



de una ruta circular, finalmente la etapa en la cual el vehículo desciende. Todas estas fases se realizan completamente autónomas. Para la implementación de este algoritmo de control se utilizó un tiempo de muestreo  $T = 0.01s$ .

Parámetros para la implementación del control Backstepping:

Tabla 4.6: Parámetros para la misión aérea utilizando el algoritmo de control BS.

<b>Parámetros</b>	<b>Valor</b>
Masa del vehículo, [ $kg$ ]	1.3
$I_x, I_y$	0.0023
$I_z$	0.0054
Tiempo de muestreo, ( $T$ ) [ $s$ ]	0.01
Subsistema Z	
Constante positiva, ( $\alpha_z$ )	2
Constante positiva, ( $k_z$ )	0.0053
Subsistema $\psi$	
Constante positiva, ( $\alpha_\psi$ )	1.34
Constante positiva, ( $k_\psi$ )	67.2
Subsistema $x-\theta$	
Constante positiva, ( $\alpha_{1x}$ )	1.57
Constante positiva, ( $k_{1x}$ )	0.152
Constante positiva, ( $\alpha_{2x}$ )	0.2
Constante positiva, ( $k_{2x}$ )	0.25
Constante positiva, ( $\alpha_{3x}$ )	0.008
Constante positiva, ( $k_{3x}$ )	57.7
Subsistema $y-\varphi$	
Constante positiva, ( $\alpha_{1y}$ )	6.31
Constante positiva, ( $k_{1y}$ )	0.001
Constante positiva, ( $\alpha_{2y}$ )	0.19
Constante positiva, ( $k_{2y}$ )	0.006
Constante positiva, ( $\alpha_{3y}$ )	0.001
Constante positiva, ( $k_{3y}$ )	35.3

La siguiente imagen muestra las fases de la tarea que el vehículo debe realizar así como las posiciones deseadas  $x_{des}$ ,  $y_{des}$  y  $z_{des}$

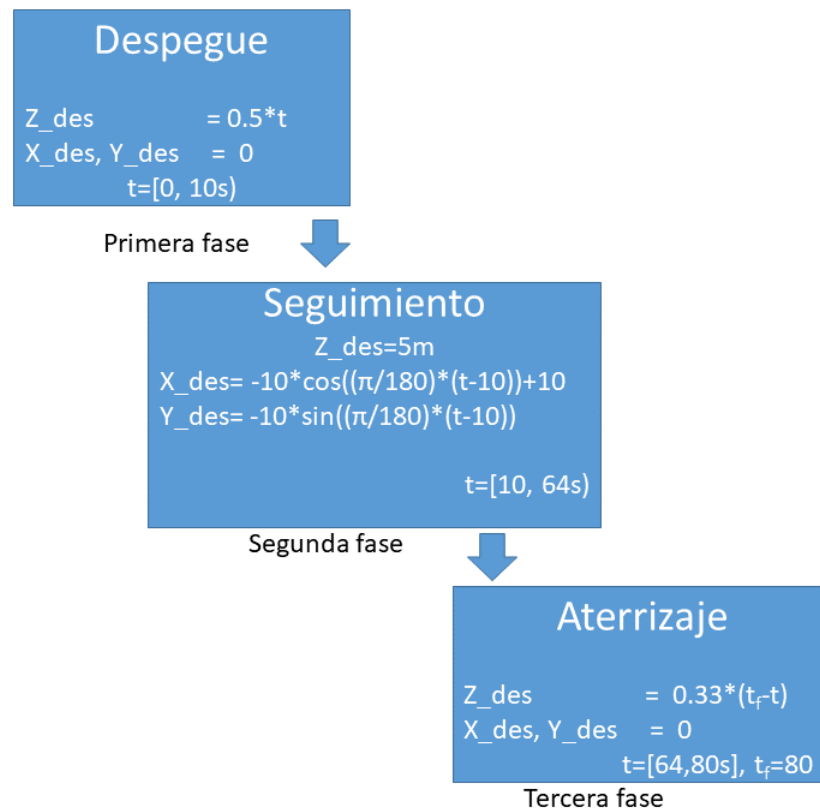


Figura 4.31: Fases de la misión aérea.

donde  $t \in [t1, t2]$ , el cual corresponde a la fase de seguimiento de la trayectoria. La siguiente gráfica muestra el comportamiento de los estados  $x$ ,  $y$  y  $z$  con respecto del tiempo.

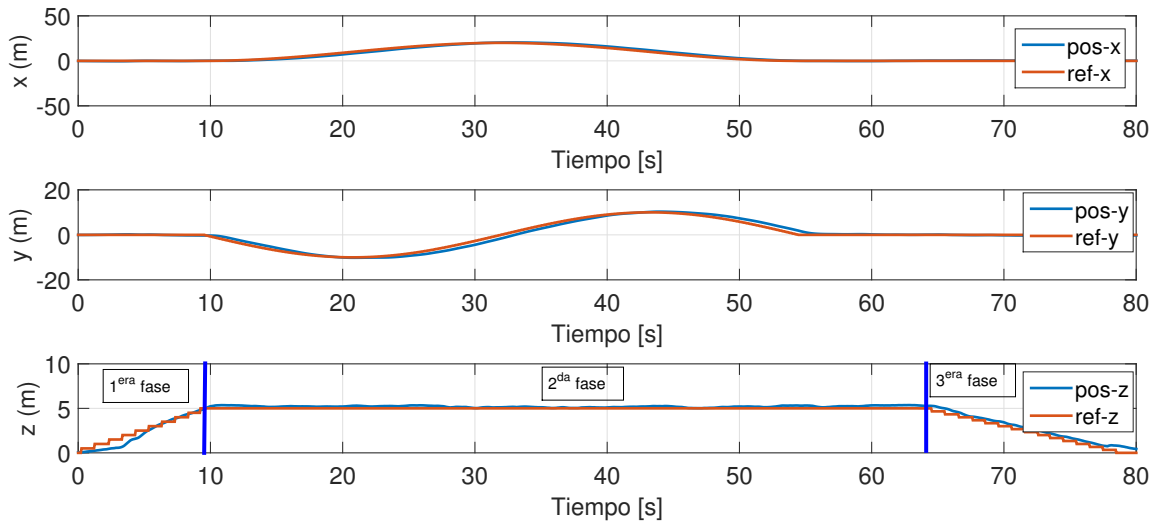


Figura 4.32: Posición en los ejes  $x$ ,  $y$  y  $z$  aplicando el control Backstepping modificado.

La trayectoria que el vehículo debe seguir es una circunferencia que inicia en el origen y tiene un diámetro de 20 metros. La gráfica 4.33 muestra la posición en 2D durante la fase de seguimiento.

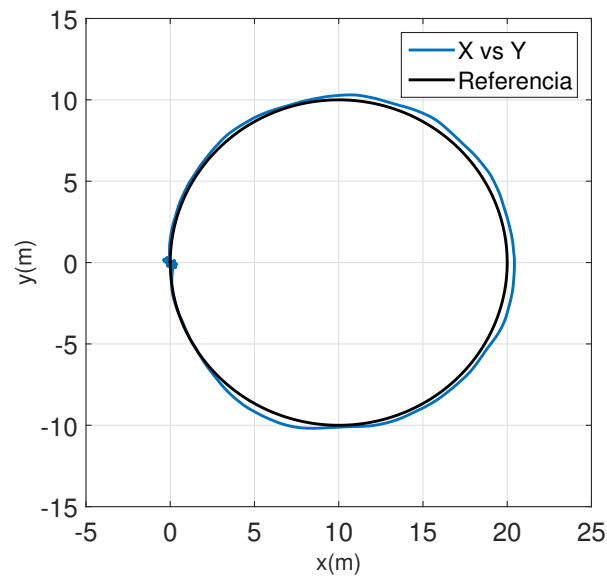


Figura 4.33: Posición  $x$  vs  $y$  durante la misión aérea, aplicando el control Backstepping.

La orientación del vehículo se muestra en la siguiente figura 4.34

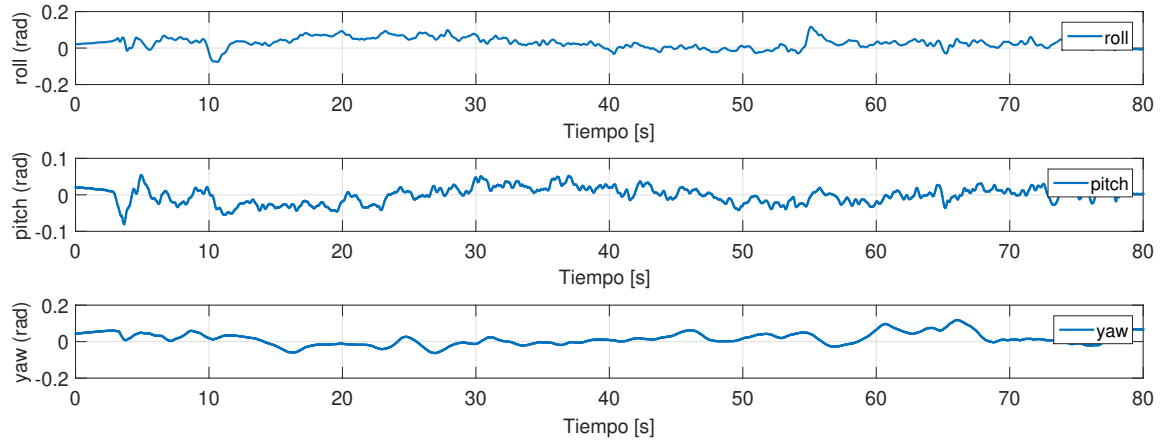


Figura 4.34: Señales de los ángulos del vehículo durante el vuelo.

La figura siguiente muestra las señales de velocidad en cada eje:

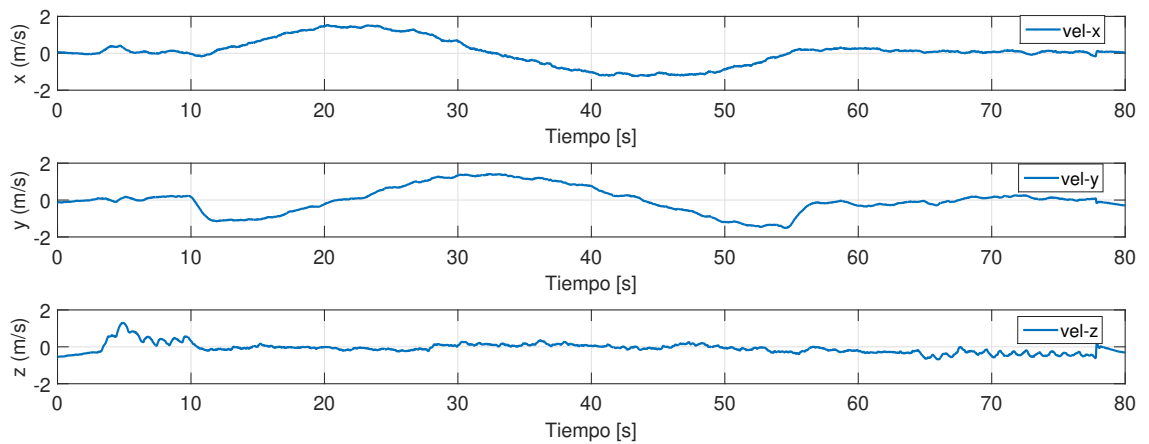


Figura 4.35: Velocidades traslacionales observadas en el vehículo durante la misión aérea.

Finalmente se muestran las señales de control. En la dinámica  $z$  se aplicó el control por modos deslizantes y para las demás el control basado en el algoritmo de Backstepping.

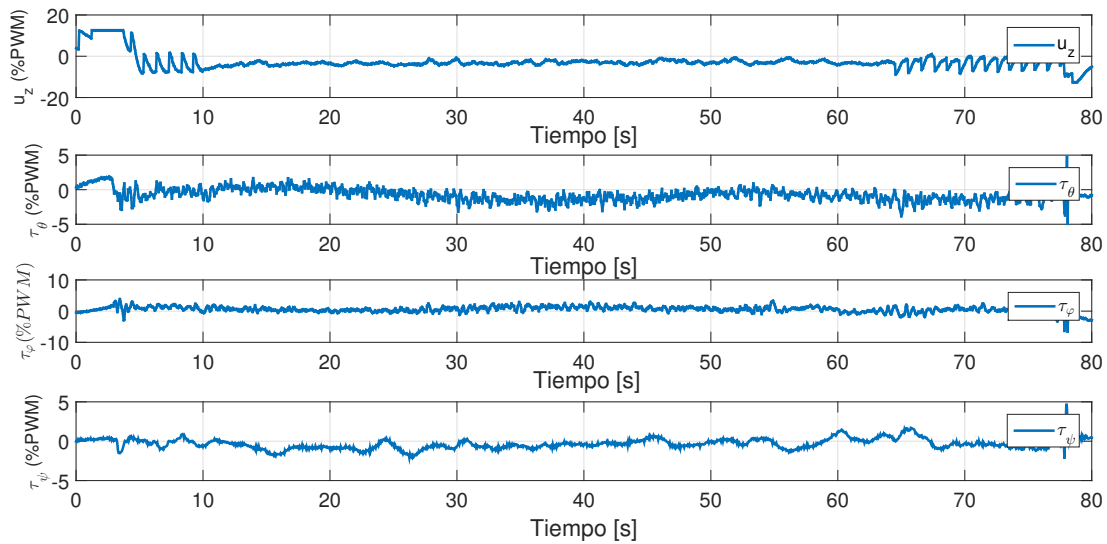


Figura 4.36: Señales de control para cada dinámica del cuadricóptero.

Para demostrar el desempeño del control y la repetitividad de los resultados, se realizaron 10 experimentos los cuales fueron intercalados entre el control por backstepping y un control Proporcional-Derivativo, el cual se sintonizó experimentalmente. Los resultados se muestran en la tabla 4.7. Se obtuvo la (IAE) Integral Absoluta del Error por cada prueba y se promediaron los resultados. También se agregó la desviación estándar del error promedio.

<i>Indicador</i>	<i>BS</i>	$\sigma$	<i>PD</i>	$\sigma$
IAE $x$	1351.63	0.063	1440.63	0.070
IAE $y$	1128.32	0.043	1260.39	0.049
IAE $z$	493.7	0.055	491.36	0.072

Tabla 4.7: Comparación de la IAE durante la misión aérea.

En la tabla 4.7 se puede observar que el error promedio es menor cuando se utiliza el algoritmo por backstepping comparado con el PD. También se puede observar que las desviaciones estándar son menores cuando se implementa el control por BS, por lo que se puede concluir que este control tiene un buen desempeño. También es importante notar que

---

este controlador involucra las no linealidades del modelo a diferencia del PD que es lineal. Estos resultados se presentaron en el artículo [12] el cual fue publicado en el IEE-Access.

## 4.8 Navegación en exteriores utilizando un sensor de barrido láser

En esta sección se presenta el método empleado para la navegación en presencia de obstáculos, específicamente “árboles”. Como ya se ha explicado en los Capítulos anteriores, se está utilizando un sensor de barrido láser para la detección de los árboles. Estos datos son analizados para hacer reconocimiento del entorno. El algoritmo general para la navegación se puede dividir en tres partes, el seguimiento de la trayectoria propuesta, la repulsión en caso de que se detecte un obstáculo y finalmente la evasión. Durante estas fases se lleva a cabo el reconocimiento del entorno es decir la adquisición, tratamiento y análisis de los datos obtenidos del sensor láser. Estas fases se explicarán a detalle en las siguientes sub-secciones.

Tabla 4.8: Algunos parámetros utilizados en el algoritmo de evasión, planeación y repulsión

Parámetros	Valor
Masa del vehículo, (m)	1.5 kg
Salud de la medición, ( $H_s$ )	150 u
Distancia mínima para analizar, ( $l_m$ )	0.4 m
Distancia máxima para analizar, ( $l_M$ )	5m
Distancia mínima de seguridad, ( $l_s$ )	3 m
Distancia mínima entre puntos consecutivos, $d_m$	0.15 m
Tiempo de muestreo, ( $T$ )	0.01 s

**El procedimiento de la navegación utilizando el sensor de barrido láser en presencia de árboles es el siguiente:**

1. Al inicio conocemos la posición inicial en  $(x_0, y_0)$  del vehículo, también conocemos la posición final  $(x_f, y_f)$  que el vehículo debe alcanzar.



2. El vehículo empieza a navegar siguiendo una línea recta mientras la ruta esté libre de obstáculos.
  
3. Si se detecta un obstáculo a una distancia menor a  $l_m$  se inicia un procedimiento de identificación; donde se estiman las dimensiones y la posición del obstáculo.
  
4. La información del obstáculo servirá para iniciar el proceso de “repulsión” que consiste en alejarse en sentido contrario a la posición del árbol, a una velocidad definida previamente.
  
5. Cuando la fase de repulsión haya terminado se inicia el proceso de “evasión” el cual consiste en rodear al obstáculo siguiendo una forma circular. Note que aquí el vehículo debe decidir si evadir por el lado derecho o izquierdo de acuerdo al análisis que haya realizado en el paso 3.
  
6. Finalmente el vehículo continúa desplazándose hacia las coordenadas finales y el procedimiento comienza a repetirse hasta que  $x \approx x_f$  y  $y \approx y_f$ .

Este procedimiento puede verse de manera gráfica en la figura 4.37.

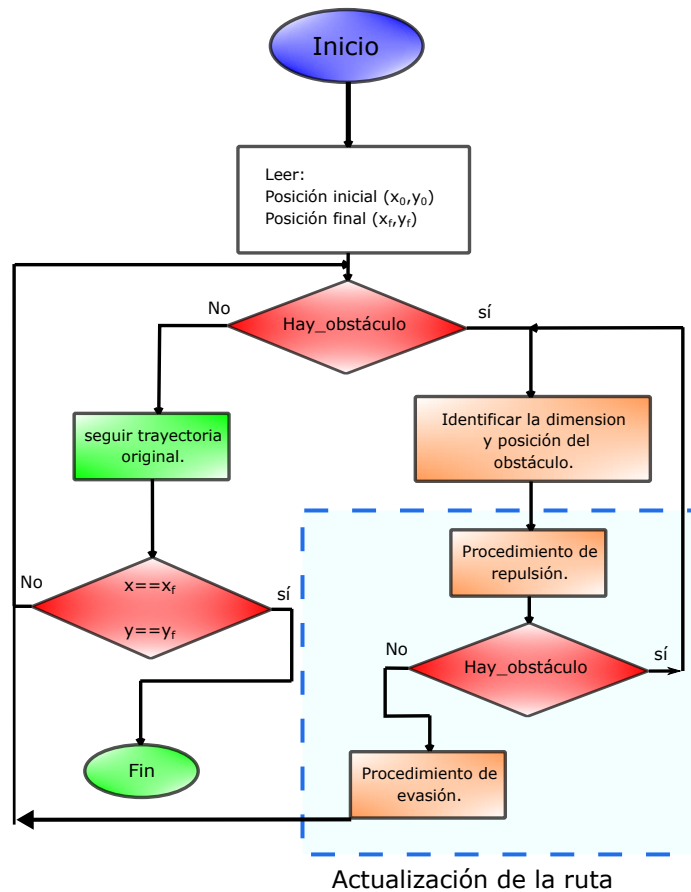


Figura 4.37: Diagrama general del proceso de navegación utilizando un sensor de barrido láser

### 4.8.1 Adquisición y tratamiento de los datos

El escaner de barrido láser (sweep v1.0) recolecta mediciones de un plano realizando un barrido con un sensor de distancia láser. Este devuelve un conjunto de datos  $W = \{T_i | i = 1, 2, 3, \dots, N\}$  por cada vuelta completada, donde  $T_i = (l_i, \bar{\theta}_i, H_i)$  es una tupla formada por una distancia, un ángulo y un parámetro de confiabilidad; el ángulo  $\bar{\theta}$  representa la dirección en que se realizó la medición y puede tomar valores dentro de  $(0, 2\pi]$ ,  $l_i \in \mathfrak{R}^+$  es la distancia al objeto detectado en metros y  $H_i$  corresponde a la salud de la medición  $i$ , ésta última

se encuentra entre  $[0, 254]$ , mientras más cercano se encuentre a 254 la medición será más confiable.

La instalación del sensor láser sobre la plataforma experimental, se realizó de manera conveniente para que el frente del multirrotor correspondiera con la posición  $\bar{\theta} = 0$  del sensor. En la figura 4.38 se establece un sistema coordenado  $b(x_b, y_b, z_b)$  ligado al cuerpo. El sensor gira de forma periódica en sentido anti-horario partiendo del eje  $x_b^+$ .

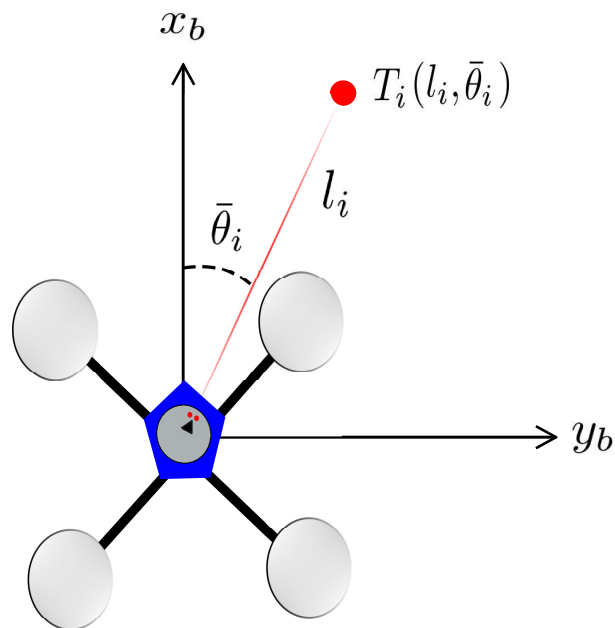


Figura 4.38: Montaje del sensor de barrido láser sobre la plataforma experimental 3.2

Los datos leídos fueron sometidos a un procedimiento de depuración con el fin de reducir el número de operaciones a realizar por el micro-controlador y también eliminar lecturas con ruido elevado (tomando de referencia  $H_i$ ). La figura 4.39 muestra las distancias medidas por el sensor en un tiempo de 5 segundos, lo cual equivale a poco más de 3 barridos completos.

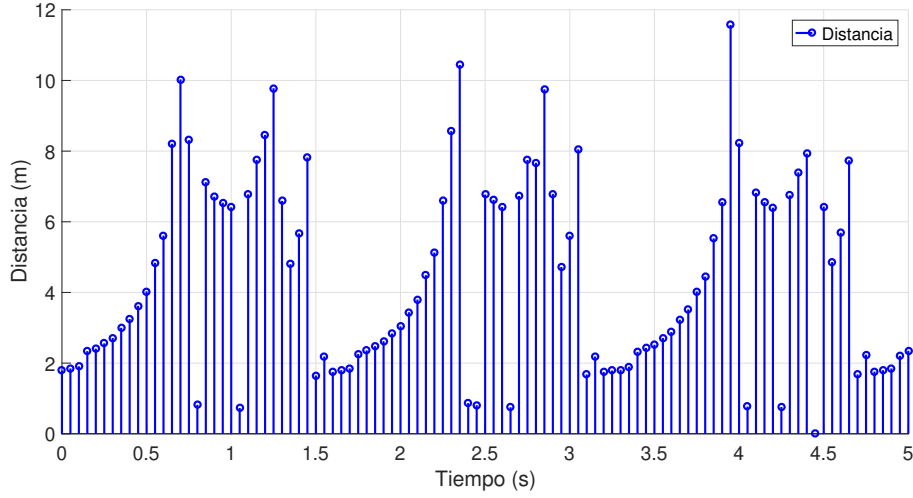


Figura 4.39: Mediciones del sensor de barrido láser sin ningún post-tratamiento

Defina ahora  $p_k$  una dupla perteneciente a  $W$ , la cual representa la ubicación  $(l_k, \bar{\theta}_k)$ , del objeto identificado en la lectura  $i$ . donde  $k = \{1, 2, 3, 4, \dots, n\}$ ,  $n \leq k$  y es el número de datos que cumplen con:

$$p_k = \{T_i | H_i > H_s \text{ y } l_m < l_i < l_M\}. \quad (4.8)$$

En la figura 4.40 se ha descartado toda  $T_i$  que no cumpla con un parámetro de confiabilidad  $H_i$  mayor a  $H_s$ , los valores descartados son aquellos que pueden causar problemas en la estimación de la posición y dimensiones del obstáculo, debido al ruido que presentó la medición.

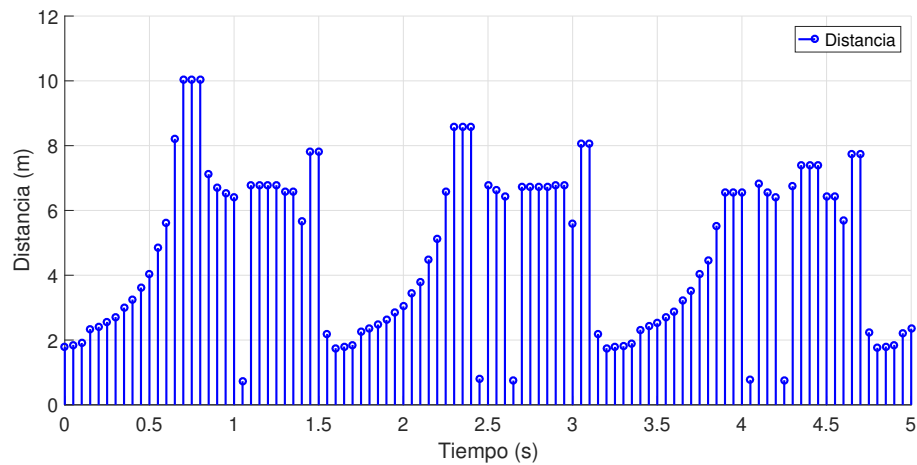


Figura 4.40: Mediciones obtenidas con el sensor láser con  $H_i$  mayor a  $H_s$ .

La siguiente figura 4.41 muestra finalmente las mediciones que cumplen con la condición 4.8, es decir las mediciones de distancia fueron umbralizadas con el objetivo de reducir el número de datos a procesar, este conjunto de datos  $p$  será utilizado para la estimación de las dimensiones y la ubicación de los obstáculos.

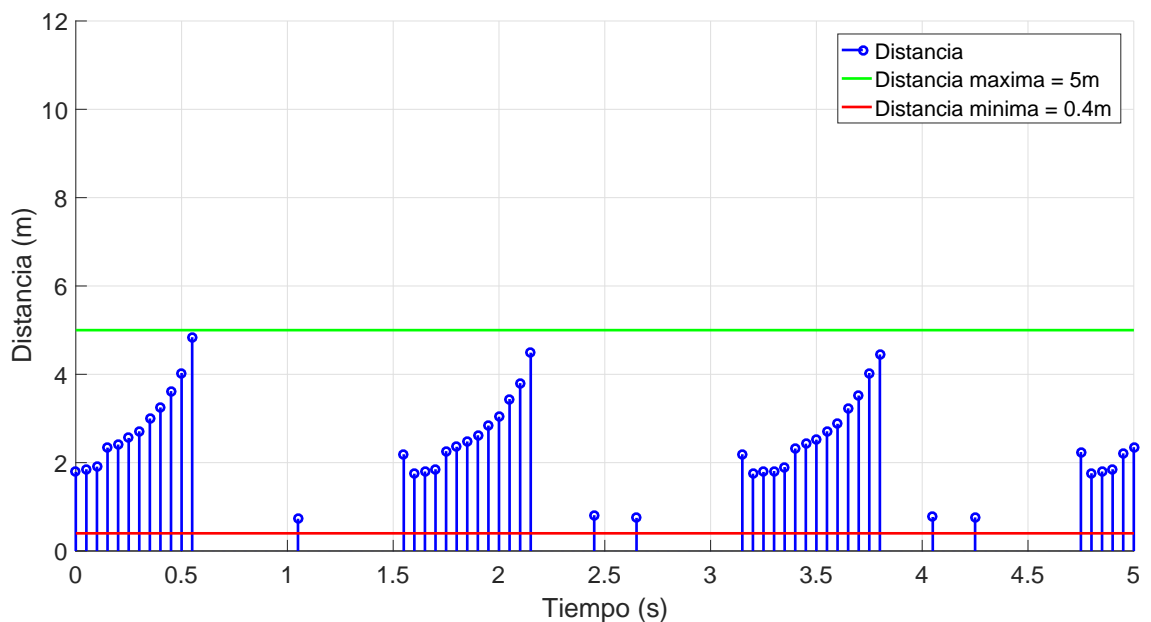


Figura 4.41: Mediciones obtenidas del sensor láser que fueron seleccionadas para el análisis de los obstáculos.

### 4.8.2 Algoritmo de repulsión

En esta subsección se aborda el diseño del algoritmo de evasión de obstáculos, el cual identificará obstáculos y generará nuevas posiciones para el UAV con el fin de evitar una colisión. Se considera un “obstáculo” a uno o varias lecturas del sensor de barrido láser con una distancia menor a una distancia segura ( $l_s$ ), con respecto al vehículo.

Con el fin de discriminar entre varios obstáculos leídos durante un mismo barrido, se ha establecido una distancia máxima  $d_m$  entre puntos leídos de un mismo obstáculo, es decir;  $\overline{P_k P_{k-1}}$  es la distancia entre 2 lecturas consecutivas y está dada por:

$$\overline{P_k P_{k-1}} = \sqrt{(x_{k-1} - x_k)^2 + (y_{k-1} - y_k)^2}, \quad (4.9)$$

donde  $x(k)$  y  $y(k)$  son las coordenadas de cada punto identificado como un obstáculo. Note que a partir de un punto detectado se considera un obstáculo.

Un punto  $p_k$  pertenece a un obstáculo si  $\overline{P_k P_{k-1}} < d_m$ , donde  $d_m$  es una distancia definida por el usuario como mínima entre  $p(k)$  y  $p(k-1)$ , en caso contrario se genera un nuevo conjunto de puntos y se estiman los parámetros del obstáculo anterior.

Para simplificar el procesamiento de datos en el autopiloto se consideró a los obstáculos como circunferencias de radio  $r_{obs}$  con centro en  $(x_{obs}, y_{obs})$  con respecto al sistema inercial,  $r_{obs}$  se estima para un obstáculo con la siguiente expresión:

$$r_{obs} = \frac{\sqrt{(x_k - x_1)^2 + (y_k - y_1)^2}}{2}. \quad (4.10)$$

Las coordenadas del centro del obstáculo se estiman proyectando  $l_{obs}$  a un ángulo  $\theta_{obs}$  respecto del sistema de referencia inercial, donde:

$$l_{obs} = \frac{\sum_1^k l_k}{k} + r_{obs} \quad (4.11)$$

y

$$\theta_{obs} = \frac{\sum_1^k \bar{\theta}_k}{k}. \quad (4.12)$$

Donde  $k$  es el número de mediciones incluidas en un obstáculo. La nueva posición a obtener por el UAV para mantenerse a una distancia segura del objeto está relacionada con la distancia que el objeto invadió la zona segura del UAV, la cual está definida como:

$$x_{New} = x_{pos} + (l_s - l_{obs} - \Delta r) \sin(\theta_{obs} + \pi) \quad (4.13)$$

y

$$y_{New} = y_{pos} + (l_s - l_{obs} - \Delta r) \cos(\theta_{obs} + \pi). \quad (4.14)$$

Donde  $x_{pos}$  y  $y_{pos}$  son las coordenadas del vehículo respecto al sistema inercial y  $\Delta r$  es un radio extra que se agrega a la nueva posición para evitar repetir continuamente el proceso de repulsión.

### 4.8.3 Algoritmo de planeación y evasión de obstáculos

En este apartado se explica el proceso de evasión, una vez que el procedimiento de identificación y repulsión se han llevado a cabo de manera satisfactoria. De acuerdo a la figura 4.42 la ubicación del obstáculo se puede estimar con:

$$x_{obs} = x_{pos} + l_{obs} \sin \theta_{obs} \quad (4.15)$$

y

$$y_{obs} = y_{pos} + l_{obs} \cos \theta_{obs}. \quad (4.16)$$

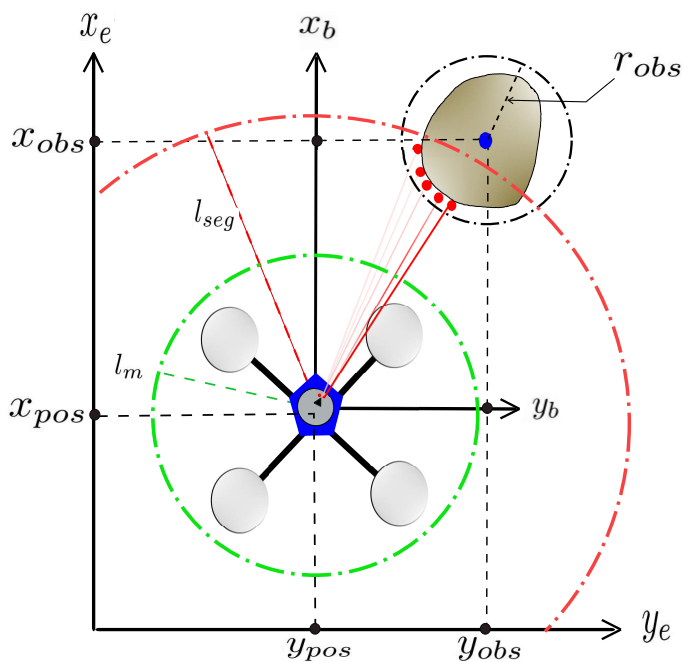


Figura 4.42: Estimación de posición y radio de un obstáculo.

El conocimiento de la ubicación del obstáculo, posición y orientación del vehículo permiten calcular una nueva trayectoria que lleve al UAV hasta la ubicación deseada y evite una colisión.



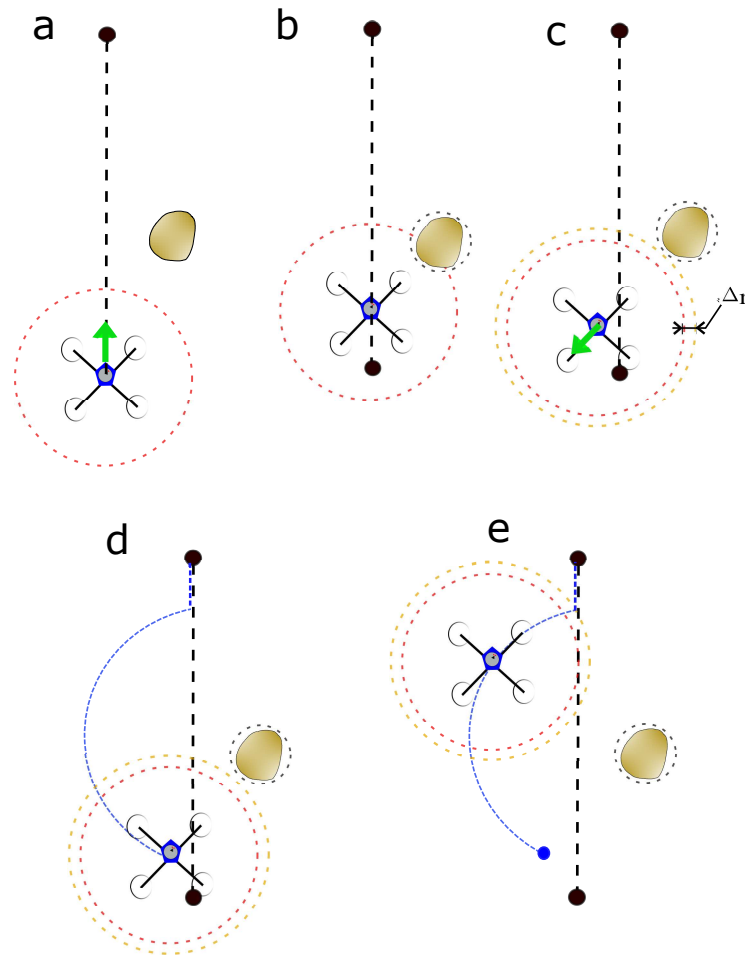


Figura 4.43: Proceso de evasión de obstáculos.

El algoritmo planteado hace uso de las dimensiones del obstáculo y además considera que el vehículo vuela a una altura constante; con esto el UAV será capaz de generar una ruta que lo alejará a una distancia segura respecto del obstáculo. Posteriormente se iniciará el seguimiento de la nueva trayectoria el cual tiene como objetivo evadir el obstáculo mientras avanza hacia la meta. Esta secuencia está ilustrada en la figura 4.43:

- En la figura 4.43a, el vehículo parte de  $(x_0, y_0)$  hacia el punto final definido  $(x_f, y_f)$ , es decir comienza con el seguimiento de la ruta original (línea negra punteada) mientras el sensor sweep realiza lecturas periódicas buscando obstáculos.
- Figura 4.43b, si durante el recorrido se identifica un obstáculo, el UAV calcula una

nueva trayectoria hacia una posición segura.

- Figura 4.43c, el vehículo realiza la navegación hacia la posición calculada con las ecuaciones 4.13 y 4.14.
- Figura 4.43d, una vez alcanzada la posición  $(x_{New}, y_{New})$ , calcula una trayectoria de segmento de arco (línea punteada azul) la cual rodeará al obstáculo a una distancia  $d_s + \Delta r$  del punto  $(x_{obs}, y_{obs})$ .
- Figura 4.43e, el vehículo realiza el seguimiento de la nueva ruta la cual concluirá hasta la intersección con la trayectoria original.

Mientras se realiza la evasión del obstáculo el algoritmo de repulsión continúa activo para evitar colisiones con objetos que se encuentren sobre la nueva trayectoria. El procedimiento anterior se muestra en el diagrama de flujo de la figura 4.37.

#### 4.8.4 Resultados experimentales

En esta subsección se presentan los resultados obtenidos implementado el algoritmo presentado anteriormente. Este algoritmo se implementó en la plataforma experimental descrita en el Capítulo 3. Las mediciones de posición traslacional se obtuvieron de dos formas: la primera se realizó con el sensor de flujo óptico y la segunda utilizando las mediciones del GPS. Cada una presentó ventajas y desventajas. Dado que la navegación se realizó bajo los árboles, la precisión del GPS no era muy buena (entre 1 y 2m), por otro lado el sensor de flujo óptico únicamente estima la velocidad en píxeles, la posición se obtuvo derivando esta señal, sin embargo presentó el problema de falsas mediciones debido a las pequeñas rotaciones que presentaba el vehículo. Las pruebas se realizaron de ambas maneras pero finalmente se eligieron los experimentos realizados con ayuda del GPS para reportar en esta tesis.

La técnica de control utilizada para hacer el seguimiento de la ruta se llevó a cabo con el control no lineal subóptimo estudiando anteriormente. La misión aérea que realizó el vehículo se divide en 3 fases:

1. Despegue: Esta fase se realizó de manera manual, es decir el piloto envió las consignas de posición utilizando el radio control.
2. Navegación: El vehículo se dejó totalmente autónomo con una altura fija y con la consigna de seguir una línea recta en el eje  $x$  es decir avanzar hasta  $(x_f = 20, y_f = 0)$ . El algoritmo descrito arriba permitió evadir los árboles que se encontró durante su recorrido.
3. Aterrizaje: El piloto retomó el control nuevamente y aterrizó la aeronave.

En la gráfica 4.44, podemos observar las primeras 3 variables de estado  $x$ ,  $y$  y  $z$  los cuales corresponden a la posición del vehículo respecto de la tierra. También podemos observar los valores deseados en cada instante de tiempo, estas señales se fueron modificando por el algoritmo de planeación y evasión de rutas ya que inicialmente el vehículo sólo debía realizar una línea recta en el eje  $x$ .

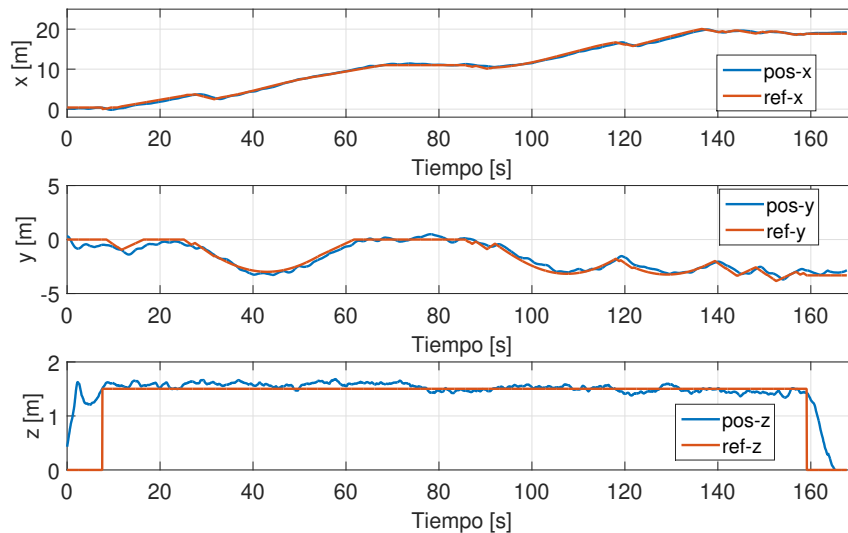


Figura 4.44: Señales de posición en los ejes  $x$ ,  $y$  y  $z$

La gráfica 4.45 muestra las señales  $\phi$ ,  $\theta$  y  $\psi$  las cuales representan la posición angular del vehículo. Estas señales nos muestran como va cambiando la orientación del vehículo durante

la navegación, también podemos observar que el ángulo de yaw  $\psi$  se mantuvo constante cercano a cero.

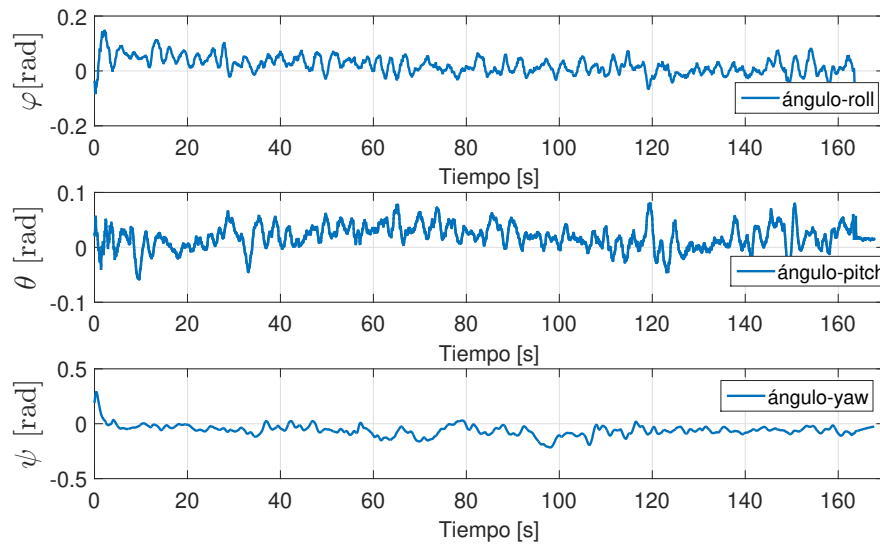


Figura 4.45: Señal de orientación roll, pitch y yaw

En la siguiente gráfica 4.46 se pueden observar las señales de velocidad traslacionales  $\dot{x}$ ,  $\dot{y}$  y  $\dot{z}$ , podemos ver que estas velocidades son menores a  $1m/s$ , es decir se eligieron velocidades bajas debido a que había un retraso (hasta 1 segundo) en la lectura de las distancias dadas por el sensor láser Sweep v1. La posición en  $z$  debía mantenerse constante, por ello se observa que la velocidad en altura está muy cercana a cero. Note que es posible observar donde el vehículo ejecutaba la maniobra de repulsión por ejemplo en  $t = 25s$  y  $t = 120s$  en  $\dot{x}$ , cuya interpretación es clara pues el vehículo se movía en línea recta hacia adelante pero al encontrarse un obstáculo el UAV debía retroceder inmediatamente, así que eso es precisamente lo que se observa en estos instantes de tiempo, un cambio en el sentido de la velocidad.

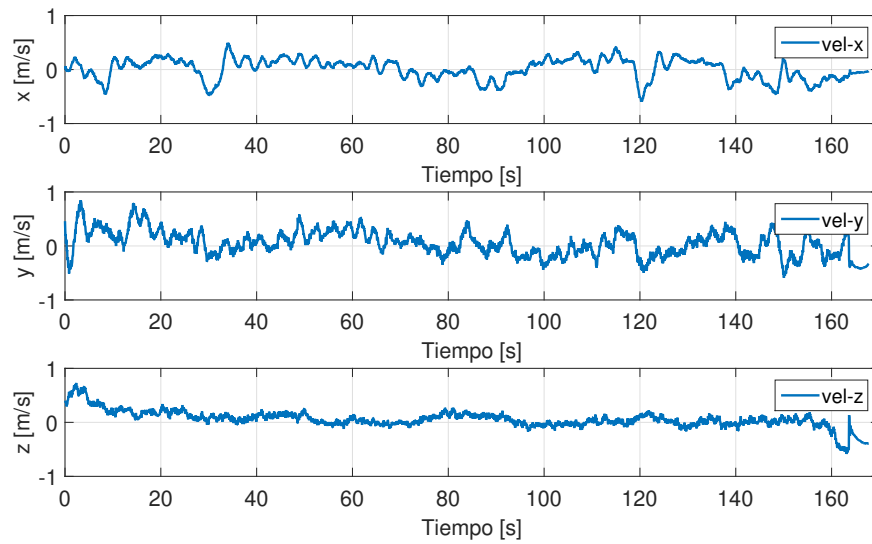


Figura 4.46: Velocidades medidas en el vehículo durante la navegación

También se agregó la señal de los errores de seguimiento 4.47 durante la navegación, estas señales nos muestran la diferencia entre la referencia que el vehículo debía seguir y la señal real medida por el GPS. El error promedio para el eje  $x$  es de  $0.2330\text{ m}$  y para  $y$  es de  $0.2612\text{ m}$ , los cuales consideramos que son aceptables respecto al error de medición que el GPS presentó durante los experimentos. El error en  $z$  es aún menor apenas  $0.062\text{ m}$ , esto debido a que la referencia en altura fue una constante. Estos promedios se obtuvieron con la siguiente fórmula:  $Error\ promedio_{x,y,z} = \frac{\sum_1^N |error_{x,y,z}(k)|}{N}$ , donde  $N$  es la cantidad de muestras de la señal.

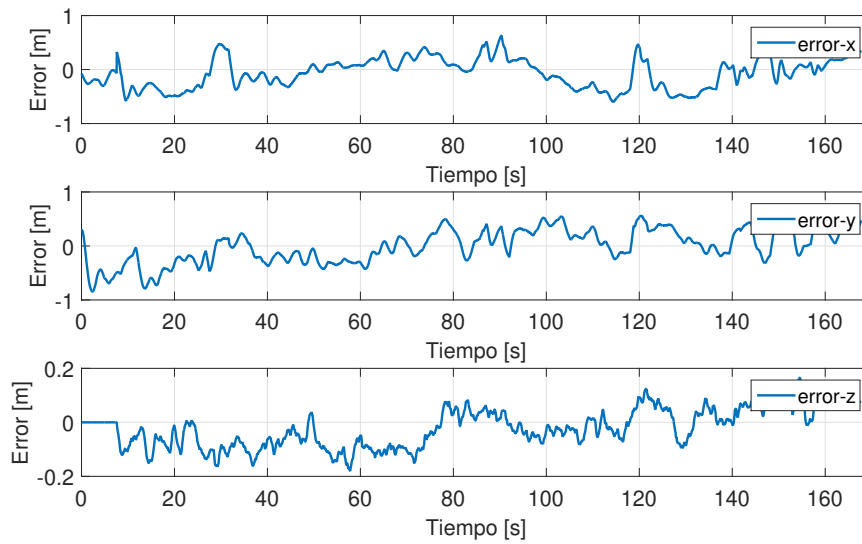


Figura 4.47: Señales del error de seguimiento en los ejes  $x$ ,  $y$  y  $z$  durante la misión aérea

La gráfica 4.48 muestra las señales de control calculadas durante la misión aérea, estas señales se presentaron en valores de  $\%PWM$  con el objetivo de ver que tan grande o pequeña es la señal que están enviando los controladores de velocidad a los motores. La parte negativa únicamente nos indica el sentido en que se ingresó la señal. Algunas observaciones que se pueden hacer por ejemplo, se puede ver que la señal de control  $u_z$  el cual corresponde a la entrada para la altura, va aumentando poco a poco, esto se debe a que el voltaje de la batería va disminuyendo por lo tanto la sustentación disminuye y en consecuencia el control de altura debe aumentar para lograr mantener el vehículo a la altura deseada. Para la entrada  $\tau_\psi$  se mantiene cercano a cero, debido a que la referencia en este ángulo se fijó en cero.

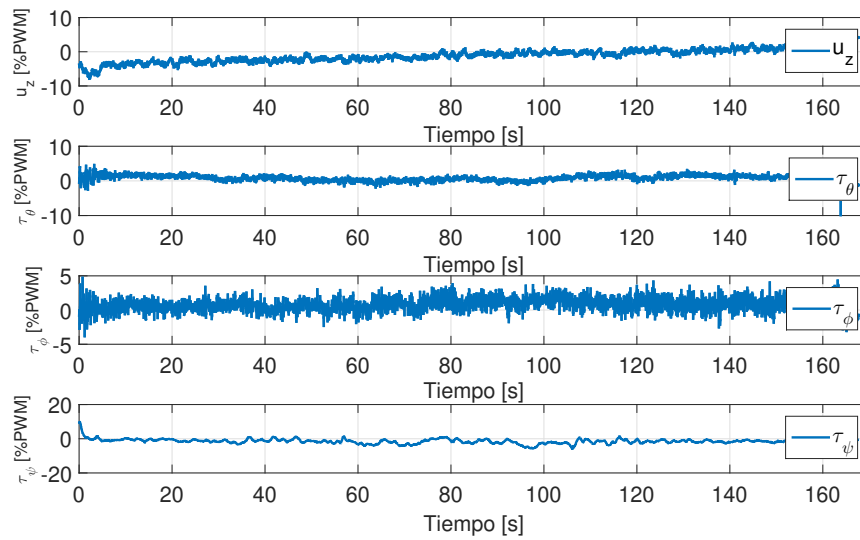


Figura 4.48: Señales de control utilizadas durante la navegación

Finalmente la gráfica 4.49 nos muestra la vista en 3 dimensiones, lo cual nos da una idea más acertada del comportamiento del vehículo durante todo el experimento. Se muestra en color verde y con líneas punteadas la referencia original que el vehículo debía seguir, pero como fue detectando algunos obstáculos, esta referencia se fue actualizando de manera automática. En color azul claro y azul fuerte podemos ver la referencia y la posición medida respectivamente. También podemos observar que el vehículo alcanza aproximadamente la referencia final  $x_f = 20\text{ m}$  pero en  $y$  no, esto se debió a que en la consigna  $(x_f, y_f)$  habían obstáculos y se puede observar como el vehículo intenta llegar a esa referencia pero el algoritmo de evasión no se lo permite.

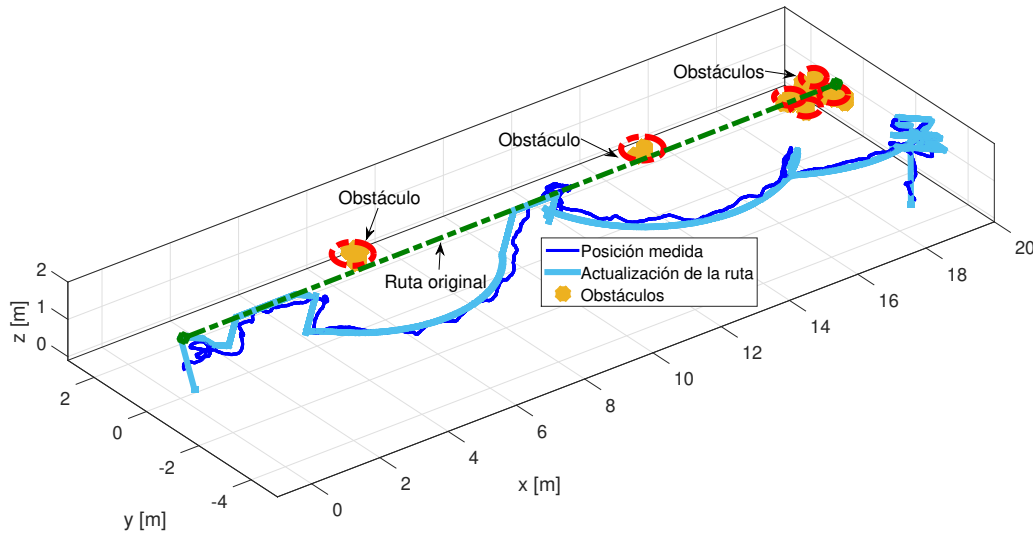


Figura 4.49: Planeación de trayectorias en el exterior y evasión de obstáculos utilizando un control subóptimo no lineal discreto de horizonte finito

## 4.9 Conclusiones

Los primeros resultados muestran la implementación de diversos algoritmos de control los cuales fueron mejorando poco a poco, esto dependió en gran medida de los sensores utilizados. Pudimos constatar que las mediciones de posición obtenidas del sensor de flujo óptico tienen mejor precisión, pero se añaden falsas mediciones en la velocidad debido a las pequeñas rotaciones que existen durante el vuelo. Por otro lado, el GPS ofrece mejores mediciones en posición y velocidad sin embargo existen algunas discrepancias en la posición traslacional entre lo medido por este sensor y lo que se observa en la realidad. Pero al final se eligieron las mediciones obtenidas por el GPS dado que la estimación de la velocidad es mejor respecto del sensor de flujo óptico.

Finalmente se utilizó una de las técnicas de control ya implementadas para hacer la parte de la navegación en el exterior en presencia de árboles. Se explicó a detalle el algoritmo



---

que involucra la parte de repulsión, planeación de trayectorias y evasión de obstáculos. Este algoritmo se puede aplicar a otro tipo de obstáculos siempre y cuando se tengan las mediciones del sensor de barrido láser.



# CAPÍTULO 5

## CONCLUSIONES Y TRABAJO FUTURO

### 5.1 Conclusiones

Las principales aportaciones de esta tesis son, la navegación en el exterior en presencia de árboles utilizando un sensor de barrido láser el cual se montó sobre una vehículo de 4 rotores, se propuso un algoritmo con enfoque geométrico lo que permitió satisfactoriamente realizar tareas de repulsión en caso de obstáculos, planeación de trayectorias y evasión. Se utilizó la técnica basada en un Control subóptimo de horizonte finito el cual fue sintetizado e implementado en un cuadricóptero previamente. Se considera que los resultados obtenidos fueron satisfactorios comparado con los pocos trabajos que están reportados en la literatura que en su mayoría muestran únicamente resultados de simulación. Estos resultados se presentan de manera experimental y en tiempo real, utilizando la plataforma experimental 3.1 lo cual supuso un gran reto en la parte tecnológica puesto que el sistema se vuelve complejo mientras más sensores se agregan. Este algoritmo fue probado evadiendo la parte baja de los árboles sin embargo el algoritmo puede funcionar para cualquier obstáculo que pueda ser detectado por el sensor de barrido láser sweep v1.

Otro aporte importante es la implementación de distintas técnicas de control para esta-

bilización y seguimiento de trayectorias los cuales fueron implementados satisfactoriamente en un vehículo de 4 rotores, en cada técnica se realizaron análisis estadísticos que muestran la viabilidad de los algoritmos propuestos. Por ejemplo en la sección 4.5 se sintetizó un control subóptimo no lineal de horizonte finito el cual está basado en la teoría de programación dinámica. Este fue modificado bajo ciertas suposiciones proponiendo una función aproximada de la ecuación de Bellman, para evitar la solución a la Ecuación Algebraica Discreta de Ricatti, el cual aún es un problema abierto para los sistemas no lineales. En la sección 2.6 se sintetizó un control basado en Backstepping combinado con un Control por Modos Deslizantes Terminales, ambos enfoques fueron modificados respecto de los algoritmos comunes para poder implementarse en la planta construida. En ambos casos se muestran resultados experimentales los cuales fueron presentados y discutidos también en revistas y/o congresos internacionales.

También se implementaron otros algoritmos basados en optimización de manera experimental:

- Control Óptimo discreto de horizonte infinito,
- Control Óptimo discreto de horizonte finito,
- Control Subóptimo discreto combinado con un control Super-twisting.

En general se puede decir que los controles basados en una estrategia de optimización son una opción viable para implementarse en un vehículo multi-rotor puesto que nos permiten tener un ahorro en el consumo energético. Sin embargo los controles no lineales y/o robustos permiten tener un mejor desempeño hablando en términos del error, aunque el consumo de energía es mayor. En el Capítulo 4 los resultados experimentales que se obtuvieron nos indican que la mejor opción es implementar estructuras de control que involucren técnicas de control basadas en optimización combinado con técnicas de control robustas, en [15] se aplicó esta técnica el cual brindó excelentes resultados que permitieron tener un buen desempeño

conservado aún así un ahorro en el consumo energético.

## 5.2 Trabajo futuro

En esta tesis se abordó la navegación basada en sensores, el objetivo inicial era poder evitar obstáculos que el vehículo se encontrara en su camino, de manera específica la evasión de árboles para lograr navegar en bosques. Sin embargo sólo se utilizó un sensor láser, lo cual permitió obtener una reconstrucción en dos dimensiones del entorno de navegación.

El trabajo futuro está abierto a muchas posibilidades, siguiendo la pauta de esta tesis, se proponen las siguientes tareas que podrían agregarse.

- Reconstrucción del entorno en 3 dimensiones para hacer seguimiento en los 3 ejes traslacionales,
- Utilizar más sensores, especialmente aquellos que se basan en visión artificial,
- Aumentar la velocidad de desplazamiento durante la navegación,
- Utilizar otros algoritmos de planeación de trayectorias,
- Implementar el algoritmo en escenarios más complejos.

La tecnología en el área de vehículos aéreos crece muy rápido por lo que cada día se fabrican más sensores los cuales podrían también implementarse para hacer navegación autónoma.



# BIBLIOGRAFÍA

- [1] AIT-JELLAL, R., AND ZELL, A. Outdoor obstacle avoidance based on hybrid visual stereo slam for an autonomous quadrotor mav. In *Mobile Robots (ECMR), 2017 European Conference on* (2017), IEEE, pp. 1–8.
- [2] ALTUĞ, E., OSTROWSKI, J. P., AND TAYLOR, C. J. Control of a quadrotor helicopter using dual camera visual feedback. *The International Journal of Robotics Research* 24, 5 (2005), 329–341.
- [3] ANIS, H., FADHILLAH, A. H. I., DARMA, S., AND SOEKIRNO, S. Automatic quadcopter control avoiding obstacle using camera with integrated ultrasonic sensor. In *Journal of Physics: Conference Series* (2018), vol. 1011, IOP Publishing, p. 012046.
- [4] ARNOLD III, W. F., AND LAUB, A. J. Generalized eigenproblem algorithms and software for algebraic riccati equations. *Proceedings of the IEEE* 72, 12 (1984), 1746–1754.
- [5] BARRY, A. J., AND TEDRAKE, R. Pushbroom stereo for high-speed navigation in cluttered environments. In *2015 IEEE International Conference on Robotics and Automation (ICRA)* (2015), IEEE, pp. 3046–3052.
- [6] BUGAYONG, A., AND RAMOS JR, M. Obstacle detection using binocular stereo vision in trajectory planning for quadcopter navigation. In *Journal of Physics: Conference Series* (2018), vol. 976, IOP Publishing, p. 012008.

- [7] CAMPOS-MACÍAS, L., GÓMEZ-GUTIÉRREZ, D., ALDANA-LÓPEZ, R., DE LA GUARDIA, R., AND PARRA-VILCHIS, J. I. A hybrid method for online trajectory planning of mobile robots in cluttered environments. *IEEE Robotics and Automation Letters* 2, 2 (2017), 935–942.
- [8] CASTILLO, P., GARCÍA, P., LOZANO, R., AND ALBERTOS, P. Modelado y estabilización de un helicóptero con cuatro rotores. *Revista Iberoamericana de Automática e Informática Industrial RIAI* 4, 1 (2007), 41–57.
- [9] DE JESUS RUBIO, J., CRUZ, J. H. P., ZAMUDIO, Z., AND SALINAS, A. J. Comparison of two quadrotor dynamic models. *IEEE Latin America Transactions* 12, 4 (2014), 531–537.
- [10] DONG, T., LIAO, X., ZHANG, R., SUN, Z., AND SONG, Y. Path tracking and obstacles avoidance of uavs-fuzzy logic approach. In *Fuzzy Systems, 2005. FUZZ'05. The 14th IEEE International Conference on* (2005), IEEE, pp. 43–48.
- [11] GARCIA, O., FLORES, D., SANTOS, O., ROMERO, H., SALAZAR, S., AND LOZANO, R. Autonomous take-off and landing on a colored platform. In *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on* (2017), IEEE, pp. 877–884.
- [12] GARCÍA, O., ORDAZ, P., SANTOS-SÁNCHEZ, O.-J., SALAZAR, S., AND LOZANO, R. Backstepping and robust control for a quadrotor in outdoor environment: An experimental approach. *IEEE Access* (2019).
- [13] GARCIA, O., SANTOS, O., ROMERO, H., SALAZAR, S., ET AL. On the tracking trajectory using optimal control in a quadrotor helicopter: Experimental results. In *2015 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)* (2015), IEEE, pp. 142–151.
- [14] GARCIA PEREZ, O. Seguimiento Óptimo de trayectorias con un cuadricóptero. Master's thesis, CINVESTAV, 2015.



- 
- [15] GONZÁLEZ, I., GARCÍA, O., SANTOS, O., SALAZAR, S., AND LOZANO, R. Robust and suboptimal control for an uav: Control synthesis and experimental validation under cross wind disturbances. *Under review* (2017).
- [16] GONZÁLEZ-HERNÁNDEZ, I., SALAZAR, S., MUÑOZ, F., AND LOZANO, R. Super-twisting control scheme for a miniature quadrotor aircraft: Application to trajectory-tracking problem. In *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on* (2017), IEEE, pp. 1547–1554.
- [17] HAUSER, J., SASTRY, S., AND MEYER, G. Nonlinear control design for slightly non-minimum phase systems: Application to v/stol aircraft. *Automatica* 28, 4 (1992), 665–679.
- [18] HRABAR, S. 3d path planning and stereo-based obstacle avoidance for rotorcraft uavs. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on* (2008), IEEE, pp. 807–814.
- [19] KHALIL, H. K., AND GRIZZLE, J. *Nonlinear systems*, vol. 3. Prentice hall New Jersey, 1996.
- [20] KIRK, D. E. *Optimal control theory: an introduction*. Courier Dover Publications, 2012.
- [21] LOZANO, R. *Unmanned aerial vehicles: Embedded control*. John Wiley & Sons, 2013.
- [22] MOHANTA, J., PARHI, D. R., MOHANTY, S., AND KESHARI, A. A control scheme for navigation and obstacle avoidance of autonomous flying agent. *Arabian Journal for Science and Engineering* 43, 3 (2018), 1395–1407.
- [23] PIXHAWK. <https://pixhawk.org>, Last modification, 2016/12/30 10:35.
- [24] PIXY CAMERA. <https://pixycam.com/pixy-cmucam5/>, Abril 2019.
- [25] PX4 FLOW. <https://docs.px4.io/en/sensor/px4flow.html>, Abril de 2019.

- [26] RAHMAN, M. F., AND SASONGKO, R. A. Obstacle avoidance for quadcopter using ultrasonic sensor. In *Journal of Physics: Conference Series* (2018), vol. 1005, IOP Publishing, p. 012037.
- [27] SÁNCHEZ, L., SANTOS, O., ROMERO, H., SALAZAR, S., AND LOZANO, R. Finite horizon nonlinear optimal control for a quadrotor: experimental results.
- [28] SANTOS, O., GARCÍA, O., ROMERO, H., SALAZAR, S., AND LOZANO, R. Finite horizon nonlinear optimal control for a quadrotor: experimental results. *ISA Transactions* (December 2017, Under review).
- [29] SANTOS, O., ROMERO, H., SALAZAR, S., GARCÍA-PÉREZ, O., AND LOZANO, R. Optimized discrete control law for quadrotor stabilization: Experimental results. *Journal of Intelligent & Robotic Systems* 84, 1-4 (2016), 67–81.
- [30] SWEEP SENSOR. <https://www.robotshop.com/en/sweep-v1-360-laser-scanner.html>, Junio de 2017.
- [31] THANH, H. L. N. N., PHI, N. N., AND HONG, S. K. Simple nonlinear control of quadcopter for collision avoidance based on geometric approach in static environment. *International Journal of Advanced Robotic Systems* 15, 2 (2018), 1729881418767575.
- [32] WANG, X., YADAV, V., AND BALAKRISHNAN, S. Cooperative uav formation flying with obstacle/collision avoidance.
- [33] YANG, X., LUO, H., WU, Y., GAO, Y., LIAO, C., AND CHENG, K.-T. Reactive obstacle avoidance of monocular quadrotors with online adapted depth prediction network. *Neurocomputing* 325 (2019), 142–158.
- [34] YU, Y., WU, Z., CAO, Z., PANG, L., REN, L., AND ZHOU, C. A laser-based multi-robot collision avoidance approach in unknown environments. *International Journal of Advanced Robotic Systems* 15, 1 (2018), 1729881418759107.