
Universidad Michoacana de San Nicolás de Hidalgo



Facultad de Ingeniería Eléctrica



**“DISEÑO DE UN CONTROLADOR PID PARA UN
CUADRICÓPTERO POR MEDIO DE ALGORITMOS
GENÉTICOS”**

TESIS

**Que para obtener el grado de:
INGENIERÍA EN ELECTRÓNICA**

Presenta:

Marvin Ortiz Corona

Asesor:

Salvador Ramírez Zavala.

Agradecimientos

A todas las personas que han estado en este proceso, gracias:

A mi padre, el cual ante su partida todo fue aún más difícil de lo que ya era, pero gracias a tus consejos y enseñanzas he podido lograr un proceso más en mi vida.

A mis amigos que han ayudado a este proceso, con el apoyo en clases y con la convivencia que ha logrado ser más a meno los días en el salón. En especial Carlos Gabriel que con su amistad pude lograr mejorar como alumno, persona, que con sus conejos logré poder adaptarme mejor a la escuela y al trabajo.

A los maestros que aportaron conocimiento, dedicación y tiempo a mi desarrollo, también por esos consejos que en algún momento fueron claves para mantenerme en la escuela, pero en especial al profesor Isidro Ignacio que, con sus enseñanzas y apoyo, he podido desarrollar esta Tesis.

A mis familiares que han ayudado de una o varias maneras, en especial a mi tía Julissa Corona, que en más de una ocasión me apoyo con su cariño y tiempo, mucha gracias.

Pero en especial forma quiero dar gracias a mi madre Erika Corona Corrales con todo mi amor y cariño te agradezco por todo lo que me has dado, sin duda has sido maravillosa como madre.

Dedicatoria

En este gran proceso, que me llena de orgullo por este logro he conocido a muchas grandes personas, hoy quiero dedicar esta obra que realice con esfuerzo y dedicación:

A mi madre Erika Corona Corrales

A mi padre Ismael Pérez Silva

A mi hermano Irving Ortiz Corona

A mis hermanas Erika y Katia.

A mi tía Julissa Corona Corral

A todas las personas que de algún modo me han ayudado en alguna parte de mi vida

Resumen

Los vehículos autónomos son equipos con la capacidad de moverse sin necesidad de un piloto, han sido utilizados desde muchos años atrás, donde nació como un sistema de defensa, su desarrollo por muchos años fue con motivos bélicos dando un gran desarrollo para mejorar estos, no fue hasta que los desarrollos en las industrias, mejora y reducción de costos de sensores ha hecho que se vuelvan equipos más baratos dependiendo del uso y capacidad de este que haya permitido ampliar sus usos, dándole usos variados como monitoreo de cultivos, recisión de estructuras civiles, localización de bancos de peses, localización de personas, acceder a lugares altamente peligrosos para humanos, monitoreo de clima, fines recreativos etc. Debido a su gran potencial como herramienta es muy importante desarrollar controladores automáticos para que estos funcionen de manera óptima, que las interferencias pueden provocar que estos tengan accidentes destrozando los equipos, además el diseño correcto de los controladores permitirá que la batería sea aprovechada al máximo.

El control automático es una ciencia que permite mejorar procesos a través de conocer la entrada y esperar una salida deseada. Existen varios tipos de control hoy en día, uno de ellos es el control convencional, el cual permite obtener controladores PID que son sistemas que corrigen la señal de control a la planta por medio del error que se tiene (la diferencia de la entrada con la salida), este proceso requiere una sintonización para que se tenga ganancias que permitan obtener respuestas deseadas.

Para este trabajo se hizo una extensa investigación del modelo matemático que describe la dinámica del cuadricóptero en cruz. Se obtuvieron matrices que describen la rotación y la posición del cuadricóptero por medio de las fuerzas generadas por cada motor. Para poder realizar la sintonización de los controladores PID se utilizó la aplicación del software MatLab de optimización con el uso de Algoritmos Genéticos (AG), la cual requiere una función que tenga como entradas las ganancias del controlador K_p (proporcional), K_i (integral) y K_d (derivativa), y de salida un índice de desempeño que permite obtener un valor numérico al comportamiento. La función por medio del modelo matemático del controlador, motores y cuadricóptero simula el comportamiento en el tiempo con las ganancias del controlador y evalúa por medio del índice de desempeño ISE (Integral del error al cuadrado) la respuesta, la app de MatLab de optimización por medio de algoritmos genéticos varía estas ganancias

hasta obtener un valor óptimo mínimo del ISE y así obtener las ganancias de los controladores.

Las simulaciones realizadas mostraron que el sistema es estable, en el control de altura corrige los errores provocados por el cambio de cualquier ángulo; además de mostrar que el uso de los algoritmos genéticos puede ser una herramienta para el diseño de controladores.

Palabras clave: PID, algoritmos genéticos, cuadricóptero, sintonización.

Abstract

Autonomous vehicles are equipment with the ability to move without the need for a pilot, they have been used for many years, where it was born as a defense system, its development for many years was for war purposes, giving great development to improve these, not It was until the developments in the industries, improvement and cost reduction of sensors has made equipment become cheaper depending on the use and capacity of this that has allowed its uses to be expanded, offering varied uses such as crop monitoring, rescission of civil structures , location of fish banks, location of people, access to highly dangerous for humans, weather monitoring, recreational fines, etc. Due to its great potential as a tool, it is very important to develop automatic controllers so that they work optimally, as interferences can cause them to have accidents destroying the equipment, in addition, the correct design of the controllers will allow the battery to be used to the maximum.

Automatic process control is a science that allows improvement through knowing the input and waiting for a desired output. There are several types of control today, one of them is conventional control, which allows obtaining PID controllers that are systems that correct the control signal to the plant by means of the error it has (the difference between the input and the output), this process requires a tuning in order to have gains that can obtain the desired responses.

For this work an extensive investigation of the mathematical model that describes the dynamics of the cross quadcopter was made. Matrices will be needed that describe the rotation and position of the quadcopter by means of the forces generated by each motor. In order to perform the tuning of the PID controllers, the application of the MatLab optimization software was reduced with the use of Genetic Algorithms (GA), which requires a function that has as inputs the controller gains K_p (proportional), K_i (integral) and K_d (derivative), and output a performance index that allows obtaining a numerical value for the behavior. The function by means of the model of the controller, motors and quadcopter simulates the behavior in time with the gains of the controller and evaluates by mathematical means the performance index ISE (Integral of the squared error) the response, the MatLab app of optimization by Through genetic algorithms, these gains are varied until obtaining a minimum optimum value of the ISE and thus obtaining the gains of the controllers.

The simulations showed that the system is stable, in the height control it corrects the errors caused by the change of any angle; in addition to showing that the use of genetic algorithms can be a tool for controller design.

Keywords: PID, genetic algorithms, quadcopter, tuning.

Contenido:

Agradecimientos.....	ii
Dedicatoria	iii
Resumen.....	iv
Abstract.....	vi
Contenido:	viii
Lista de Tablas	xii
Lista de Figuras.....	xiii
Lista de Símbolos y Abreviaturas.....	xv
Capítulo 1.....	1
1.1 <i>Introducción.....</i>	<i>1</i>
1.2. <i>Antecedentes</i>	<i>1</i>
1.3. <i>Objetivo general</i>	<i>4</i>
1.3.1. <i>Objetivos Particulares</i>	<i>4</i>
1.4. <i>Justificación</i>	<i>5</i>
1.5. <i>Metodología</i>	<i>5</i>
1.6. <i>Contenido de la tesis.....</i>	<i>6</i>
Capítulo 2.....	7
2.1. <i>Historia de los drones (UAV)</i>	<i>7</i>
2.2. <i>Clasificación de los drones.....</i>	<i>8</i>
2.3. <i>Ventajas y desventajas de los cuadricópteros</i>	<i>9</i>
2.4. <i>Aplicaciones de los drones</i>	<i>9</i>
2.5. <i>Estructura y componentes básicos de UAV de ala rotativa.....</i>	<i>10</i>
2.5.1. <i>Controlador de vuelo</i>	<i>10</i>

2.5.2. Armazón o cuerpo.....	10
2.5.3. Motores	11
2.5.4. Variadores.....	12
2.5.5. Hélices	12
2.5.6. Batería.....	13
2.5.7. Soporte anti-vibraciones	13
2.5.8 Sensores	14
2.6. <i>Movimientos de cuadricóptero</i>	14
2.7 <i>Rotación del cuadricóptero</i>	15
2.7.1 Rotación angular- Roll Φ alabeo	15
2.7.2. Rotación angular- Pitch θ cabeceo	15
2.7.3. Rotación angular- Yaw Ψ guiñada	16
2.8. <i>Movimiento traslacional</i>	16
2.8.1. Traslación eje z	16
2.8.2. Traslación en el eje x	17
2.8.3. Traslaciones en el eje y	17
Capítulo 3.....	18
3.1. <i>Modelo matemático</i>	18
3.1.1. Representación en Espacio de Estados	18
3.1.2. Fases de la modelización matemática	18
3.1.3. Protocolo para la elaboración de modelo matemático	19
3.2 <i>Modelado matemático del cuadricóptero</i>	20
3.2.1 Sistema de referencia	20
3.2.2 Ecuaciones de movimiento traslacional.....	21

3.2.3 Ecuaciones cinemática rotacional	23
3.3 Simplificación del sistema	29
3.4 Identificación del sistema	30
Capítulo 4.....	32
4.1 Sistemas de control	32
4.1.1 Control convencional.....	32
4.1.2 Control PID	33
4.2 Sintonización por algoritmos genéticos.....	34
4.3 Ciclo de evolución	35
4.3.1 Función de evaluación	36
4.3.2 Parámetros de los algoritmos genéticos.....	38
4.4 Diseño de controlador PID.....	39
4.2.1 Diseño del control de altura	40
4.2.2 Diseño del controlador de orientación (ángulos roll ϕ , pitch θ y yaw ψ).	41
4.2.3 Representación de los actuadores	41
4.3 Sintonización PID	42
4.3.1 Herramienta de optimización de MatLab	43
Capítulo 5.....	45
5.1 Resultados control Altura	45
5.2 Resultados control orientación Roll ϕ y Pitch θ	49
5.3 Resultados de control de orientación de Yaw ψ	54
5.4 Pruebas del cuadricóptero.....	60
5.5 Comparación con trabajos anteriores.....	67
Capítulo 6.....	69

6.1 Conclusiones.....	69
Referencias	71
Apéndice 1. Toolbox de optimización de MatLab	73
Anexo 1 Función de MatLab.....	77

Lista de Tablas

<i>Tabla 1 Parámetros del cuadricóptero (extraído de Paúl Ortiz, 2014).....</i>	<i>30</i>
<i>Tabla 2 Resultados de ganancias de control de altura.....</i>	<i>45</i>
<i>Tabla 3 Resultados de controladores</i>	<i>48</i>
<i>Tabla 4. Ganancias controlador Roll/pitch</i>	<i>50</i>
<i>Tabla 5 Resultados controladores Roll/Pitch</i>	<i>54</i>
<i>Tabla 6 Ganancias controladores yaw</i>	<i>55</i>
<i>Tabla 7. Resultados de controladores Yaw.....</i>	<i>59</i>
<i>Tabla 8. Resultados altura a diferentes entradas</i>	<i>61</i>
<i>Tabla 9. Resultados de Roll y Altura a diferentes entradas de Roll</i>	<i>64</i>

Lista de Figuras

Figura 1 Vehículos aéreos no tripulados	7
Figura 2 tipos de drones a. dron de ala fija b. dron Hexa cuadricóptero.....	8
Figura 3 Ejemplos de aplicaciones drones; a. fertilización b. exploración.....	10
Figura 4 Armazón de un cuadricóptero.....	11
Figura 5. Motor brushless y brushed	12
Figura 6 ESC.....	12
Figura 7 Hélices para drones	13
Figura 8 Batería LiPo para cuadricóptero	13
Figura 9 Soporte anti-vibraciones de cuadricóptero	14
Figura 10 Representación de cuadricóptero.....	14
Figura 11 Giro roll.....	15
Figura 12 Giro pitch.....	15
Figura 13 Giro Yaw	16
Figura 14 Desplazamiento en el eje z	16
Figura 15 Desplazamiento en el eje x	17
Figura 16 Desplazamiento en el eje x	17
Figura 17 Sistemas de referencia	20
Figura 18 Diagrama de bloques del cuadricóptero	40
Figura 19 Toolbox MatLab.....	44
Figura 20. Altura del cuadricóptero con controlador 1	46
Figura 21. Altura del cuadricóptero con controlador 2	46
Figura 22 Altura del cuadricóptero con controlador 3	47
Figura 23. Altura del cuadricóptero con controlador 4	47
Figura 24. Altura del cuadricóptero con controlador 5	48
Figura 25. Controlador N1 de Roll/Pitch	51
Figura 26. Controlador N2 de roll/Pitch	51
Figura 27. Controlador N3 de Roll/Pitch	52
Figura 28. Controlador N4 de Roll/Pitch	52
Figura 29. Controlador N5 de Roll/Pitch	53
Figura 30. Controlador N6 de Roll/Pitch	53
Figura 31. Controlador N1 Yaw	56
Figura 32. Controlador N2 Yaw.....	56
Figura 33. Controlador N3 Yaw	57
Figura 34. Controlador N3 Yaw	57

<i>Figura 35. Controlador N4 Yaw</i>	<i>58</i>
<i>Figura 36. Controlador N5 Yaw</i>	<i>58</i>
<i>Figura 37. Controlador N6 Yaw</i>	<i>59</i>
<i>Figura 38. Comportamiento de altura del cuadricóptero a distintas entradas.....</i>	<i>60</i>
<i>Figura 39. Comportamiento de altura del cuadricóptero a una entrada negativa.....</i>	<i>61</i>
<i>Figura 40. Comportamiento de Roll, Pitch y Yaw a entradas diferentes de altura.....</i>	<i>62</i>
<i>Figura 41 Comportamiento de altura y ángulo Roll con distintas entradas de Roll.....</i>	<i>63</i>
<i>Figura 42. Comportamiento de ángulo Roll del cuadricóptero a una entrada negativa</i>	<i>63</i>
<i>Figura 43. Comportamiento de Angulo Pitch y Yaw con diferentes entradas de Roll</i>	<i>64</i>
<i>Figura 44. Comportamiento de altura y ángulo Yaw con distintas entradas de Yaw</i>	<i>65</i>
<i>Figura 45. Comportamiento de ángulo Yaw del cuadricóptero a una entrada negativa</i>	<i>65</i>
<i>Figura 46. Comportamiento de Angulo Pitch y Roll con diferentes entradas de Yaw</i>	<i>66</i>
<i>Figura 47. Comportamiento de altura del trabajo de Paul Ortiz 2017.....</i>	<i>67</i>
<i>Figura 48. Controles de orientación Paul Ortiz</i>	<i>68</i>

Lista de Símbolos y Abreviaturas

PID proporcional integral derivativo

θ Angulo Roll

ϕ Angulo Pitch

ψ Angulo yaw

F_i =Fuerza en i-ésimo motor

F =Fuerza total

R_x = Matriz rotación eje x

R_y = Matriz rotación eje y

R = Matriz rotación tres ejes

K_f = Constante de empuje

r = vector posición

I = Matriz inercia

ξ =Vector velocidad en relación con $\{b\}$

H = Momento angular

L = Distancia centro de masa a centro de hélice

W = Vector de posición

T_q = Fuerza de arrastre relacionada con fuerza de empuje

Capítulo 1

1.1 Introducción

Un cuadricóptero es un vehículo aéreo no tripulado que actualmente es utilizado para diversos motivos desde recreacionales, investigación, bélicos, exploración, cultivo y pesca, principalmente en lugares donde el acceso humano es complicado por lo peligrosa que la tarea es y/o donde se requiere abracar grandes distancias con un método más sencillo y rápido.

Debido a los grandes usos de estos equipos es necesario que estos cuenten con sistemas que permitan realizar estas tareas, considerando que están expuestas a entornos donde hay interferencias climáticas donde no siempre se puede predecir, por lo que es muy importante diseñar sistemas de control que permitan a estos equipos operar de manera correcta.

El uso del controlador PID permite controlar procesos lineales e invariantes en el tiempo por lo que el modelo dinámico de un cuadricóptero debe ser linealizado y además que este descompuesto en cada variable a controlar debido a que los PID solo permiten una entrada al controlador.

Para cumplir con los objetivos de esta tesis de obtener un controlador PID fue necesario la profunda investigación del modelo matemático del cuadricóptero además de las leyes de control convencional. Para la obtención del controlador se utilizó la aplicación de optimización de MatLab por medio de algoritmos genéticos, la cual varía las ganancias del controlador y evalúa cuales tienen mejor desempeño en el sistema.

1.2. Antecedentes

El desarrollo de vehículos autónomos voladores por parte de la ingeniería humana lleva años con distintos fines de uso, desde bélicos ofensivos y defensivos hasta civiles, para tareas de alto riesgo donde el acceso humano sea imposible o se requiere un alto grado de precisión.

Gracia a los desarrollos industriales, sensores, equipos y el desarrollo de la ciencia de control han permitido mejorar los parámetros de estos y ampliar sus usos. Actualmente existen varios trabajos de modelado, simulación e implementación de cuadricópteros, donde se obtienen

resultados similares en cuanto a los controladores PID, donde en general se logra estabilizar pero no de la manera esperada, volviendo susceptibles a los vehículos a perturbaciones como es el caso de Ernesto Paiva (2016); que mediante las simplificaciones consideradas, obtuvo 4 sistemas SISO de un cuadricóptero, en simulación observó que los movimientos de traslación del cuadricóptero no se detienen debido a que no consideró los coeficientes de rozamiento con el aire. Logró mejorar la robustez del sistema añadiendo un término integral, sin embargo, el control no fue eficiente ya que se producen pequeñas oscilaciones no deseadas sobre el cuadricóptero. La maniobrabilidad y la robustez del cuadricóptero aumentan al cambiar a una configuración de estructura en equis, debido a que en esta configuración siempre trabajan los cuatro rotores para realizar movimientos de roll y de pitch. Alejandro Antona (2018); reportó obtener un modelado más detallado en comparación con anteriores a él del comportamiento dinámico del sistema y modeló la aceleración en cada grado de libertad en función de comandos. El modelo desarrollo no refleja fielmente la realidad sin embargo gracias al desarrollo de la red neuronal logró compensar los errores de modelado y ajustarse, a su vez desarrolló un algoritmo de entrenamiento on-line de la red neuronal para mejorar el control.

Verónica Ortiz & Pablo Pulla (S. F.); diseñaron para la Universidad de las Fuerzas Armadas ESPE un dron funcional con un peso 2052 gr funcional y capaz de cargar 1000 gr con un sistema PID con ganancias $K_p = 65$, $K_d = 45$ y $K_i = 5$ con respuesta de tiempo de retardo 0.1 s, tiempo levantamiento = 0.2 s y tiempo pico = 0.6 s, todas las pruebas que presentan las realizan en simulación.

Antonio Pico (2012); diseñó un controlador PID con evolución diferencial obteniendo resultados positivos, pero no del todo satisfactorios. El control de altura obtuvo dos distintos resultados, al momento de colocar el cuadricóptero horizontal el posicionamiento fue bueno, en cambio, al inclinar el cuadricóptero tuvo variaciones significativas, este debido a que no considero la inclinación para calibrar los sensores, otro problema es que el sistema de almacenamiento al descargarse pierde altura y requiere un reajuste de la constante de ganancia el cual no fue diseñado.

Luis Romero (2014); implementó un controlador PID y construyó su propio cuadricóptero donde los resultados del control permitieron estabilizar el sistema, pero no de manera muy optima debido a que reportó que presentaban fallos cuando las perturbaciones era muy

bruscas y/o se mantenían por mucho tiempo, además reporta que su prototipo aumentaba las perturbaciones al sistema debido a que los motores y las hélices vibraban de tal manera que los sensores no funcionaban de correctamente, además su sistema de comunicación presentaba un retardo significativo el cual deterioró el funcionamiento del controlador.

Michell P. Ehrlich & Luis F. Lupian (2015); implementaron un sistema de control para un cuadricóptero con el cual solo experimentaron en un lugar aislado, pero reportaron haber cumplido con la estabilización de giro y elevación, aun su desarrollo carece de pruebas en el exterior, además aportan la importancia del uso adecuado de sensores y calibración.

Luis Romero Paredes (2014); diseño un control PID para un cuadricóptero los resultados de las simulaciones fueron muy buenas en el caso del control de altura tiene una estabilización de menos de 1 segundo y un error en estado estable muy bajo, para el caso del controlador de Roll, Pitch y Yaw tiene un pequeño sobreimpulso menor al 10% y errores muy pequeños en estado estable. En los casos implementados en cuadricóptero, los ángulos Roll y Pitch el controlador no logra un control como las simulaciones y tarda aproximadamente 14s en llegar a un estado oscilatorio con un error menor al 20%, para el caso de control en Yaw logró un controlador muy efectivo que logra seguir la señal de entrada; en el caso de control de altura el sistema presenta oscilaciones del 12.5%, para casos con una entrada constante o una entrada que cambia. En el caso de tener varias entradas el sistema logra un control sobre él, pero mantiene pequeñas oscilaciones en los ángulos Roll y Yaw.

Antonio Pico Villalpando (2012); diseño un software de sintonización de PID por método de evolución diferencial, utilizando un cuadricóptero en funcionamiento para sintonizar el controlador en tiempo real logrando obtener un buen control, pero este tiene bastantes variaciones del punto de referencia cuando este se encuentra inclinado.

Eduardo Suarez, et al (2016); reportó un controlador multivariable sintonizado por método de mínimos cuadrados combinado con un regulador óptimo cuadrático demostrando mejores efectos del control en comparación de un controlador PID convencional, teniendo un error de 0.2562 por el criterio de error medio, donde los controles de Roll y Pitch tienen un error de 28% lo cual es suficiente para que el sistema opere de manera correcta. Además, comprobó que al cambiar parámetros al sistema como la longitud de los motores al centro del cuadricóptero y la masa el controlador, el algoritmo logra identificar correctamente el sistema y sigue al punto de referencia.

Paúl Ortiz Gonzales (2014); generó dos controladores uno PID y otros robusto H_{∞} , En el caso del PID este generó un sobreimpulso de 30%, teniendo oscilaciones, pero a los 10 s este logra obtener un estado estacionario, debido a que el modelo fue acotado en ángulos Roll, Pitch y Yaw a 0, por lo que el autor concluye que la estabilidad del cuadricóptero se vería comprometida por estos errores y el sistema estaría limitado al eje xy. En el caso del controlador robusto reporta una mejor respuesta en caso de incertidumbres, además que tiene una mejor respuesta en tiempo de establecimiento.

Jhon Bonilla Becerra (2019); Desarrolló un control robusto acoplado para un cuadricóptero donde el sistema con una sola entrada logra obtener un error en estado estable de 0, además de no tener ningún sobreimpulso en el caso de tener multientradas tiene pequeñas oscilaciones en las demás variables generando sobreimpulsos, que son amortiguados y atenuados completamente en menos de 1s.

Alejandro Antona Castañares (2018). Diseño un controlador por medio de redes neuronales con modelo Bayesiano en el cual logra mejorar la respuesta a trabajos previos del autor con controladores PID, donde errores de posición son de 0.1m, 0.05 m y 0.08 m en los ejes “x, y” y “z”, aunque también muestra en su trabajo que el sistema tiene variaciones en su orientación Roll, Pitch y Yaw al influir en dos variables a la vez.

1.3. Objetivo general

El objetivo de esta tesis es diseñar los controladores PID necesarios para controlar la dirección y altura de un cuadricóptero en simulación; utilizando la aplicación del software MatLab de optimización por medio de algoritmos genéticos para obtener las ganancias de los controladores y observar las respuestas del cuadricoptero para las ganancias diseñadas.

1.3.1. Objetivos Particulares

- Investigar el funcionamiento estático y dinámico de los drones cuadricópteros.
- Obtener el modelo matemático de un cuadricóptero.
- Simular el modelo del cuadricóptero en una función de MatLab
- Por medio de la aplicación de optimización de MatLab obtener las ganancias de los controladores que generen una mejor respuesta en orientación y altura en el cuadricóptero.

1.4. Justificación

El uso actual de drones es muy amplio de diversos sectores como industriales, servicios de protección, rescate y recreativos, los cuales, gracias a los avances actuales de producción, manufactura, desarrollo de microcontroladores y sensores de mayor resolución y precisión han permitido que se diversifiquen sus uso y capacidades, logrando producir de distintos tamaños y precios, siendo accesible varios de estos a cualquier persona. El desarrollo de los controladores ha permitido ampliar las aplicaciones de los cuadricópteros, pero también los fines de los cuadricópteros requiere que tengan controles que aseguren la respuesta ante las posibles perturbaciones, las cuales constantemente afectan la operación de los cuadricópteros, por lo que desarrollar controladores que aseguren el funcionamiento de los equipos es muy importante y para este desarrollo es conveniente contemplar las múltiples entradas y múltiple salidas que tienen los cuadricóptero

1.5. Metodología

Para el trabajo se requirió obtener los controles de altura, Roll, Pitch y Yaw de un cuadricóptero, debido a que los cuadricópteros son sistemas múltiples entradas y múltiple salida; este trabajo contempla la dinámica de las variables a controlar y como estas influyen en las otras.

Para el desarrollo de los controladores se realiza una investigación de las ecuaciones que rigen a un cuadricóptero en altura y ángulos de dirección, se obtiene la función de transferencia de los motores utilizados por un cuadricóptero comercial “*Adrucopter*”, que describe la fuerza ejercida por el motor. Con las ecuaciones dinámicas del cuadricóptero y función de transferencia del motor, se realiza una función en MatLab con la acción de control PID y se obtiene el índice de desempeño ISE (índice del error cuadrático). Con esta función desarrollada se sintoniza los controles en la aplicación de MatLab de optimización por medio de algoritmos genéticos (GA) donde por medio de iteraciones y evaluaciones del error la aplicación de Matlab calculada las ganancias de los controles que conseguían una mejor respuesta.

1.6. Contenido de la tesis

En el capítulo 1, se plantean los conceptos muy básicos de las unidades autónomas voladoras o también conocido cuadricóptero, en este capítulo se aborda generalidades, se plantean los objetivos que esta tesis tiene, además se plantea la importancia que debe de haber para un correcto diseño de controladores para cuadricópteros, y los pasos generales para el desarrollo de los objetivos.

En el capítulo 2, se abordan conceptos particulares de los cuadricópteros. La historia de los cuadricópteros como surgen y en qué dirección se han dirigido los esfuerzos de su desarrollo, además de los conceptos de modo de vuelo de estos muy importantes conceptos que se deben tomar en cuenta para el desarrollo de los controladores, y se describen las partes que conforman el cuadricóptero.

En el capítulo 3, se describe el modelo dinámico del cuadricóptero en dos matrices; la primera es la matriz traslacional que describa las aceleraciones en los ejes “ x ”, “ y ” y “ z ” y la segunda, es la matriz rotacional la cual expresa las aceleraciones en los ángulos Roll, Pitch y Yaw de un cuadricóptero.

En el capítulo 4, se definen los sistemas de control PID, los cuales por medio del error (diferencia del valor deseado con el valor actual) generan una señal de control para llevar al sistema al valor deseado, además se explica cómo los algoritmos genéticos trabajan para conseguir los valores del controlador PID.

En el capítulo 5, se muestran los resultados gráficos de simulación, de las diferentes ganancias obtenidas por medio del toolbox de MatLab de optimización, y además se muestran las respuestas de las ganancias elegidas como mejores.

En el capítulo 6, se concluye los resultados obtenidos y se aborda la comparación con otros trabajos similares.

Capítulo 2

Los vehículos aéreos no tripulados son también conocidos como **VANT** en español, o en inglés **UAV** (Unmanned Aerial Vehicle) como se muestra en la figura 1, y son básicamente aviones que no requieren de un piloto humano a bordo. Estos son controlados manualmente a través de un mando por un piloto en tierra, o de forma autónoma con su correspondiente programación gracias a la electrónica de la que dispone (García Muñoz, 2015).



Figura 1 Vehículos aéreos no tripulados

2.1. Historia de los drones (UAV)

Si bien los drones o **UAV** se han puesto de moda en los últimos tiempos debido a su bajo costo y facilidad de uso, la evolución de ellos data de un siglo y medio atrás. Los momentos claves para su desarrollo comienzan con el general chino Zhuge Liang (180–234 ad) que utilizó superficies de papel cerradas con el aire interior calentado mediante lámparas de aceite; de este modo los enemigos veían luces moviéndose por el aire que atribuían a fuerzas divinas, lo cual les hacía huir (Marquez, 2015). No fue hasta el 22 de agosto del año 1849, donde se retoma este desarrollo por tropas austrohúngaras al lanzar globos no pilotados cargados con 150kg de explosivos. Subsecuente a este hecho se siguió desarrollando de manera bélica durante más de un siglo hasta fechas actuales (Verdú, 2017) (Delgado, 2016). El primer UAV operativo fue el Kattering Bug en 1917, era capaz de alcanzar con cierta precisión blancos en un rango de 40 millas, gracias a su ingenioso sistema mecánico de control. (Marquez, 2015).

El abaratamiento en los costos de los equipos electrónicos, junto con la eliminación de costos excesivos en sensores, softwares, baterías, fuselajes, tamaño, etc., dio paso a la apertura del mercado de los drones (Oliver, S.F.).

2.2. Clasificación de los drones

Los drones se pueden clasificar de distintas maneras, una de ellas que consisten en el modo de operación de estos que abarca 2 categorías de drones (Pérez, 2019).

- Drones de ala fija. Tienen un diseño similar a la mayoría de las aeronaves donde la aerodinámica toma un papel muy importante; ya que sirve para que se genere una fuerza sustentadora y así permanecer en el aire (figura 2.a). Por ello están diseñados para realizar vuelos de altura a grandes velocidades, pero sin una gran capacidad de carga ni de maniobrabilidad. En estos drones, los motores se colocan de modo que el eje del motor coincida con la dirección en la que va el dron. Para maniobrar realizan cambios en la inclinación de aletas y así conseguir un empuje lateral para ir cambiando de dirección. En estos tipos de drones incluso usan motores de combustión para conseguir el empuje y así alcanzar mayores velocidades (Pérez, 2019).
- Drones de ala rotativa (multirotores, figura 2.b). Estos drones tienen la gran diferencia respecto a los de ala fija en la manera en la que se sustentan en el aire. Generan una sustentación a través de las hélices de sus rotores colocados en vertical. Existen subtipos dependiendo de la cantidad de rotores que lo formen (Pérez, 2019):
 - 3 motores: tricópteros.
 - 4 motores: cuadricópteros.
 - 6 motores: hexacópteros.
 - 8 motores: octacópteros.



Figura 2 tipos de drones a. dron de ala fija b. dron Hexa cuadricóptero

2.3. Ventajas y desventajas de los cuadricópteros

Los cuadricópteros poseen muchas importantes ventajas que les permiten tener un número elevado de aplicaciones. Entre tales ventajas se encuentran (R. Miranda, & Et al. 2018):

1. Capacidad para mantenerse suspendidos en el aire (Hovering). Esta capacidad les permite operar en ambientes pequeños y congestionados.
2. Capacidad de maniobrabilidad elevada en comparación a helicópteros convencionales.
3. Poseen una relación de peso empuje elevada.
4. Uso de propelas o hélices pequeñas para producir un empuje deseado. Esta propiedad reduce también sus problemas dinámicos y estructurales, además les permite obtener estructura mecánica simple.
5. Reciben daños menores por colisiones en comparación con un helicóptero.

También hay desventajas que se presentan cuando se emplea un cuadricóptero. Un ejemplo de las desventajas es su elevado consumo de energía. Este problema es de gran importancia porque reduce el tiempo de vuelo del vehículo (R. Miranda, & Et al. 2018).

Para eliminar esta desventaja; existen algunos trabajos que proponen apagar motores en momentos específicos, o bien utilizar algoritmos de optimización que permitan obtener un empuje óptimo. Además, se ha propuesto una metodología que permite reducir el consumo de energía mediante un proceso de sintonización de las ganancias de los controladores, así como una etapa de diseño de trayectorias (R. Miranda, & Et al. 2018).

Otra desventaja de los cuadricópteros tiene que ver con las ecuaciones diferenciales que lo describen dinámicamente. Específicamente se emplea dos ecuaciones diferenciales, la primera que describe su dinámica traslacional y la otra que modela la dinámica rotacional. Sin embargo, éstas poseen términos que describen acoplamientos no lineales complejos desde el punto de vista de control automático (R. Miranda, & Et al. 2018).

2.4. Aplicaciones de los drones

Dejando a un lado los usos bélicos; estos equipos ofrecen amplias posibilidades de aplicación al sector de la ingeniería civil como: inspecciones de infraestructura, investigación atmosférica, levantamientos topográficos, filmación de películas y fotografía deportiva,

cultivos de precisión, control de caza, localización de bancos de pesca, mantenimiento de parque eólicos e infraestructuras energéticas, control medioambiental, gestión de riesgos y desastres naturales, exploración geoló-minera, etc. (figura 3) (Rejado, 2015).

En los países más avanzados se están impulsando diferentes proyectos de investigación, que abarcan desde el ensayo de material aeronáutico en condiciones peligrosas de vuelo, viento, lluvia intensa, y demás, pasando por el desplazamiento y reparto de grandes cargas e incluso en personas en vuelos comerciales en vuelos no tripulados, hasta el desarrollo de tecnologías concretas como son las pilas de combustible de hidrógeno (Rejado, 2015).



Figura 3 Ejemplos de aplicaciones drones; a. fertilización b. exploración

2.5. Estructura y componentes básicos de UAV de ala rotativa

Las UAV se componen de diferentes componentes físicos básicos los cuales les proporciona rigidez, movilidad, control y suministro energético (Pérez, 2019).

2.5.1. Controlador de vuelo

El controlador de vuelo se puede considerar como el cerebro de un dron. Es el encargado de gestionar la potencia asignada a cada rotor para conseguir ciertos movimientos del dron o incluso estabilidad. Todo esto mediante modelos matemáticos ya implantados que modelan todo tipo de drones (Pérez, 2019).

2.5.2. Armazón o cuerpo

Consiste en la estructura rígida que le dará forma y consistencia al dron. En ella irán instalados todos los elementos por lo que deberá ser rígida pero ligera. El tipo de cuerpo determina la envergadura del dron, así como una idea del peso que lo constituirá (figura 4) (Pérez, 2019).



Figura 4 Armazón de un cuadricóptero

2.5.3. Motores

Los motores usados en este tipo de dron dependen mucho de la carga que tienen que levantar, así como la potencia instantánea que se le demanda. Hay dos tipos de motores usados para los drones: motores brushless y motores brushed (Pérez, 2019).

Los motores brushed o con escobilla, suelen usarse cada vez menos y sólo se usan para levantar cargas muy pequeñas. Esto es así ya que el constante régimen de giro del motor provoca grandes pérdidas en las escobillas en forma de calor, y así reduce su eficiencia y su vida útil. La única ventaja notable de los motores brushed consiste en que no necesita de un control electrónico para poder mover el rotor adecuadamente, pero con la tecnología de la electrónica actual es fácilmente solucionable.

Los motores brushless o sin escobilla, ofrecen unas prestaciones más eficientes y adecuadas al peso del dron. Dentro de los motores brushless, hay dos subtipos atendiendo a qué parte del motor realiza el giro: los inrunner y los outrunner. Los outrunner consiguen el movimiento girando una carcasa exterior en la que está alojada el eje rotatorio, mientras que los inrunner poseen un núcleo rotatorio que se encuentra dentro del motor. La principal diferencia en cuanto a prestaciones es la relación velocidad de giro con el torque de éste. Los inrunner pueden alcanzar regímenes de giro bastante mayores, pero a costa de un menor torque. Suelen usarse para drones tipo de ala fija ya que la propia aeronave se ayuda de la aerodinámica para sustentarse (Pérez, 2019).



Figura 5. Motor brushless y brushed

2.5.4. Variadores

Al estar tratando con motores brushless outrunners, es necesario el uso de un dispositivo electrónico capaz de controlar la velocidad del motor, este dispositivo es conocido como el ESC (Electronic Speed Controller) o variador (figura 6), aunque también pueden ser usados para aplicar un freno dinámico (Pérez, 2019).

El ESC se encuentra conectado tanto a la fuente como a la controladora, y a partir de los datos que reciba de la controladora, suministrará un voltaje u otro al motor. Cambiando el orden de la disposición de cables que poseen se puede conseguir que el motor gire en sentido contrario al estipulado. Los ESC manejan grandes cantidades de corriente, por lo que los fabricantes aconsejan elegir ESC que puedan manejar aproximadamente un 10% o 25% más de la corriente que el motor pueda consumir. Dicho esto, los ESC pueden tener la capacidad de manejar 10 amperios en modelos más pequeños o hasta 200 amperios en modelos grandes (Pérez, 2019).



Figura 6 ESC

2.5.5. Hélices

Las hélices se componen generalmente de un cubo al que van acopladas las aspas. Estas al interactuar con el aire generan mediante sustentación una fuerza tractora que hace que la

hélice se mueva en el fluido en alguna dirección. Las hélices requieren de un par de torsión en el cubo que las haga girar. Estas hélices son acopladas al motor para que se les proporcionara el par que requieren para moverse (figura 7) (Colonia, 2004).



Figura 7 Hélices para drones

2.5.6. Batería

Es el elemento que aporta la energía al sistema para poder funcionar (figura 8). En el mercado hay infinidad de tipos de batería atendiendo a especificaciones de voltaje y corriente (Pérez, 2019).



Figura 8 Batería LiPo para cuadricóptero

2.5.7. Soporte anti-vibraciones

Consiste en dos plataformas, una atornillada al cuerpo del dron, y otra en la que va sujeta el controlador de vuelo (figura 9). Entre las dos plataformas hay unos elementos de silicona que los une y absorbe todo tipo de vibraciones. Esto es beneficioso para la controladora ya que la vibración causada por el movimiento de los motores puede provocar una vibración en el cuerpo del dron. Dicha vibración puede acabar siendo transmitida por el cuerpo hasta la controladora y así interferir en las mediciones del acelerómetro y giroscopio instalados en la controladora (Pérez, 2019).



Figura 9 Soporte anti-vibraciones de cuadricóptero

2.5.8 Sensores

Los sensores son dispositivos capaces de obtener la información de una variable física y poder transmitir a variables eléctricas las cuales son leídas por el controlador de vuelo, estos sensores son muy importantes debido a que proporcionan información relevante del estado del dron, como la altitud, velocidad, aceleración, estado de la batería (Pérez, 2019).

2.6. Movimientos de cuadricóptero

El cuadricóptero tiene cuatro rotores: frontal (M1), lateral derecho (M4), posterior (M3) y lateral izquierdo (M2) (figura 10). Los rotores M1 y M3 giran en sentido contrario de las manecillas del reloj mientras que los otros dos en contra (E. Nieto & F. Vaca, 2020).

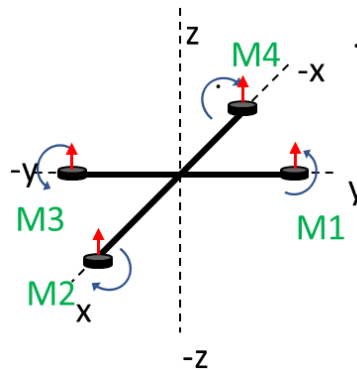


Figura 10 Representación de cuadricóptero

Los movimientos que tiene un cuadricóptero son de rotación y traslación, determinados a tres ejes fijos (X, Y, Z) y a través de la oscilación de los ángulos de navegación: Yaw (Ψ) en el eje Z, Pitch (θ) en el eje Y y Roll (Φ) en el eje X. El cuadricóptero tiene seis grados de libertad: tres traslacionales (x, y, z) y tres rotacionales (Yaw (Ψ), pitch (θ) y roll (Φ)) (E. Nieto & F. Vaca, 2020).

2.7 Rotación del cuadricóptero

2.7.1 Rotación angular- Roll Φ alabeo

Rotación sobre el eje x, se obtiene al reducir o aumentar las velocidades de uno de los rotores 2 o 4, y el contrario se aumenta o reduce, el par de rotores restantes permanece invariante en velocidad, al igual que el torque total. Se conoce como movimiento de alabeo, hacia la izquierda y derecha como se ve en la figura 2.11 (E. Nieto & F. Vaca, 2020).

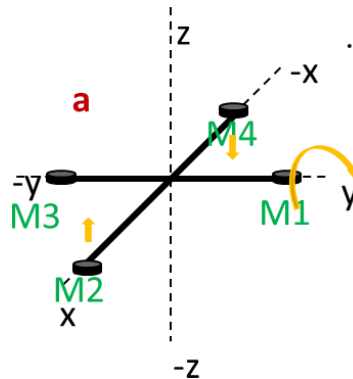


Figura 11 Giro roll

2.7.2. Rotación angular- Pitch θ cabeceo

La Rotación sobre el eje y, se obtiene al aumentar o reducir una de las velocidades de uno de los rotores 1 o 3, y el contrario se reduce o aumenta, el par de rotores restantes permanece invariante en velocidad, al igual que el torque total. Este es el movimiento de cabeceo Pitch, hacia adelante y atrás como se ve en la 2.12 (E. Nieto & F. Vaca, 2020).

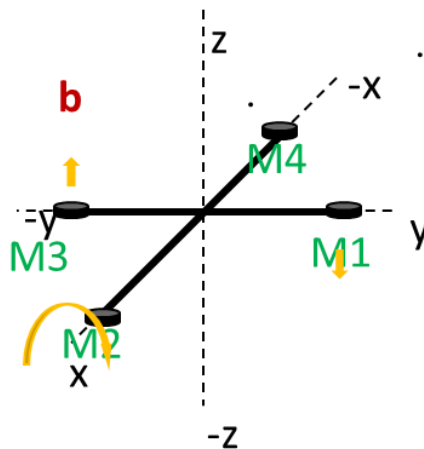


Figura 12 Giro pitch

2.7.3. Rotación angular- Yaw Ψ guiñada

Rotación sobre el eje z. Yaw Movimiento de Guiñada. La velocidad de los rotores 1 y 3 se aumenta o reduce, mientras que los rotores 2 y 4 se disminuyen o aumenta para mover la orientación. Esto produce un desbalance aerodinámico de los torques que permite que el cuadricóptero gire en su propio eje. Si todos los rotores mantienen la misma velocidad, el torque total será cero; y si además el empuje total de la aeronave es igual a su peso, esta se mantendrá en vuelo estacionario como se muestra en la figura 2.13 (E. Nieto & F. Vaca, 2020).

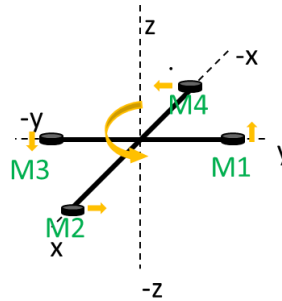


Figura 13 Giro Yaw

2.8. Movimiento traslacional

2.8.1. Traslación eje z

Al momento que los cuatro rotores estén girando a la misma velocidad, estos producen una fuerza en su centro de masa en el eje z, si esta fuerza es superior a la de la gravedad el vehículo provocando un desplazamiento positivo únicamente sobre el eje z como se muestra en la figura 2.14 (E. Nieto & F. Vaca, 2020).

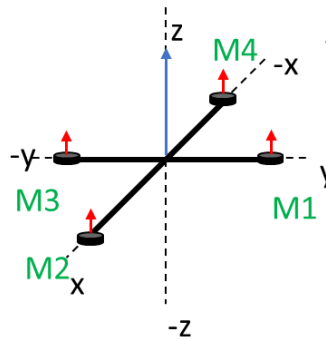


Figura 14 Desplazamiento en el eje z

2.8.2. Traslación en el eje x

Si los cuatros rotores se encuentran girando y el cuadricóptero esta suspendido en el aire y los rotores 2 y 4 tienen una velocidad distinta entre ellos mismos, la fuerza resultante en su centro de masa ira en dirección al eje z y al eje x provocando una inclinación del cuadricóptero (roll), provocando un desplazamiento en el eje x y z como se muestra en la figura 2.15 (E. Nieto & F. Vaca, 2020).

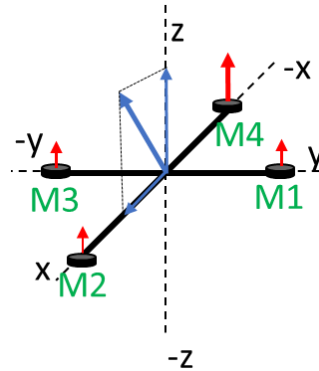


Figura 15 Desplazamiento en el eje x

2.8.3. Traslaciones en el eje y

Si los cuatros rotores se encuentran girando, y el cuadricóptero está suspendido en el aire y los rotores 1 y 3 tienen una velocidad distinta entre ellos mismos, la fuerza resultante en su centro de masa irá en dirección al eje z y al eje y provocando una inclinación del cuadricóptero (roll), provocando un desplazamiento en el eje y y z como se muestra en la figura 2.15 (E. Nieto & F. Vaca, 2020).

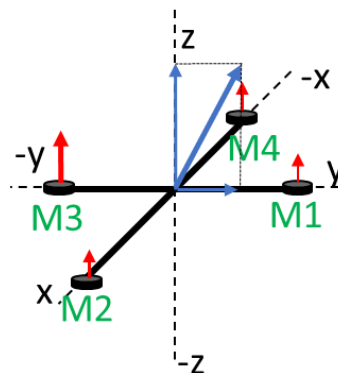


Figura 16 Desplazamiento en el eje x

Capítulo 3

3.1. Modelo matemático

Un sistema es un modelo de un dispositivo o de un conjunto de ellos existentes en el mundo real (sistema físico). En general, el estudio de sistemas físicos consta de cuatro partes: modelaje, descripción matemática, análisis y diseño. Para desarrollar el modelo de un sistema físico es necesario un profundo conocimiento de este y de los rangos de operación. Una vez obtenido el modelo, el paso siguiente es la descripción matemática la cual se obtiene utilizando leyes físicas (D. Giraldo y E. Giraldo, 2009).

Los modelos matemáticos son la interpretación de un sistema físico, químico, social o biológico que por medio de ecuaciones matemáticas describen el comportamiento de este sistema dependiendo de una o más variables de entrada.

En la visión determinista de la modelización matemática, presuponemos que el fenómeno en estudio lo rige una ley que determina las consecuencias a partir de las causas y es posible expresar esta ley en términos matemáticos (L. Cervantes, 2015).

3.1.1. Representación en Espacio de Estados

Un estado es un conjunto de variables más pequeño, de forma que, el conocimiento de esta variable junto con una entrada determina el comportamiento del sistema, el cual el sistema puede ser, físico, biológico, social y otros (K. Ogata, 2010).

Las variables de estado son las variables que constituyen en completo un sistema dinámico, donde existen n variables de estado dependiendo del sistema que describen a este. El espacio n -dimensional cuyos ejes de coordenadas están formados por el eje x_1 , eje x_2 , ..., eje x_n , donde $x_1, x_2, \dots, x_n$ son las variables de estado, se denomina espacio de estados. Cualquier estado se puede representar como un punto en el espacio de estados (K. Ogata, 2010).

3.1.2. Fases de la modelización matemática

Podemos describir el proceso de la modelización matemática a través de sus principales etapas (L. Cervantes, 2015):

- ✚ Estudio de la situación real: Para el estudio del sistema se requiere comprender el fenómeno por lo cual hay que tener énfasis en el conocimiento de las definiciones del sistema, identificar las preguntas que se desean conocer y las hipótesis subyacentes, revisar los datos que se tengan del sistema, identificar que variables se tienen como entrada y que variables son las que se obtienen y por último identificar las leyes conocidas del fenómeno desde el área del conocimiento (L. Cervantes, 2015).
- ✚ Elaboración del modelo matemático: En varios casos, es necesario reducir el sistema donde se debe identificar el fenómeno principal y separa los fenómenos involucrados para obtener el más sencillo y analizar sus componentes, en varios casos se deberá utilizar medios de solución numérica en lugar de solución analítica. El primer paso es la revisión de los datos que se tienen en caso de no tener datos, se realiza experimentos que proporcionen datos relevantes, identificar si los datos obtenidos pertenecen a una entrada o salida, identificar leyes conocidas, y elegir el modelo que describa el fenómeno (L. Cervantes, 2015).
- ✚ Solución del modelo: Los modelos pueden ser funciones o razones de cambio, en el caso del segundo hay que identificar el grupo que pertenece que puede ser; determinista o estocástico, discreto o continuo, de dimensión finita o infinita (L. Cervantes, 2015).
- ✚ Validación del modelo: Partiendo de las soluciones del modelo comparándola con la información del sistema en caso de tener buena coincidencia el modelo es aceptado o validado (L. Cervantes, 2015).

3.1.3. Protocolo para la elaboración de modelo matemático

1. Comprensión del fenómeno físico a modelar
2. Identificar variables independientes y dependientes
3. Excluir fenómenos que no sean de interés en el modelado
4. Identificar que leyes son aplicables para el modelo
5. Diseñar una ecuación diferencial que describa su comportamiento
6. Representar por medio de ecuaciones en espacio de estados la ecuación obtenida.
7. Validar el modelo físico.

3.2 Modelado matemático del cuadricóptero

Para la obtención del modelo matemático del cuadricóptero se emplean las leyes Newton-Euler, donde se obtienen dos modelos principales un sistema de referencia inercial $\{I\}$, y un sistema de referencia del cuadricóptero $\{B\}$ como se muestra en la figura 17.

El objetivo del control es mover el cuadricóptero por un plano perpendicular formado por los ángulos de los rotores, en este caso se puede variar el torque que presenta cada hélice, este genera una inclinación que dará dirección al movimiento del cuadricóptero.

3.2.1 Sistema de referencia

Para obtener las ecuaciones de la dinámica del sistema, es necesario expresar diversas fuerzas en el sistema de referencia inercial y uno en sistema de referencia del cuadricóptero. El cuadricóptero se mueve en el transcurso del tiempo, este cambia de posición respecto a un punto de referencia que se supone fijo. La posición del dron se referirá a través de dos puntos: uno fijo en tierra $\{I\} = \{a_1, a_2, a_3\}$, y el otro en el centro de masa del cuadricóptero $\{B\} = \{b_1, b_2, b_3\}$. La posición y traslación del cuadricóptero se describen mediante el movimiento de los ángulos Roll ϕ , Pitch θ y Yaw ψ , en el sistema de referencia B se emplean las ecuaciones dinámicas de rotación y traslación, este punto está en el centro del cuadricóptero y a él se refieren los giros y movimientos que realizan (E. Nieto & F. Vaca, 2020).

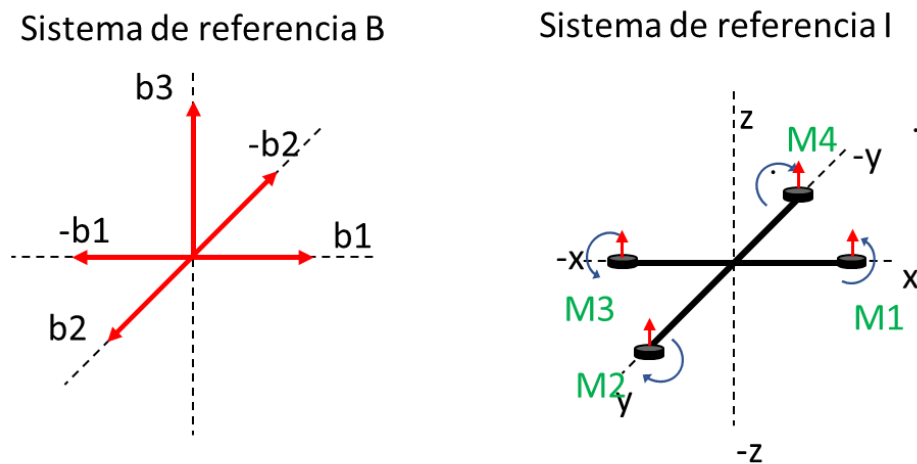


Figura 17 Sistemas de referencia

3.2.2 Ecuaciones de movimiento traslacional

En la figura 17 el sistema de referencia {B} se encuentra en el centro de masa. La ubicación de la masa del cuadricóptero y su orientación con respecto al sistema de referencia {I} define su configuración con respecto al sistema de referencia {B}.

Los rotores M1, M2, M3 y M4 se encuentran alineados con los vectores b_1 , b_2 , $-b_1$ y $-b_2$, respectivamente. Los rotores M1 y M3 giran alrededor del eje definido por b_3 en contra de las manecillas del reloj, mientras que los rotores 2 y 4 lo hacen alrededor del eje b_3 a favor de las manecillas del reloj, la orientación del cuadricóptero con respecto al eje {I} se obtiene por medio de la matriz de rotación R, la cual se conforma por los ángulos de Roll ϕ , Pitch θ y Yaw ψ , estos representan la rotación con respecto a los ejes x , y , y z respectivamente.

Para una más fácil interpretación se utilizan abreviaciones para las funciones trigonométricas para la función seno se utilizará solamente **S**, para la función coseno se utilizará solamente **C** y en caso de la función tangente **T**.

Para la rotación en el eje x

$$R_x(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & C\phi & -S\phi \\ 0 & S\phi & C\phi \end{pmatrix} \quad (3.1)$$

Para la rotación en el eje y

$$R_y(\theta) = \begin{pmatrix} C\theta & 0 & S\theta \\ 0 & 1 & 0 \\ -S\theta & 0 & C\theta \end{pmatrix} \quad (3.2)$$

Para la rotación en el eje z

$$R_z(\psi) = \begin{pmatrix} C\psi & -S\psi & 0 \\ S\psi & C\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3.3)$$

La matriz de rotación de cosenos directores del eje B respecto al eje inercial I viene dada por:

$$R = R(z, \psi), R(y, \theta), R(x, \phi)$$

$$R(\eta) = \begin{pmatrix} C\psi C\theta & C\psi s\theta - S\psi C\phi & C\psi S\theta c\phi + S\psi S\phi \\ S\psi C\theta & S\psi S\theta S\phi + C\psi C\phi & S\psi S\theta C\phi - C\theta C\psi S\phi \\ -S\theta & C\theta S\phi & C\theta C\phi \end{pmatrix} \quad (3.4)$$

Debido a la propiedad de ortogonalidad la matriz de rotación respecto al eje de coordenadas del cuerpo B es la transpuesta de R_1 y es:

$$R(\eta) = \begin{pmatrix} C\psi C\theta & S\psi C\theta & -S\theta \\ C\psi S\theta S\phi - S\psi C\theta & S\psi S\theta S\phi + C\psi C\phi & C\theta S\phi \\ C\psi S\theta C\phi + S\psi S\phi & S\psi S\theta S\phi - C\psi S\phi & C\theta C\phi \end{pmatrix} \quad (3.4)$$

Donde el vector η representa el vector de ángulos con respecto a “x”, “y” y “z”

$$\eta = [\phi, \theta, \psi]^T$$

Cada una de las hélices produce una fuerza de empuje F_i en la dirección de b_3 . La fuerza F_i se relaciona con la velocidad rotacional w_i de cada rotor, de acuerdo con la siguiente ecuación

$$F_i = c_t p A_{ri} r_i^2 w_i^2; \quad i = 1, 2, 3, 4$$

$$F_i = K_F w_i^2 \quad (3.5)$$

Donde A_{ri} [m²] es el área generada por la rotación de la hélice, r_i [m] es el radio, c_T (adimensional) es el coeficiente de empuje que depende de la hélice y ρ [Kg m⁻³] representa la densidad del aire, nótese que en 3.5 se expresan las constantes en una sola K_F [Kg m], donde $K_F > 0$.

Cuando el cuadricóptero se encuentra suspendido en una posición fija (hovering), el empuje total se calcula como.

$$F = F_1 + F_2 + F_3 + F_4$$

$$F = \sum_{i=1}^4 F_i = k_F \sum_{i=1}^4 w_i^2 \quad (3.6)$$

Donde se supone que cada rotor genera el mismo empuje.

De acuerdo con la segunda ley de Newton, se necesita una fuerza F para producir en ella una aceleración ($F = ma$). Ahora si se considera un sistema S conformado por N partículas, la aceleración del sistema S en $\{I\}$ es equivalente a la aceleración de un objeto puntual único de masa m igual al total de la masa S , ubicado en el centro de S .

El vector de posición del centro de masa con respecto al sistema $\{I\}$, se calcula empleando:

$$r = \sum_{i=1}^N m_i p_i \quad (3.7)$$

Donde m_i , p_i , representan la masa y el vector de posición de la i -ésima partícula, respectivamente, y r es el vector posición del centro de masa del vehículo. Por lo tanto, la velocidad lineal del centro de masa se relaciona con la fuerza F y con el momento lineal L_m :

$$F = m \frac{d}{dt} v$$

$$L_m = mv$$

La fuerza corresponde a su razón de cambio con el tiempo.

$$F = \frac{d}{dt} L_m$$

finalmente tomando los datos de R de la ecuación 3.4 que especifica la fuerza total en cada eje, r es el vector posición (x, y, z), y la definición de $F = F_1 + F_2 + F_3 + F_4$ de 3.6, se tiene entonces, la ecuación dinámica que describe la aceleración del centro de masa es en cada eje:

$$m\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + \begin{pmatrix} C\psi S\theta C\phi + S\psi S\phi \\ S\psi S\theta C\phi - C\psi S\phi \\ C\phi C\theta \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ F_1 + F_2 + F_3 + F_4 \end{pmatrix} \quad (3.8)$$

Las ecuaciones de movimiento traslacional se obtienen al descomponer el vector $\ddot{r}$ en sus componentes; despejando m .

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + \begin{pmatrix} C\psi S\theta C\phi + S\psi S\phi \\ S\psi S\theta C\phi - C\psi S\phi \\ C\phi C\theta \end{pmatrix} F \quad (3.8)$$

$$\ddot{x} = \frac{1}{m} (c\psi s\theta c\phi + s\psi s\phi) F \quad (3.9)$$

$$\ddot{y} = \frac{1}{m} (s\psi s\theta c\phi - c\psi s\phi) F \quad (3.10)$$

$$\ddot{z} = -g + \frac{1}{m} (c\theta c\phi) F \quad (3.11)$$

Donde 3.8 está en relaciona al sistema de referencia $\{I\}$

3.2.3 Ecuaciones cinemática rotacional

Para el análisis del movimiento rotacional se define el momento angular H de la siguiente manera.

$$H = I\xi$$

Donde $I \in \mathbb{R}^{3 \times 3}$ es el tensor de inercia y ξ representa el vector de velocidad angular de $\{B\}$ con respecto de $\{I\}$. El tensor inercial (I) se define a continuación:

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} [kgm^2]$$

Los términos I_{xx} , I_{yy} e I_{zz} corresponden a los momentos de inercia principales del cuadricóptero a los ejes x , y y z del sistema de referencia $\{B\}$.

De acuerdo con la ecuación de Newton-Euler, los pares externos $\tilde{\tau}_1 = [\tau_2, \tau_3, \tau_4]^T \in \mathbb{R}^3$ aplicados al cuadricóptero quedan:

$$\tilde{\tau}_1 = \frac{d}{dt}H = I\dot{\xi} + \xi \times I\xi \quad (3.12)$$

Las hélices producen un par debido al que producen los rotores, por lo que las hélices 1 y 3 producen un par M_1 y M_3 en dirección a b_3 , mientras que las hélices 2 y 4 producen un par M_2 y M_4 en dirección a $-b_3$.

Donde τ_2 representa la diferencia entre el par de los rotores izquierdo rotor 2 y derecho rotor 4, τ_3 representa la diferencia entre el par de los rotores delantero rotor 1 y trasero rotor 3, y τ_4 representa la diferencia entre los rotores girando en sentido a las manecillas del reloj y los que van en contra a las manecillas.

$$\begin{aligned} \tau_2 &= L[F_2 - F_4] \\ \tau_3 &= L[F_1 - F_3] \\ \tau_4 &= M_1 - M_2 + M_3 - M_4 \end{aligned} \quad (3.13)$$

El par generado en cada eje de rotación se calcula como:

$$M_i = k_m w_i^2 \quad (3.14)$$

Donde k_m es el coeficiente de arrastre que depende de la aceleración angular del i -ésimo rotor multiplicado por su momento de inercia, con la expresión 3.6, 3.13 y 3.14 se obtiene 3.15 que expresa todos los torques generados y la fuerza F todos en relación a la aceleración angular de cada rotor.

Empleando 3.6, 3.13 y 3.14, el empuje total del vector F se tiene de la siguiente manera:

$$\tau_0 = \begin{bmatrix} F \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} = \begin{bmatrix} k_f & k_f & k_f & k_f \\ 0 & Lk_f & 0 & -Lk_f \\ Lk_f & 0 & -Lk_f & 0 \\ k_m & -k_m & k_m & -k_m \end{bmatrix} \begin{bmatrix} w_1^2 \\ w_2^2 \\ w_3^2 \\ w_4^2 \end{bmatrix} \quad (3.15)$$

En el sistema de referencia $\{B\}$, las componentes de velocidad angular son:

$$\xi = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (3.16)$$

Los elementos p, q, r se relacionan con la derivada de los ángulos ϕ, θ y ψ , debido a que la matriz de rotación R se parametriza con los ángulos ϕ, θ y ψ por medio de las rotaciones x-y-z con base en esto se obtiene la siguiente relación.

$$R^T = R_{y,-\theta} R_{x,-\phi} R_{z,-\psi}$$

La ecuación notación $R_{a,b}$ se refiere un ángulo b, con respecto al eje a.

La ecuación R^T hace mención que para tener la orientación del sistema de referencia {I} con respecto a {B} primero se realiza una rotación en el eje y en un ángulo $-\phi$, es decir que $\dot{\phi}$ se encuentra en dirección de x, siguiendo el mismo razonamiento puede verificarse que $\dot{\theta}$ se encuentra en dirección del eje y para después realizar la rotación $[R_{x,-\phi}]$, por último, $\dot{\psi}$ se encuentra en dirección de z para después realizar $[R_{y,-\theta}]$, $[R_{x,-\theta}]$.

Por lo que los comportamientos del vector de velocidad angular del cuadricóptero con respecto al eje de referencia {B} se puede expresar de la siguiente manera.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \dot{\theta} i_2 + \dot{\phi} [R_{x,-\theta}] i_1 + \dot{\psi} [R_{x,-\theta}] [R_{y,-\theta}] i_3 \quad (3.17)$$

Donde R_y y R_x son las matrices de rotación 3.2 y 3.1 respectivamente, las cuales al sustituir en 3.15 se relacionan los vectores $\xi, \dot{\eta}$, de la siguiente manera:

$$\begin{aligned} \begin{bmatrix} p \\ q \\ r \end{bmatrix} &= \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\phi & S\phi \\ 0 & -S\phi & C\phi \end{bmatrix} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} C\theta & 0 & -S\theta \\ -S\theta S\phi & C\phi & C\theta S\phi \\ -C\phi S\theta & -S\phi & C\theta C\phi \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \\ \begin{bmatrix} p \\ q \\ r \end{bmatrix} &= \begin{bmatrix} 1 & 0 & -S\theta \\ 0 & C\phi & C\theta S\phi \\ 0 & -S\phi & C\theta C\phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \end{aligned} \quad (3.18)$$

Definiendo:

$$\begin{aligned} \xi &= \begin{bmatrix} p \\ q \\ r \end{bmatrix} \\ W &= \begin{bmatrix} 1 & 0 & -S\theta \\ 0 & C\phi & C\theta S\phi \\ 0 & -S\phi & C\theta C\phi \end{bmatrix} \\ \dot{\eta} &= \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \\ \xi &= W \dot{\eta} \end{aligned} \quad (3.18)$$

Empleando la ecuación 3.13, se puede reordenar:

$$I \begin{bmatrix} \dot{P} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} L[F_2 - F_4] \\ L[F_1 - F_3] \\ M_1 - M_2 + M_3 - M_4 \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} x I \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (3.19)$$

Al resolver las ecuaciones se tiene:

$$I \begin{bmatrix} \dot{P} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} -qr[I_{zz} - I_{yy}] \\ -pr[I_{zz} - I_{xx}] \\ pq[I_{xx} - I_{yy}] \end{bmatrix} - \begin{bmatrix} \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} \quad (3.20)$$

Donde se obtiene en la ecuación 3.20 un modelo dinámico en relación con el sistema de referencia {B}.

Para el diseño de controladores es importante que estén referenciados en el sistema de referencia deseado. En este caso es conveniente representar las matrices en relación al sistema de referencia {I}.

Las ecuaciones dinámicas relacionadas con su movimiento rotacional se expresarán utilizando los ángulos roll (ϕ), pitch (θ) y guiñada (ψ). Empleando 3.18

$$\xi = W\dot{n}$$

Por lo que empleando 3.12 quedando de la siguiente manera como indica Paul Ortiz 2017

$$IW\ddot{\eta} = -[W\dot{\eta} x IW\dot{\eta}] - IW\dot{\eta} + \begin{bmatrix} \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} \quad (3.21)$$

Resolviendo 3.21

$$\begin{aligned} IW &= \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} 1 & 0 & -S\theta \\ 0 & C\phi & C\theta S\phi \\ 0 & -S\phi & C\theta C\phi \end{bmatrix} = \begin{bmatrix} I_{xx} & 0 & -I_{xx}S\theta \\ 0 & I_{yy}C\phi & I_{yy}C\theta S\phi \\ 0 & -I_{zz}S\phi & I_{zz}C\theta C\phi \end{bmatrix} \\ W\dot{n} &= \begin{bmatrix} 1 & 0 & -S\theta \\ 0 & C\phi & C\theta S\phi \\ 0 & -S\phi & C\theta C\phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \dot{\phi} - \dot{\psi}S\theta \\ \dot{\theta}C\phi + \psi C\dot{\theta}S\phi \\ -\dot{\theta}S\phi + \dot{\psi}C\theta C\phi \end{bmatrix} \\ IW\dot{n} &= \begin{bmatrix} [\dot{\phi} - \dot{\psi}S\theta]I_{xx} \\ [\dot{\theta}C\phi + \psi C\dot{\theta}S\phi]I_{yy} \\ [-\dot{\theta}S\phi + \dot{\psi}C\theta C\phi]I_{zz} \end{bmatrix} \\ -[W\dot{\eta} x IW\dot{\eta}] &= - \begin{bmatrix} [\dot{\theta}C\phi + \psi C\dot{\theta}S\phi][-\dot{\theta}S\phi + \dot{\psi}C\theta C\phi][I_{yy} - I_{zz}] \\ [-\dot{\theta}S\phi + \dot{\psi}C\theta C\phi][\dot{\phi} - \dot{\psi}S\theta][I_{zz} - I_{xx}] \\ [\dot{\phi} - \dot{\psi}S\theta][\dot{\theta}C\phi + \psi C\dot{\theta}S\phi][I_{xx} - I_{yy}] \end{bmatrix} \end{aligned}$$

$$\begin{aligned}
-\dot{W}\dot{n} &= - \begin{bmatrix} 0 & 0 & -\dot{\theta}C\theta \\ 0 & -\dot{\phi}S\phi & -\dot{\theta}S\theta C\phi + \dot{\phi}C\theta C\phi \\ 0 & -\dot{\phi}C\phi & -\dot{\theta}S\theta C\phi - \dot{\phi}C\theta S\phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \\
-I\dot{W}\dot{n} &= - \begin{bmatrix} -I_{xx}\dot{\psi}\dot{\theta}C\theta \\ I_{yy} [-\dot{\theta}\dot{\phi}S\phi - \dot{\psi}\dot{\theta}S\theta C\phi + \dot{\psi}\dot{\phi}\dot{C}\theta C\phi] \\ I_{zz} [-\dot{\theta}\dot{\phi}C\phi - \dot{\psi}\dot{\theta}S\theta C\phi - \dot{\psi}\dot{\phi}C\theta S\phi] \end{bmatrix}
\end{aligned}$$

Donde se puede reescribir todo de la siguiente manera para tener un modelo dinámico en relación con $\{I\}$.

$$I_r\ddot{\eta} = \begin{bmatrix} K_1 \\ K_2 \\ K_3 \end{bmatrix} + \begin{bmatrix} \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} \quad (3.22)$$

Donde

$$\begin{aligned}
I_r &= \begin{bmatrix} I_{xx} & 0 & -I_{xx}S\theta \\ 0 & I_{yy}C\phi & I_{yy}C\theta S\phi \\ 0 & -I_{zz}S\phi & I_{zz}C\theta C\phi \end{bmatrix} \\
K_1 &= -[\dot{\theta}C\phi + \psi C\dot{\theta}S\phi][-\dot{\theta}S\phi + \psi C\theta C\phi][I_{yy} - I_{zz}] + I_{xx}\dot{\psi}\dot{\theta}C\theta \\
K_2 &= -[-\dot{\theta}S\phi + \psi C\theta C\phi][\dot{\phi} - \dot{\psi}S\theta][I_{zz} - I_{xx}] - I_{yy}[-\dot{\theta}\dot{\phi}S\phi - \dot{\psi}\dot{\theta}S\theta C\phi + \dot{\psi}\dot{\phi}\dot{C}\theta C\phi] \\
K_3 &= -[\dot{\phi} - \dot{\psi}S\theta][\dot{\theta}C\phi + \psi C\dot{\theta}S\phi][I_{xx} - I_{yy}] - I_{zz}[-\dot{\theta}\dot{\phi}C\phi - \dot{\psi}\dot{\theta}S\theta C\phi - \dot{\psi}\dot{\phi}C\theta S\phi]
\end{aligned}$$

Ahora con estas ecuaciones 3.8 y 3.20 representa el modelo en relación con relación a $\{I\}$.

Ahora se procede a obtener una matriz relacionadas con su movimiento rotacional, esta matriz es muy importante para el diseño de controladores, debido a que este movimiento rotacional es un factor que afecta en la trayectoria de un cuadricóptero.

La ecuación 3.21 se pre multiplica por un W^T lo que resulta:

$$W^T I W \ddot{\eta} = -W^T [W \dot{\eta} \times I W \dot{\eta}] - W^T I \dot{W} \dot{\eta} + W^T \begin{bmatrix} \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} \quad (3.23)$$

Definiendo los disiguientes términos:

$$\begin{aligned}
M &= W^T I W \\
C(\eta, \dot{\eta}) &= -W^T (W \dot{\eta}) I W - W^T I \dot{W} \\
\tau_o &= W^T \begin{bmatrix} \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix}
\end{aligned} \quad (3.24)$$

Empleando las definiciones de la ecuación 3.24 se obtiene la siguiente ecuación que describe su movimiento rotacional expresadas en {I} (Paul Ortiz 2017).

$$M(\eta)\ddot{\eta} + C(\eta, \dot{\eta})\dot{\eta} = \tau_0 \quad (3.25)$$

Es decir

$$M(\eta) = \begin{bmatrix} I_{xx} & 0 & -I_{xx}S\theta \\ 0 & I_{yy}C^2\phi + I_{zz}S^2\phi & (I_{yy} - I_{zz})C\phi S\phi C\theta \\ -I_{xx}S\theta & (I_{yy} - I_{zz})C\phi S\phi C\theta & I_{xx}S^2\theta + I_{yy}S^2\phi C^2\theta + I_{zz}C^2\phi C^2\theta \end{bmatrix} \quad (3.26)$$

Para cálculos de los componentes de $C_o(\eta, \dot{\eta})$, se emplea los símbolos de Christoffel.

$$C_{kj} = \begin{bmatrix} C_{1jk} \\ C_{2jk} \\ \vdots \\ C_{njk} \end{bmatrix}^T \dot{\eta} \quad (3.27)$$

$$C_{ijk} = \frac{1}{2} \left[\frac{dm_{kj}}{dq_i} + \frac{dm_{ki}}{dq_j} - \frac{dm_{ij}}{dq_k} \right]$$

la matriz $C(\eta, \dot{\eta})$ antisimétrica es (Paul Ortiz 2017):

$$C(\eta, \dot{\eta}) = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \quad (3.28)$$

Donde:

$$\begin{aligned} C_{11} &= 0 \\ C_{12} &= (I_{xx} - I_{zz})(\dot{\theta}c\theta s\phi + \dot{\psi}s^2\phi c\theta) + (I_{zz} - I_{yy})\dot{\psi}C^2\phi C\theta \\ &\quad - I_{xx}\dot{\psi}C\theta \\ C_{13} &= (I_{zz} - I_{yy})\dot{\psi}C\phi S\phi C^2\theta \\ C_{21} &= (I_{zz} - I_{yy})(\dot{\theta}c\phi s\phi + \dot{\psi}S^2\phi C\theta) + (I_{yy} - I_{zz})\dot{\psi}C^2\phi C\theta \\ &\quad + I_{xx}\dot{\psi}C\theta \\ C_{22} &= (I_{xx} - I_{yy})\dot{\phi}C\phi S\phi \\ C_{23} &= -I_{xx}\dot{\psi}S\theta C\theta + I_{yy}\dot{\psi}S^2\phi C\theta S\theta + I_{zz}\dot{\psi}C^2\phi S\theta C\theta \\ C_{31} &= (I_{yy} - I_{zz})\dot{\psi}C^2\theta S\phi C\phi - I_{xx}\dot{\theta}C\theta \\ C_{32} &= (I_{zz} - I_{yy})(\dot{\theta}C\phi S\phi S\theta + \dot{\phi}S^2\phi C\theta) \\ &\quad + (I_{yy} - I_{zz})\dot{\phi}C^2\phi C\theta + I_{xx}\dot{\psi}S\theta C\theta \\ &\quad - I_{yy}\dot{\psi}S^2\phi S\theta C\theta - I_{zz}\dot{\psi}C^2\phi S\theta C\theta \end{aligned}$$

$$C_{33} = (I_{yy} - I_{zz})\dot{\phi}c\phi s\phi c^2\theta - I_{yy}\dot{\theta}s^2\phi c\theta s\theta - I_{zz}\dot{\theta}c^2\phi c\theta s\theta + I_{xx}\dot{\theta}c\theta s\theta$$

3.3 Simplificación del sistema

Las no linealidades del sistema provocan una elevada complejidad para la identificación de la planta, por lo que, en muchos trabajos de investigación y revisión de literatura, se plantea la simplificación del modelo mediante ángulos pequeños es decir para $\phi = 0$ y $\theta = 0$ es decir $\cos(\phi) \approx \cos(\theta) \approx \cos(\psi) \approx 1$ y $\sin(\phi) \approx \sin(\theta) \approx \sin(\psi) \approx 0$, como se muestra a continuación:

$$M(\eta) = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (3.29)$$

$$C(\eta, \dot{\eta}) = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \quad (3.30)$$

Donde:

$$\begin{aligned} C_{11} &= 0 \\ C_{12} &= (I_{zz} - I_{yy})\dot{\psi} - I_{xx}\dot{\psi} \\ C_{13} &= 0 \\ C_{21} &= (I_{yy} - I_{zz})\dot{\psi} + I_{xx}\dot{\psi} \\ C_{22} &= 0 \\ C_{23} &= 0 \\ C_{31} &= -I_{xx}\dot{\theta} \\ C_{32} &= (I_{yy} - I_{zz})\dot{\phi} \\ C_{33} &= 0 \end{aligned}$$

Resolviendo 3.21 se tiene:

$$\begin{aligned} I_{xx}\ddot{\phi} + (I_{zz} - I_{yy})\dot{\psi}\dot{\theta} - I_{xx}\dot{\psi}\dot{\theta} &= \tau_{\phi} \\ I_{yy}\ddot{\theta} + (I_{yy} - I_{zz})\dot{\psi}\dot{\phi} + I_{xx}\dot{\psi}\dot{\phi} &= \tau_{\theta} \\ I_{zz}\ddot{\psi} + (I_{yy} - I_{zz})\dot{\phi}\dot{\theta} + I_{xx}\dot{\theta}\dot{\phi} &= \tau_{\psi} \end{aligned}$$

Despejando y rescribiendo se tiene las ecuaciones de movimiento rotacional:

$$\ddot{\phi} = \frac{(I_{yy} - I_{zz})}{I_{xx}} \dot{\theta} \dot{\psi} + \dot{\psi} \dot{\phi} + \frac{\tau_{\phi}}{I_{xx}} \quad (3.23)$$

$$\ddot{\theta} = \frac{(I_{zz} - I_{xx})}{I_{yy}} \dot{\psi} \dot{\phi} - \dot{\psi} \dot{\phi} + \frac{\tau_{\theta}}{I_{yy}} \quad (3.24)$$

$$\ddot{\psi} = \frac{(I_{xx} - I_{yy})}{I_{zz}} \dot{\phi} \dot{\theta} + \dot{\phi} \dot{\theta} + \frac{\tau_{\psi}}{I_{zz}} \quad (3.25)$$

3.4 Identificación del sistema

Para los parámetros del cuadricóptero se utilizaron los reportados por Paúl Ortiz en 2014 en el trabajo de “Diseño de Controlador PID y Robusto del Cuadricóptero “**Adrucopter**”, obteniendo los siguientes parámetros.

Tabla 1. Parámetros del cuadricóptero (extraído de Paúl Ortiz, 2014)

Parámetro		Valor	Unidad
<i>Motor</i>	m_m	77.6	g
<i>Hélice</i>	m_p	13.67	g
<i>Cuerpo</i>	m_g	1025.25	g
<i>Batería</i>	m_b	284.62	g
<i>Inercia eje x</i>	I_{xx}	0.0116	Kgm ²
<i>Inercia eje y</i>	I_{yy}	0.0116	Kgm ²
<i>Inercia eje z</i>	I_{zz}	0.0232	Kgm ²
<i>Fuerza de arrastre</i>	K_m	0.0636	S.U.
<i>Masa total</i>	m	1.6286	kg

Con los parámetros de la tabla 1 y la ecuación 3.25 se tiene la matriz $M(\eta)$ de la siguiente manera:

$$M(\eta) = \begin{bmatrix} 0.0116 & 0 & 0 \\ 0 & 0.0116 & 0 \\ 0 & 0 & 0.0232 \end{bmatrix} \text{Kgm}^2 \quad (3.26)$$

Los motores fueron caracterizados por Paúl Ortiz (2014), donde por medio de simulaciones numéricas obtuvieron una función de transferencia que relacione una entrada de voltaje y una salida en representación de fuerza (Newtons), teniendo la siguiente función de transferencia para cada uno los motores:

$$G(S) = \frac{0.975}{0.2s + 1} \quad (3.27)$$

Donde:

$$K = 0.975$$

$$\tau = 0.2$$

Capítulo 4

4.1 Sistemas de control

Los modelos de control permiten hacer el análisis cuantitativo de las respuestas (salidas) debidas a la aplicación de ciertas señales de entrada. Si la respuesta del sistema no es satisfactoria, el sistema debe ser mejorado u optimizado ya sea ajustando ciertos parámetros o en otros casos introduciendo compensadores. Un sistema de control es aquel cuyo fin es obtener varias respuestas deseadas a partir de ciertas entradas (D. Giraldo y E. Giraldo, 2009). Para un mejor diseño del controlador se pueden utilizar algún simulador que permita verificar el resultado del diseño del controlador sin llegar a afectar la planta. Algunos de los simuladores empleados pueden ser Simnon, y Matlab. Para este trabajo se utiliza la plataforma de simulación de Matlab que es una plataforma de programación y cálculo numérico utilizada por millones de ingenieros y científicos para analizar datos, desarrollar algoritmos y crear modelos. Combina un entorno de escritorio perfeccionado para el análisis iterativo y los procesos de diseño con un lenguaje de programación que expresa las matemáticas de matrices y vectores directamente (MatLab, 2021).

4.1.1 Control convencional

La teoría de control clásico o convencional surge en los años 1900-1950 donde los desarrollos teóricos y prácticos posicionan al control automático como una ciencia potencial. Los principales fundamentos en lo que se basa el control clásico son la prueba de estabilidad de Ruth-Hurwitz desarrollada en 1874, en 1932 Harry Nyquist propuso la solución de los sistemas con amplificación retroalimentados basada en la respuesta a la frecuencia, con ayuda de los desarrollos en la década de 1940 se desarrollan trabajos de análisis matemáticos impulsando la disciplina de control, en 1942 se genera la primer formula empírica para la solución de sintonizar las ganancias de controladores por parte de J. G. Ziegler y N. B. Nichols, en 1948 Walter R. Evans desarrolló el método del lugar de las raíces para determinar la estabilidad de los sistemas de una sola entrada, al uso de los métodos en el dominio S principalmente en el lugar geométrico de las raíces permite calcular con rapidez y exactitud los valores de interés del control (I. Lázaro, 2019).

4.1.2 Control PID

El algoritmo teórico elemental del controlador PID es:

$$u(t) = K \left[e(t) + \frac{1}{T} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right] \quad (4.1)$$

La señal de control resulta entonces igual a la suma de tres términos: el término **P** (que es proporcional al error), el término **I** (proporcional a la integral del error) y el término **D** (que es proporcional a la derivada del error). Los parámetros del controlador son la ganancia proporcional K , el tiempo de integración T_i y el tiempo de derivación T_d (L. Desbrough, R. Miller, 2002).

Los controladores PID probablemente sean los más usados en la industria, esto es debido a su relativa simplicidad y su desempeño satisfactorio en un amplio rango de procesos. Estos han venido evolucionando conforme al progreso de la tecnología y en la actualidad es implementado muy frecuentemente en forma digital. El éxito de los controladores PID también es debido al hecho de que usualmente representan el componente fundamental de sistemas de control más sofisticados, que pueden ser implementados cuando una ley de control básica no es suficiente para obtener el desempeño requerido. La ley de control PID consiste en aplicar correctamente la suma de tres tipos de acciones de control: una acción proporcional, una acción integral y una acción derivativa (Antonio Pico, 2012).

- **Acción proporcional**

La acción de control proporcional es proporcional al error de control actual de acuerdo con la expresión (Antonio Pico, 2012):

$$u(t) = K_p e(t) = K_p (r(t) - y(t)) \quad (4.2)$$

Donde k_p es la ganancia proporcional. El significado de la acción proporcional es bastante sencillo, ya que implementa la típica operación de incrementar la variable de control, cuando el error de control es grande (Antonio Pico, 2012).

La principal desventaja de usar un controlador proporcional puro es que este produce un error en estado estable (Antonio Pico, 2012).

- **Acción integral**

La acción de control integral es usada cuando se requiere que el controlador corrija el error en estado estable. Este se denota con la expresión (Antonio Pico, 2012):

$$u(t) = K_i \int_0^t e(\tau) d\tau \quad (4.3)$$

Donde k_i es la ganancia integral. La acción integral actúa sobre los valores pasados del error de control.

- **Acción derivativa**

Mientras la acción proporcional está basada en el valor actual del error de control y la acción integral está basada los valores pasados del error de control, la acción derivativa se basa en la predicción de los valores futuros del error de control. La ley derivativa de control puede expresarse como (Antonio Pico, 2012):

$$u(t) = K_d \frac{de(t)}{dt} \quad (4.4)$$

La acción derivativa tiene un gran potencial de mejorar el desempeño del controlador, ya que esta puede anticiparse a una tendencia incorrecta del error de control y actuar contra ello. Sin embargo, tiene algunos problemas críticos, como la sensibilidad al ruido, que la hace no ser adoptada en muchos casos prácticos (Antonio Pico, 2012).

4.2 Sintonización por algoritmos genéticos

Los Algoritmos Genéticos (AG) son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización. Están basados en el proceso genético de los organismos vivos. A lo largo de las generaciones, las poblaciones evolucionan en la naturaleza de acorde con los principios de la selección natural y la supervivencia de los más fuertes. El desarrollo de los AG, se debe en gran medida al profesor investigador John Holland de la Universidad de Michigan. A finales de la década de los 60 desarrolló una técnica que imitaba en su funcionamiento a la selección natural, aunque originalmente esta técnica recibió el nombre de “planes reproductivos”, a raíz de la publicación en 1975 de su libro “Adaptation in Natural and Artificial Systems” se conoce principalmente con el nombre de AG (Othon Colorado, Et al. 2018).

Básicamente, un AG consiste en una población de soluciones codificadas de forma similar a cromosomas; cada uno de estos cromosomas tendrá asociado un ajuste, valor de bondad o fitness, que cuantifica su validez como solución al problema; en función de este valor se le darán más o menos oportunidades de reproducción. Los AG tienen una fuerte base biológica,

en sus orígenes los algoritmos evolutivos consistieron en copiar procesos que tienen lugar en la selección natural, este concepto había sido introducido, por Charles Darwin. A pesar de que aún hoy en día no todos los detalles de la evolución biológica son completamente conocidos existen algunos hechos apoyados sobre una fuerte evidencia experimental. Algunas de las alternativas para el cálculo de las constantes del controlador PID son las técnicas de computación emergentes entre las cuales están los AG (Othon Colorado, Et al. 2018).

La idea fundamental de los algoritmos genéticos consiste en encontrar una solución aceptable a un problema por medio del mejoramiento de un conjunto de individuos, cuya función de evaluación corresponde a una solución del problema. Esta optimización se realiza mediante procesos selectivos y de intercambio de información genética (Othon Colorado, Et al. 2018).

4.3 Ciclo de evolución

En el algoritmo de AG existe la presencia de un lazo con un criterio de detención que es el encargado de poner fin a la evolución de la población. Cada ciclo representa una nueva generación. Este ciclo está controlado por el criterio de detención adoptado. Existen varios criterios de detención (Luis Hernández, S. F.):

- Detener después de N iteraciones
- Detener cuando el fitness medio de la población cae dentro de una banda predefinida.
- Detener cuando la diversidad poblacional es baja
- Otros

Dentro del ciclo de evolución pueden observarse tres acciones concretas (Luis Hernández, S. F.):

- Seleccionar
- Cruzar
- Mutar

La selección implica determinar cuáles cromosomas serán considerados para procrear y así formar la siguiente generación (Luis Hernández, S. F.).

4.3.1 Función de evaluación

Para el correcto funcionamiento de un Algoritmo Genético se debe de poseer un método que indique si los individuos de la población representan o no buenas soluciones al problema planteado. Por lo tanto, para cada tipo de problema que se desee resolver debería derivarse un nuevo método, al igual que ocurrirá con la propia codificación de los individuos. De esto se encarga la función de evaluación, que establece una medida numérica de la bondad de una solución. Esta medida recibe el nombre de ajuste. En la naturaleza el ajuste (o adecuación) de un individuo puede considerarse como la probabilidad de que ese individuo sobreviva hasta la edad de reproducción y se reproduzca. Esta probabilidad debería estar ponderada con el número de descendientes. En el mundo de los Algoritmos Genéticos se emplearía esta medición para controlar la aplicación de los operadores genéticos. Es decir, permite controlar el número de selecciones, cruces, copias y mutaciones llevadas a cabo. La aproximación más común consiste en crear explícitamente una medida de ajuste para cada individuo de la población. A cada uno de los individuos se les asigna un valor de ajuste escalar por medio de un procedimiento de evaluación bien definido este procedimiento de evaluación sería específico del dominio del problema en el que se aplica el Algoritmo Genético. También puede calcularse el ajuste mediante una manera “coevolutiva”. Por ejemplo, el ajuste de una estrategia de juego se determina aplicando esa estrategia contra la población entera (o en su defecto una muestra) de estrategias de oposición (Marcos Gestal, 2013).

- **Fitness puro:** $r(i,t)$

Es la medida de ajuste establecida en la terminología natural del propio problema. La ecuación 4.4 establece el cálculo del valor de bondad de un individuo i en un instante t (generación) (Marcos Gestal, 2013).

$$r(i, t) = \sum_{j=1}^{N_c} |s(i, j) - c(i, j)| \quad (4.4)$$

Donde:

$s(i, j)$ = valor deseado para el individuo i en el caso de j

$c(i, j)$ = valor obtenido por el individuo i para el caso j

N_c = Número de caso

- **Fitness estandarizado:** $s(i, t)$

Para solucionar esta dualidad ante problemas de minimización o maximización se modifica el ajuste puro de acuerdo con la ecuación 4.5 (Marcos Gestal, 2013).

$$s(i, t) = \begin{cases} r(i, t) & \text{minimización} \\ r_{\max} - r(i, t) & \text{maximización} \end{cases} \quad (4.5)$$

En el caso de problemas de minimización se emplea directamente la medida de fitness puro. Si el problema es de maximización se resta de una cota superior $r_{\max}$ del error el fitness puro. Empleando esta métrica, la bondad de un individuo será mayor cuanto más cercano esté a cero el valor del ajuste. Por lo tanto, dentro de la generación t , un individuo i siempre será mejor que uno j si se verifica que $s(i, t) < s(j, t)$ (Marcos Gestal, 2013).

- **Fitness ajustado:** $a(i, t)$

Se obtiene aplicando la transformación reflejada en la ecuación 4.6 al fitness estandarizado (Marcos Gestal, 2013).

$$a(i, t) = \frac{1}{1 + s(i, t)} \quad (4.6)$$

De esta manera, el fitness ajustado tomará siempre valores del intervalo $[0 - 1]$. Cuando más se aproxime el fitness ajustado de un individuo a 1 mayor será su bondad (Marcos Gestal, 2013).

- **Fitness Normalizado:** $n(i, t)$

Los diferentes tipos de fitness vistos hasta ahora hacen referencia únicamente a la bondad del individuo en cuestión. El fitness normalizado introduce un nuevo aspecto: indica la bondad de una solución con respecto al resto de soluciones representadas en la población. Considerando una población de tamaño N , se obtiene según la ecuación 4.7 (Marcos Gestal, 2013).

$$n(i, t) = \frac{a(i, t)}{\sum_{k=1}^N a(k, t)} \quad (4.7)$$

Al igual que el fitness ajustado, siempre tomará valores del intervalo $[0 - 1]$, con mejores individuos cuanto más próximo esté a la unidad. Pero a diferencia de antes, un valor cercano a 1 no sólo indica que ese individuo represente una buena solución al problema, sino que, además, es una solución destacadamente mejor que las proporcionadas por el resto de la

población. La suma de los valores del fitness normalizado de todos los individuos de una población dará siempre 1 (Marcos Gestal, 2013).

4.3.2 Parámetros de los algoritmos genéticos

Una premisa es conseguir que el tamaño de la población sea lo suficientemente grande para garantizar la diversidad de soluciones. Se aconseja que la población sea generada de forma aleatoria para obtener dicha diversidad. En caso de que la población no sea generada de forma aleatoria habrá que tener en cuenta que se garantice una cierta diversidad en la población generada. Para el estudio de los algoritmos genéticos hay que tener en cuenta una serie de parámetros (Jorge Arranz & Antonio Parra. S. F.).

- **Tamaño de la población**

Este parámetro indica el número de cromosomas que tiene la población para una generación determinada. En caso de que esta medida sea insuficiente, el algoritmo genético tiene pocas posibilidades de realizar reproducciones con lo que se realizaría una búsqueda de soluciones escasa y poco óptima. Por otro lado, si la población es excesiva, el algoritmo genético será excesivamente lento (Jorge Arranz & Antonio Parra. S. F.).

- **Probabilidad de cruce**

Indica la frecuencia con la que se producen cruces entre los cromosomas padre, es decir, que haya probabilidad de reproducción entre ellos. En caso de que no exista probabilidad de reproducción, los hijos serán copias exactas de los padres. En caso de haberla, los hijos tendrán partes de los cromosomas de los padres. Si la probabilidad de cruce es del 100% el hijo se crea totalmente por cruce, no por partes (Jorge Arranz & Antonio Parra, S. F.).

- **Reemplazo**

Cuando en vez de trabajar con una población temporal se hace con una única población, sobre la que se realizan las selecciones e inserciones, debería tenerse en cuenta que para insertar un nuevo individuo debería eliminarse previamente otro de la población. Existen diferentes métodos de reemplazo (Marcos Gestal, 2013):

- a) Aleatorio: el nuevo individuo se inserta en un lugar cualquiera de la población.
- b) Reemplazo de padres: se obtiene espacio para la nueva descendencia liberando el espacio ocupado por los padres.

- c) Reemplazo de similares: una vez obtenido el ajuste de la descendencia se selecciona un grupo de individuos (entre seis y diez) de la población con un ajuste similar. Se reemplazan aleatoriamente los que sean necesarios.
- d) Reemplazo de los peores: de entre un porcentaje de los peores individuos de la población se seleccionan aleatoriamente los necesarios para dejar sitio a la descendencia.

- **Copia**

La copia es la otra estrategia reproductiva para la obtención de una nueva generación a partir de la anterior. A diferencia del cruce, se trata de una estrategia de reproducción asexual. Consiste simplemente en la copia de un individuo en la nueva generación. El porcentaje de copias de una generación a la siguiente es relativamente reducido, pues en caso contrario se corre el riesgo de una convergencia prematura de la población hacia ese individuo. De esta manera el tamaño efectivo de la población se reduciría notablemente y la búsqueda en el espacio del problema se focalizaría en el entorno de ese individuo. Lo que generalmente se suele hacer es seleccionar dos individuos para el cruce, y si este finalmente no tiene lugar, se insertan en la siguiente generación los individuos seleccionados (Marcos Gestal, 2013).

- **Probabilidad de mutación**

Nos indica la frecuencia con la que los genes de un cromosoma son mutados. Si no hay mutación, los descendientes son los mismos que había tras la reproducción. En caso de que haya mutaciones, parte del cromosoma descendiente es modificado y si la probabilidad de mutación es del 100%, la totalidad del cromosoma se cambia. En este caso, no se cambian simplemente unos bits del cromosoma, sino que se cambian todos, lo que significa que se produce una inversión en el cromosoma y no una mutación por lo que la población degenera muy rápidamente (Jorge Arranz & Antonio Parra. S. F.).

4.4 Diseño de controlador PID

Para el diseño de los controladores del cuadricóptero se requieren varias etapas como se muestran en la Figura 18. La primer etapa o lazo externo controla la posición deseada en los ejes “ x , y ” y “ z ”, este lazo genera una segunda señal de entrada a los controladores de altura y orientación. La segunda etapa o lazo de altitud, tiene como entrada la altura deseada o Z_d que genera la señal de salida fuerza F que va hacia los motores. La tercera etapa o lazo

interno, en el lazo de control de orientación; y obtiene como entradas de referencia los ángulos deseados de Roll (ϕ), Pitch (θ) y ángulo yaw (ψ) y genera las acciones de control de τ_2 , τ_3 y τ_4 correspondiente a los pares de los ejes para entrar al bloque de cuadricóptero que es el modelo matemático dinámico del mismo dado por las ecuaciones 3.23, 3.24 y 3.25 los cuales generan las señales n y $\dot{n}$ que sirve como retroalimentación de control de orientación y además con estas señales se calcula por medio de la ecuación 3.8 las aceleraciones traslacionales que se requieren para obtener el vector r de posición.

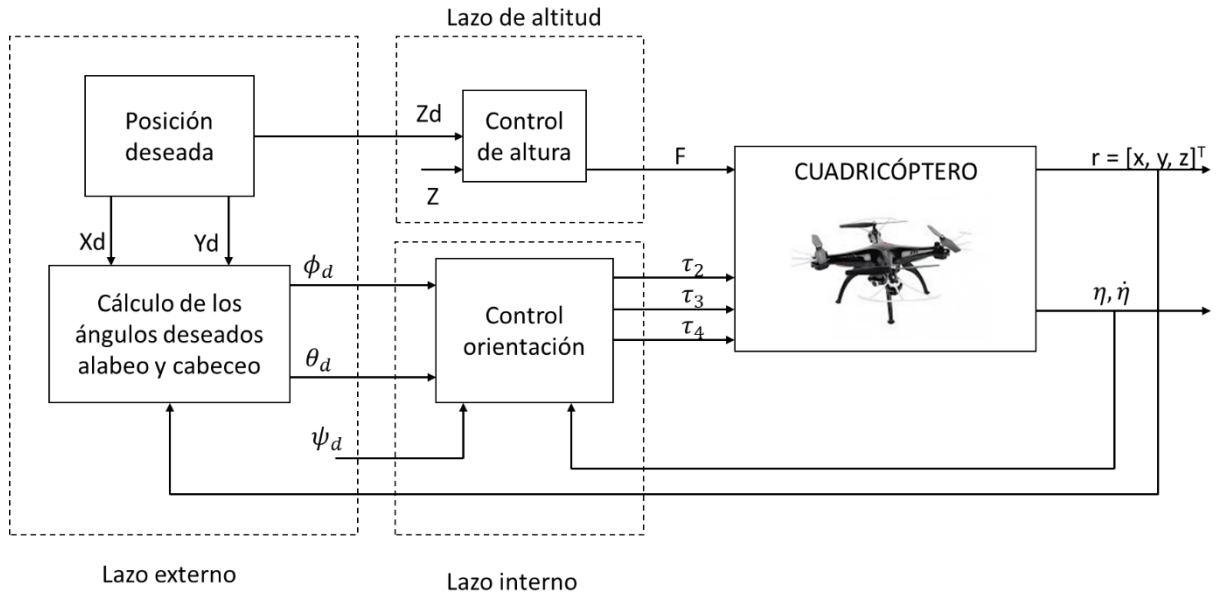


Figura 18 Diagrama de bloques del cuadricóptero

4.2.1 Diseño del control de altura

Para el diseño del control de altura se retoma la ecuación 3.11 y se plantea la siguiente ecuación para la señal de control F :

$$F(t) = K_h e_h(t) + K_{ih} \int_0^t e_h(\tau) d\tau + K_{dh} \frac{de_h(t)}{dt} + g \quad (4.1)$$

Donde a la señal de control PID se le agrega el valor de la gravedad (g) debido a que este controlador es el desplazamiento en el eje z el cual se encuentra bajo el efecto de la gravedad.

4.2.2 Diseño del controlador de orientación (ángulos roll ϕ , pitch θ y yaw ψ).

De acuerdo con las ecuaciones 3.23, 3.24 y 3.25 se aplicarán para el diseño de los controladores para los ángulos Roll, Pitch y Yaw respectivamente, las fórmulas de la señal de control quedan de la siguiente manera.

$$T_{\phi}(t) = K_{pr}e_r(t) + K_{it} \int_0^t e_r(\tau)d\tau + K_{dr} \frac{de_r(t)}{dt} \quad (4.2)$$

$$T_{\theta}(t) = K_{pp}e_p(t) + K_{ip} \int_0^t e_p(\tau)d\tau + K_{dp} \frac{de_p(t)}{dt} \quad (4.3)$$

$$T_{\psi}(t) = K_{py}e_y(t) + K_{iy} \int_0^t e_y(\tau)d\tau + K_{dy} \frac{de_y(t)}{dt} \quad (4.4)$$

4.2.3 Representación de los actuadores

Para la representación de los actuadores motor 1, 2, 3 y 4 del sistema se utilizó la función de transferencia obtenida por Paúl Ortiz (2014) ecuación 3.27, donde representa la fuerza que genera los motores.

Cada motor genera una fuerza, la combinación de estas fuerzas genera los movimientos del cuadricóptero, hay 4 motores, el motor 1 se encuentra en el vector b1, el motor 2 en el vector b2, el motor 3 en el vector -b1 y por último el motor 4 ubicado en el vector -b2 como se muestra en la figura 17.

Todos los motores ejercen una fuerza en el eje b3 para generar un desplazamiento en el eje z se requiere que los cuatro motores generen una fuerza superior a la de la gravedad (figura 14).

Para que se genere un ángulo roll los motores 2 y 4 deben cambiar la fuerza ejercida por ambos, si el torque generado del motor 4 al centro del cuadricóptero es mayor que el torque del motor 2 al centro del cuadricóptero se generará un ángulo roll positivo en el cuadricóptero cambiando el vector de fuerza F donde además de proporcionar una fuerza en el eje b3 habrá un desplazamiento en el eje b1 (figura 11).

Para que se genere un ángulo pitch los motores 1 y 3 deben cambiar la fuerza ejercida por ambos, si el torque generado del motor 3 al centro del cuadricóptero es mayor que el torque del motor 1 al centro del cuadricóptero se generará un ángulo pitch positivo en el

cuadricóptero cambiando el vector de fuerza F donde además de proporcionar una fuerza en el eje b_3 habrá un desplazamiento en el eje b_2 (figura 12).

Para generar un ángulo yaw se requiere que los motores 1 y 3 que giran en sentido contrario a las manecillas generando una fuerza de empuje sea menor a la fuerza de empuje producida por los motores 2 y 4, este cambio no afectaría el vector de fuerza F , pero si la posición de los motores (figura 13).

Así, las señales de control U_1 , U_2 , U_3 y U_4 de los motores 1, 2, 3 y 4 quedan de la siguiente manera.

$$U_1 = \frac{F}{4} - T_\theta - T_\psi \quad (4.5)$$

$$U_2 = \frac{F}{4} - T_\phi + T_\psi \quad (4.6)$$

$$U_3 = \frac{F}{4} + T_\theta - T_\psi \quad (4.7)$$

$$U_4 = \frac{F}{4} + T_\phi + T_\psi \quad (4.8)$$

4.3 Sintonización PID

Para la sintonización de las ganancias del controlador PID por medio de algoritmos genéticos (GA) se requiere una función que tenga como entrada el vector de las ganancias del controlador a sintonizar y se obtenga de salida un índice de desempeño (**Anexo1**), debido a que la toolbox solo puede trabajar con un solo índice de desempeño solo se puede sintonizar un controlador a la vez.

El índice de desempeño utilizado es el criterio de la integral del error al cuadrado, el cual su fórmula es la siguiente

$$j = \int_0^T e^2(t) dt \quad (4.6)$$

Donde

$$e = \text{error}$$

$$j = \text{índice de desempeño}$$

Este índice fue agregado en la función de MatLab, en cada iteración calcula el error, este error se utiliza por medio de un método de integración numérico de regla trapezoidal se

obtiene su valor constantemente, el resultado de este método de integración es el que regresa la función a la toolbox y toma como criterio de evaluación para el desarrollo del método de optimización.

4.3.1 Herramienta de optimización de MatLab

En el apéndice 1 se detalla el funcionamiento del toolbox, y su metodología según el usuario JL Mendoza-Soto (2020), de acuerdo con el canal Youtube.

En la figura 19 se muestra la aplicación de la utilería de Optimización de Matlab, donde en la opción **Solver** se selecciona Algoritmos Genéticos; en la opción **Fitness Function** se coloca el vector de ganancias de controlador (k_a), el nombre de la función (tesis) donde recibe el vector de ganancias además el código generada permite elegir que controlador sintonizar, por eso para este código tiene la opción de elegirlo donde puede tomar 4 valores: 1, 2, 3 y 4 donde corresponde a control altura, control roll, control pitch y control yaw ($k_a,1$); en la opción **Number of variables** se pone el número de variables que depende el sistema ya que en este caso se sintoniza un controlador PID son 3; además se coloca un rango de ganancias (máxima y mínima) que se pueden obtener con el algoritmo colocadas en el recuadro de **Bounds** esto con el fin de observar las respuestas que genera con diferentes ganancias en este caso de 1 a 100,

Además, se pueden añadir parámetros para el caso de tener variables no lineales las cuales para este trabajo no se utilizaron.

Para los parámetros del algoritmo que se utiliza los predeterminados por la aplicación.

En las simulaciones las entradas se programan desde la función realizada para el caso de altura solamente se le aplicó una entrada escalón unitario (1m) al instante $t = 0$ s. Par el caso de los ángulos roll y pitch primero el sistema tendrá una entrada escalón unitaria al instante $t = 0$ s en altura con las ganancias del controlador obtenido de altura y después al instante $t = 4$ s se aplicó una entrada de 10° (0.1745 rad).

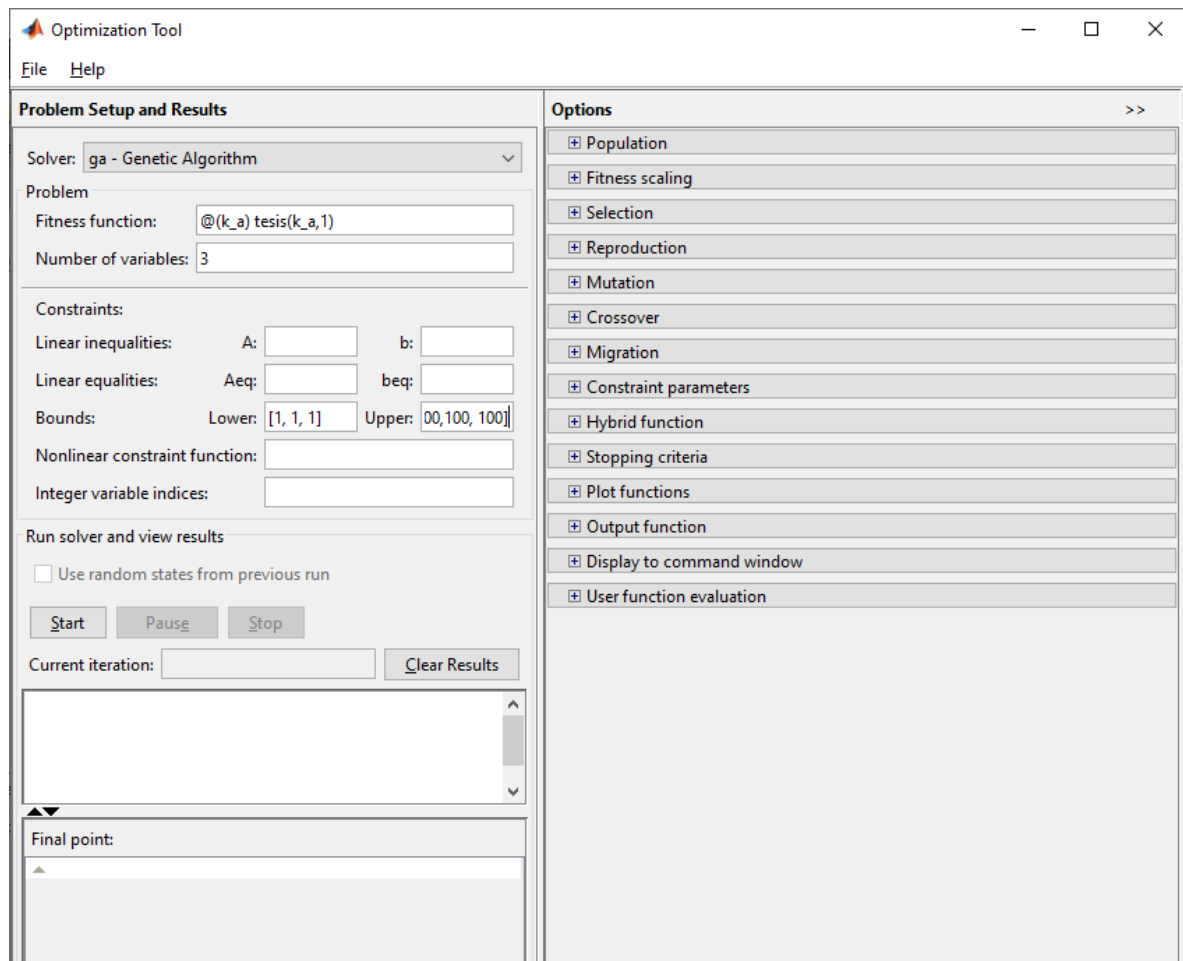


Figura 19 Toolbox MatLab

Capítulo 5

5.1 Resultados control Altura

Para las pruebas de altura se utiliza una referencia de 1m en el tiempo de 0 segundos y la simulación dura 8 segundos en total, además el intervalo de ganancias se varia en varios rangos (tabla 2), estos valores fueron acotados con el fin de ver cuánto mejora la ganancia en función con respecto a las ganancias de los controladores. Par el caso de los ángulos Roll y Pitch primero el sistema tendrá una entrada escalón unitaria al instante de tiempo de 0 segundos en altura con las ganancias del controlador obtenido de altura y después al instante de tiempo de 4 segundos se aplicó una entrada de 10° (0.1745 rad).

Los resultados obtenidos por la aplicación de MatLab se muestran en la tabla 4, en la columna Kp, Ki y Kd son las ganancias de la parte proporcional, integral y derivativa calculadas como la mejores por medio del algoritmo genético, el índice de desempeño es el valor numérico obtenido del análisis del error por medio de la ecuación 4.6, que fue agregada en la función para el uso del toolbox (Anexo 1), con las ganancias de los controladores marcados en la fila correspondiente.

Tabla 2 Resultados de ganancias de control de altura

N.	Intervalo de ganancias	Kp	Ki	Kd	Índice desempeño
1	0 – 100	63.502	19.364	20.63	0.3701
2	0 - 25	24.948	16.556	11.057	0.3713
3	0 -20	19.917	16.307	9.421	0.3781
4	0-18	17.912	15.623	8.513	0.3838
5	0 -15	14.88	14.956	10.096	0.4488

En las 20, 21, 22, 23 y 24 se muestra el comportamiento del cuadricóptero con los controladores 1, 2 3, 4 y 5 respectivamente mostrados en la tabla 2, todo con una señal de entrada 1m al instante 0, todas las simulaciones se realizaron con 8 segundos simulación y bajo los mismos parámetros del cuadricóptero.

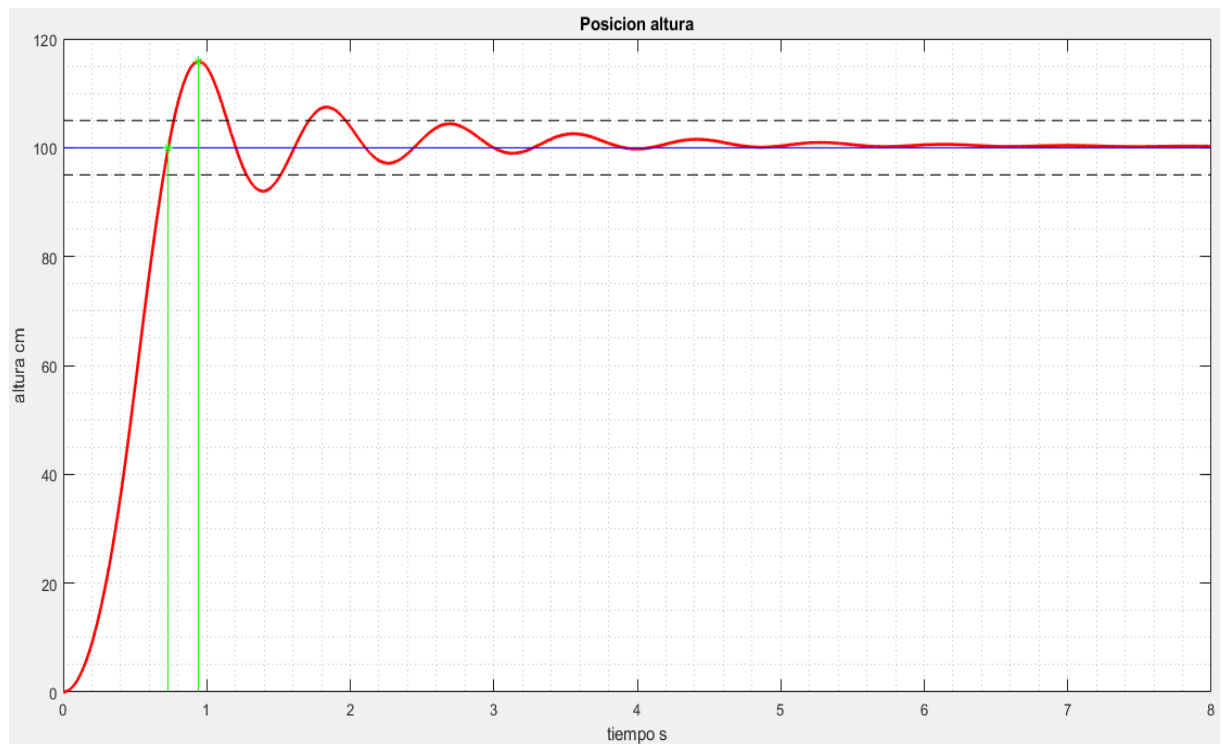


Figura 20. Altura del cuadricóptero con controlador 1

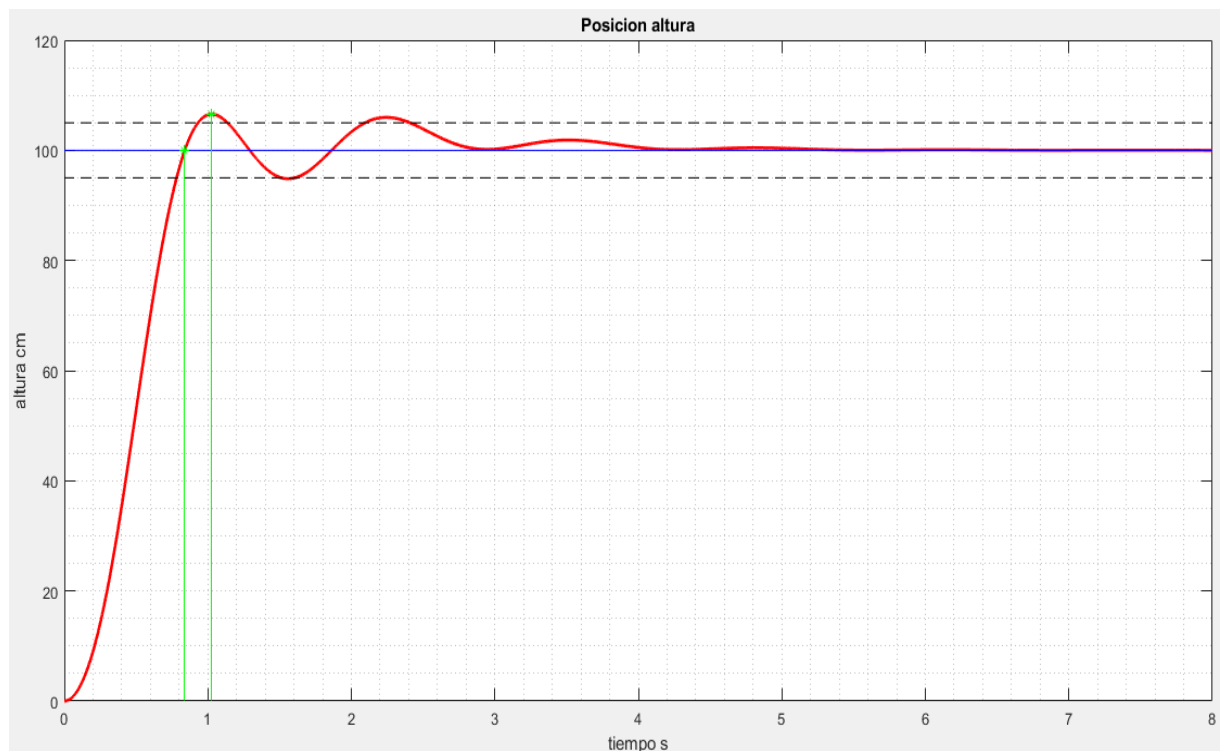


Figura 21. Altura del cuadricóptero con controlador 2

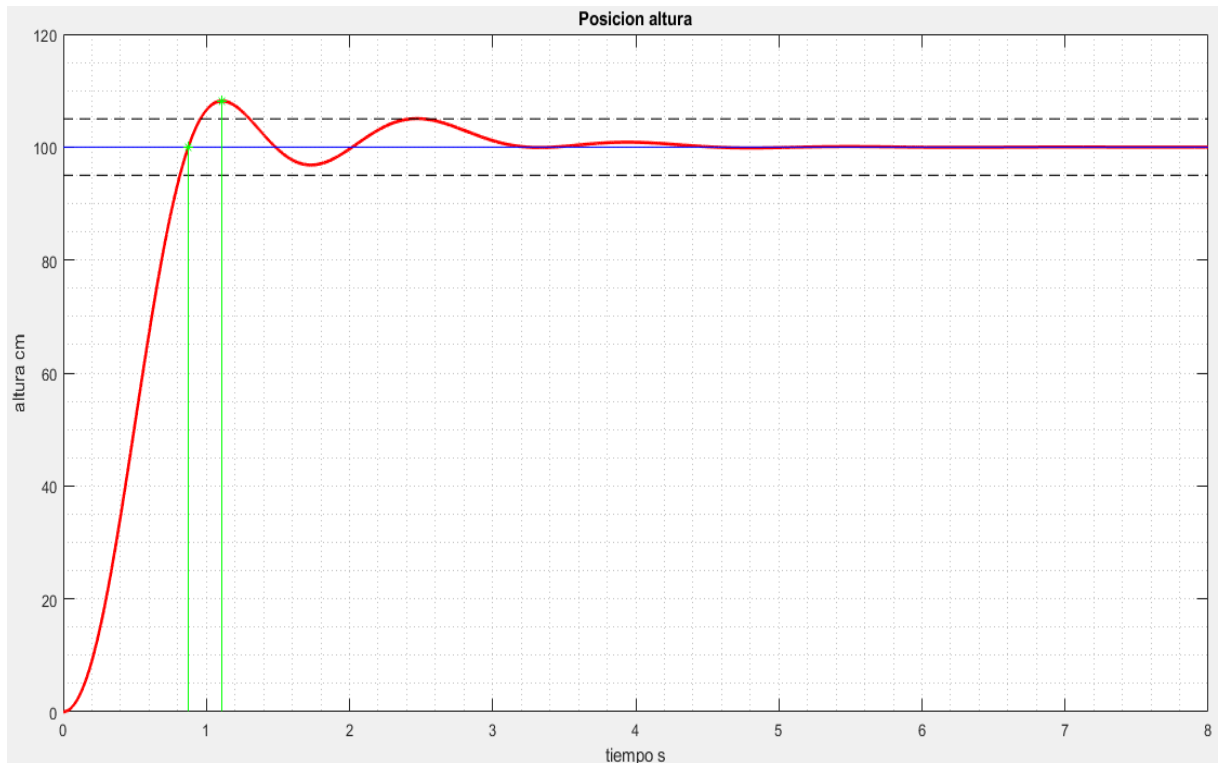


Figura 22 Altura del cuadricóptero con controlador 3-

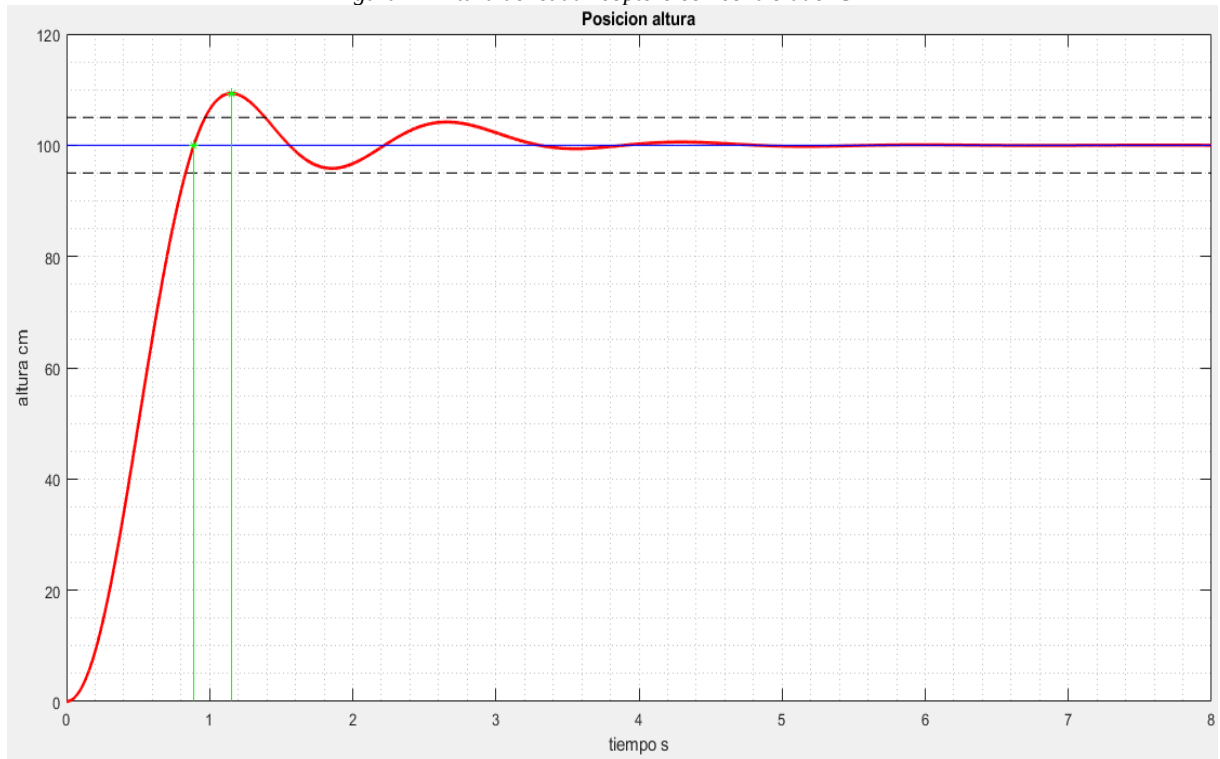


Figura 23. Altura del cuadricóptero con controlador 4

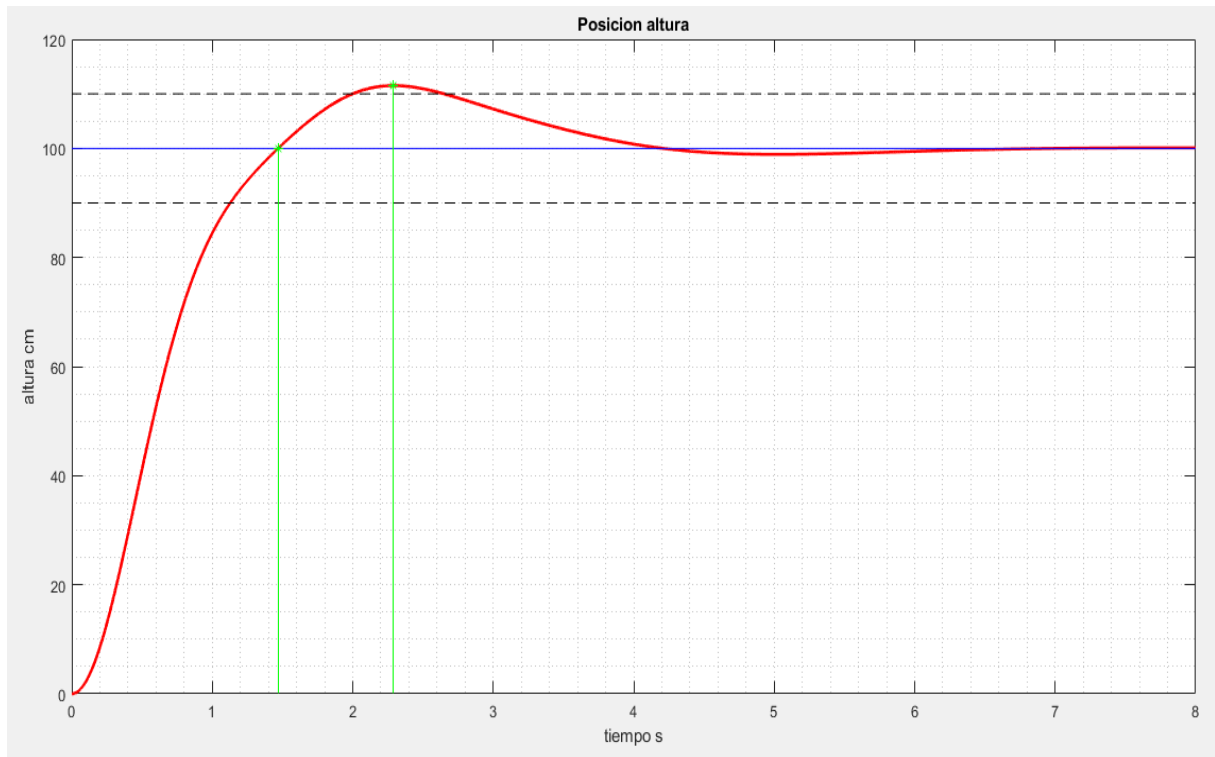


Figura 24. Altura del cuadricóptero con controlador 5

En la tabla 3 se muestran los datos más relevantes de las figuras 20, 21, 22, 23 y 24 se muestran en la tabla 3, N es el número de figura, Tr tiempo de levantamiento del cuadricóptero que es el tiempo en el que tarda el cuadricóptero llegar a la referencia, Mp es el máximo sobre impulso es el valor máximo que alcanza la respuesta del sistema, Tp es el tiempo en que el sistema alcanza el máximo valor, Ts es el tiempo en el que el sistema alcanza la referencia con una tolerancia de 5%.

Tabla 3 Resultados de controladores

N	Tr	Mp	Tp	Ts
1	0.732 s	1.15 m	0.944 s	2 s
2	0.839 s	1.065 m	1.02 s	2.4 s
3	0.873 s	1.081 m	1.11 s	1.25
4	0.89 s	1.093 m	1.15 s	1.4 s
5	1.468 s	1.115 m	2.29 s	2.8 s

Los controladores N2 y N3 tienen los rendimientos más deseados, el controlador N2 tiene un sobre impulso de 6.5% y alcanza la referencia en 0.839 s pero tarda 2.4 segundos para que su valor llegue a la referencia con un 5% de tolerancia, mientras que el controlador N3 tiene

un sobre impulso del 8.1%, alcanza la referencia en 0.873 s y alcanza el valor de la referencia con un 5% de tolerancia en 1.25s, por lo que se elige el controlador N3 para continuar con el desarrollo del controlador del cuadricóptero.

5.2 Resultados control orientación Roll ϕ y Pitch θ

Para el controlador de Roll y Pitch debido a que los parámetros son iguales las ganancias serán iguales, y como el sistema es multivariable; una entrada afecta a más de una variable, por lo que se propone una modificación en el índice de rendimiento J_m (ecuación 4.6) en la cual tenga un valor “ a_1 ” que multiplica el índice de error ángulo Roll o Pitch y un valor “ a_2 ” que multiplica el índice de desempeño de altura. Ambos índices suman 1 y con esto se tiene un índice modificado que evalúa el ángulo Roll o Pitch y la altura simultáneamente, para que la aplicación de MatLab sintonice las ganancias sin perder estabilidad en la altura.

$$J_m = a_1 J_h + a_2 J_r \quad (5.1)$$

Donde:

- J_m índice de eficiencia modificado
- J_h índice de eficiencia altura
- J_r índice de eficiencia Roll/Pitch
- a_1 y a_2 son los coeficientes del porcentaje de cada índice de desempeño.

Para probar la metodología propuesta de un índice de desempeño combinado el valor correspondiente al índice de Roll, se varió desde 1 hasta 0.5 con intervalos de 0.1, en la tabla 4 se muestran los resultados obtenidos en la aplicación de las ganancias del controlador PID K_p , K_i y K_d respectivamente, los valores de a_1 , a_2 y los índices de desempeño del ángulo (J_r), altura (J_h) y el índice modificado J_m .

La fórmula 5.1 fue añadida al código donde se parametrizo el ángulo para que ambos tuvieran valor similar, para el uso de la app en MatLab se añadió una entrada de 1m en el tiempo de 0 segundos, y a los cuatro segundos se añade una entrada de 10° al cuadricóptero.

Tabla 4. Ganancias controlador Roll/Pitch

<i>N</i>	<i>a</i>₁	<i>a</i>₂	<i>K</i>_p	<i>K</i>_i	<i>K</i>_d	<i>J</i>_h	<i>J</i>_r	<i>J</i>_m
1	0	1	17.774	24.549	6.571	0.0157	0.0299	0.0299
2	0.1	0.9	7.538	0.613	6.073	0.0067	0.0217	0.0202
3	0.2	0.8	1.229	1.88	2.955	0.0003	0.0233	0.017
4	0.3	0.7	0.1	0.104	1.387	0.0002	0.021	0.014
5	0.4	0.6	0.1	0.1	1.264	0.0002	0.021	0.012
6	0.5	0.5	0.323	2.685	2.522	0.0002	0.021	0.01

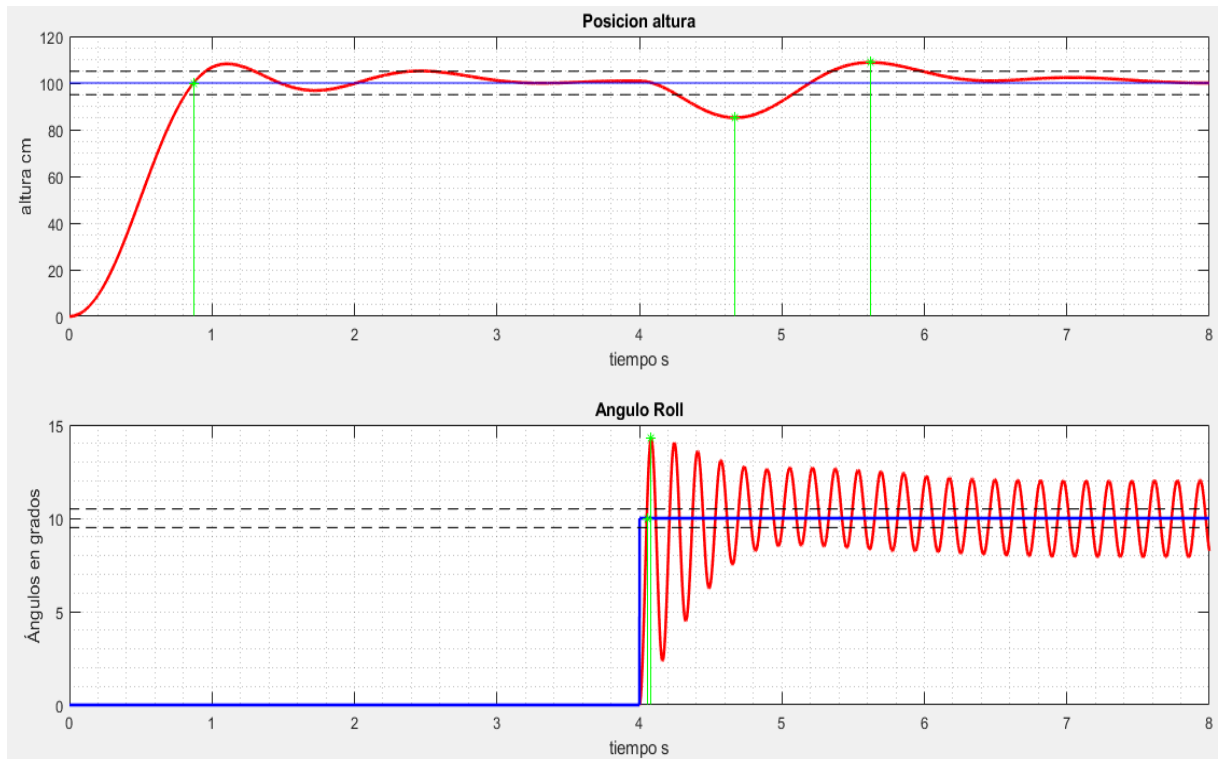


Figura 25. Controlador N1 de Roll/Pitch

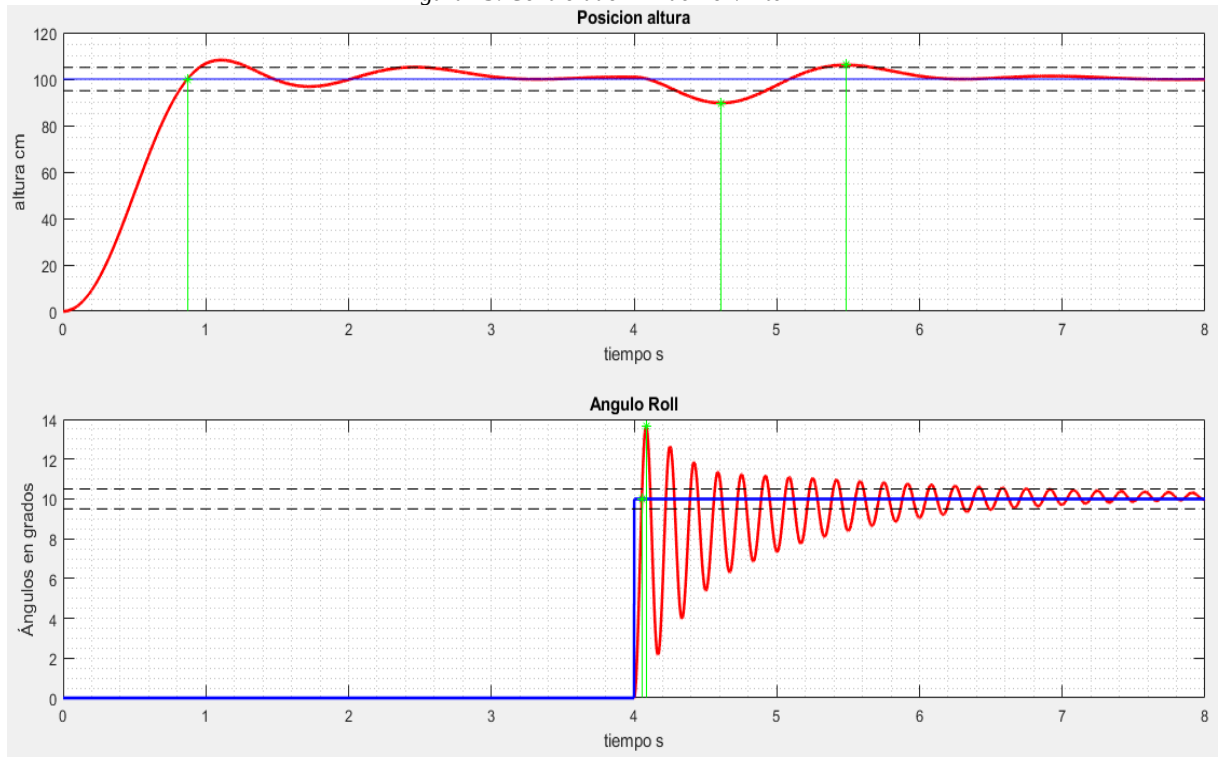


Figura 26. Controlador N2 de Roll/Pitch

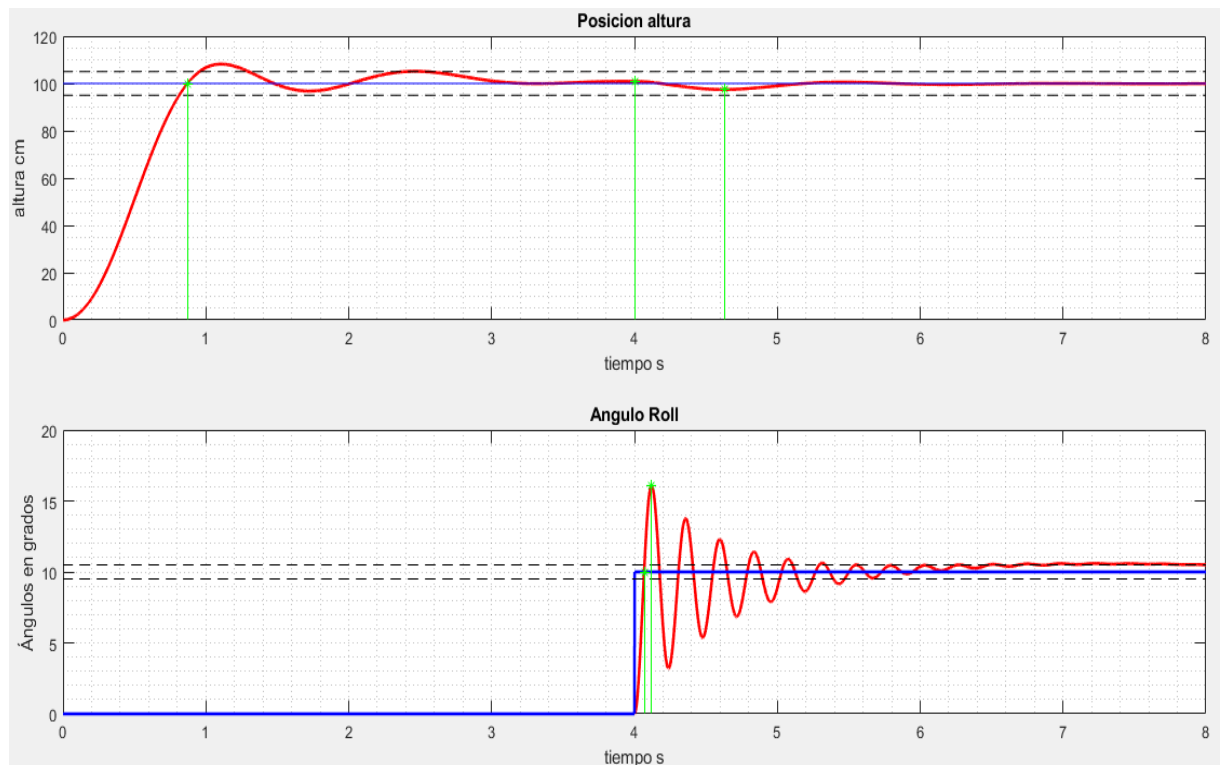


Figura 27. Controlador N3 de Roll/Pitch

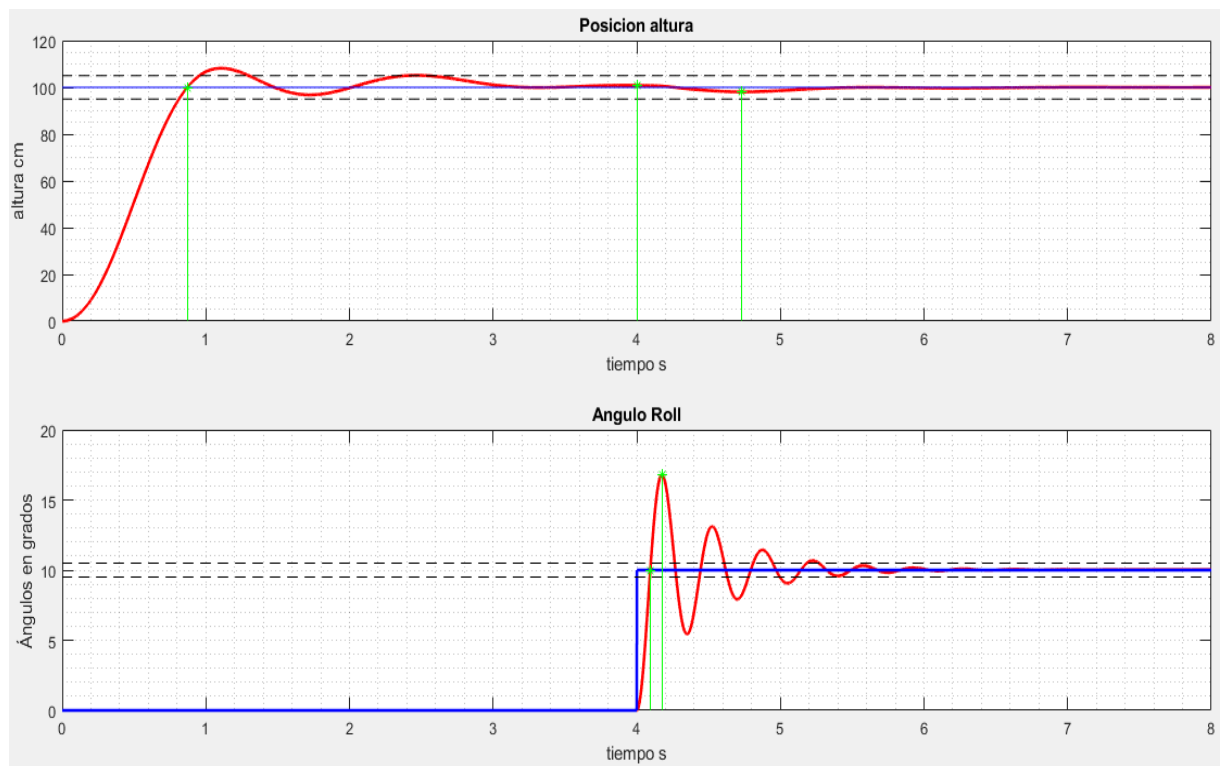


Figura 28. Controlador N4 de Roll/Pitch

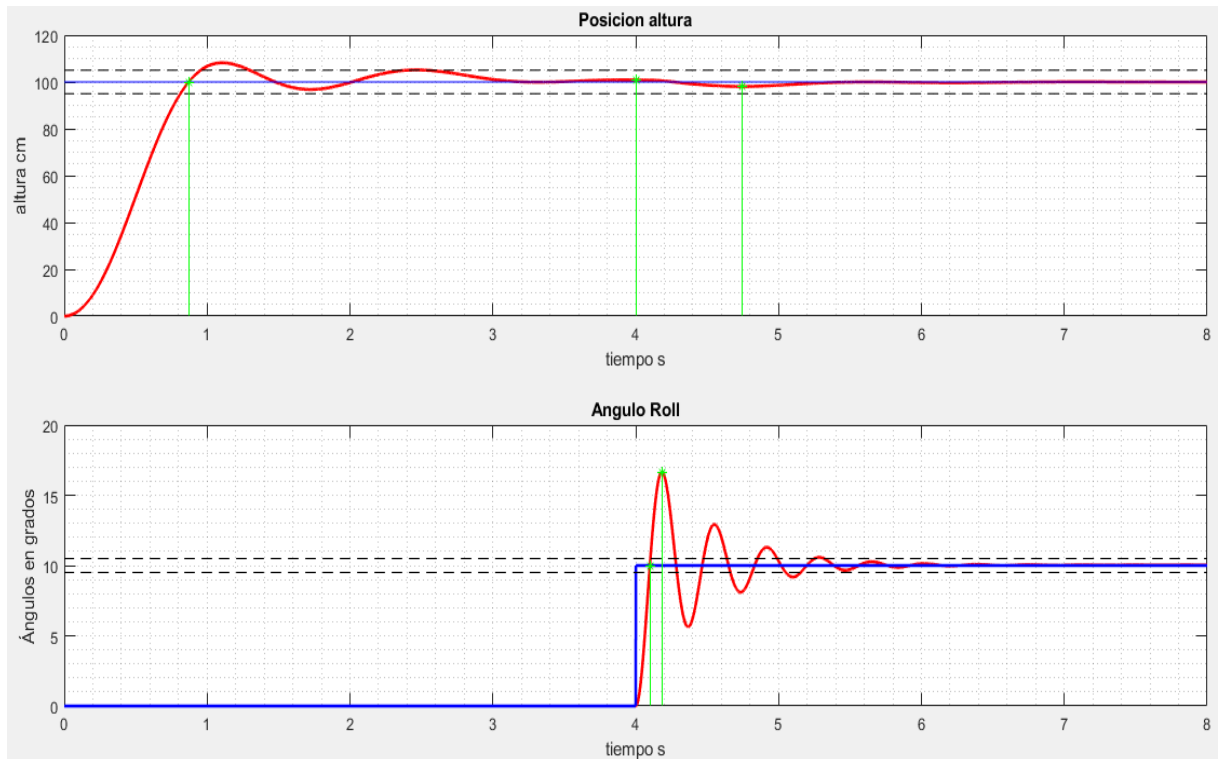


Figura 29. Controlador N5 de Roll/Pitch

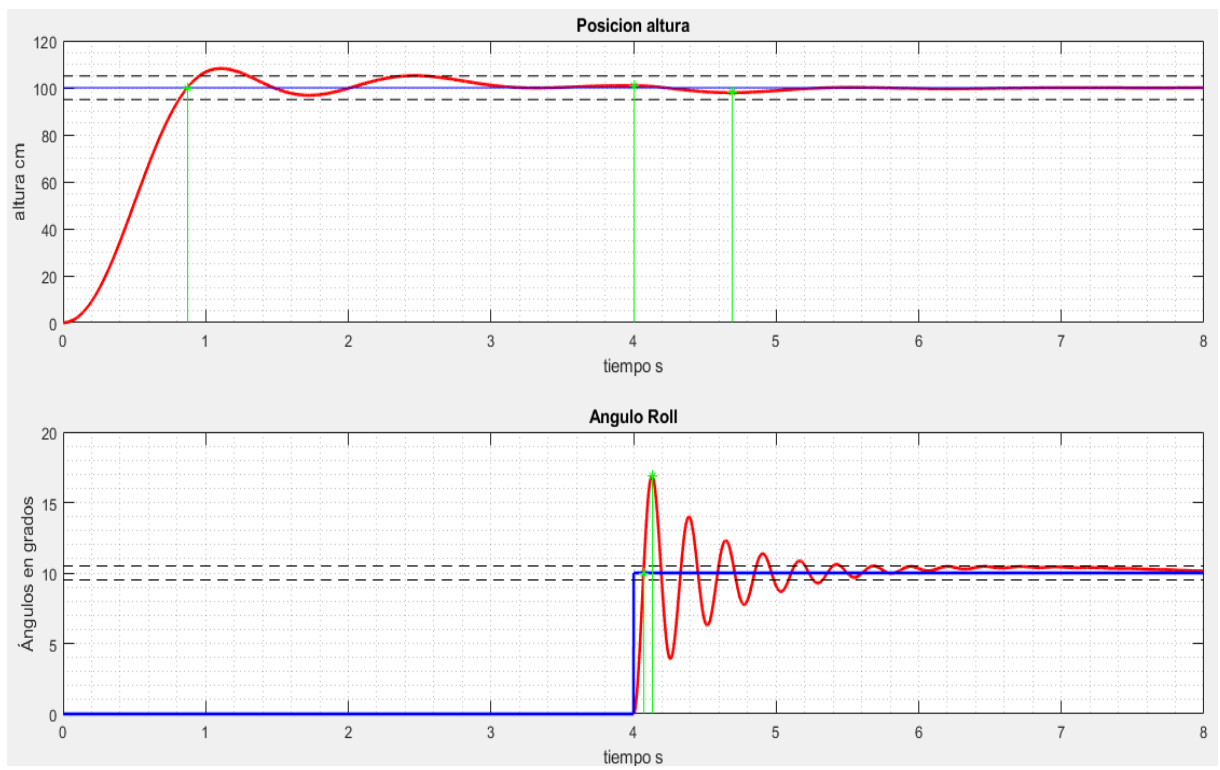


Figura 30. Controlador N6 de Roll/Pitch

La tabla 5 muestra los principales resultados de los controladores obtenidos por la aplicación de MatLab, T_r es el tiempo en que el cuadricóptero alcanza la referencia después de aplicar la entrada de referencia que para este trabajo fue a los 4 segundos, M_p es el máximo sobre impulso, T_p es el tiempo en que el M_p sucede después de aplicar la entrada de referencia que para este trabajo fue a los 4 segundos, T_s es el tiempo en que el sistema tiene un error igual o menor al 5% de la referencia después de aplicar la entrada de referencia que para este trabajo fue a los 4 segundos, ND es valor no disponible (a que en el tiempo de simulación no se encontraba el valor correspondiente), K_+ es la altura máxima que el sistema tiene después de aplicar la entrada de 10° al cuadricóptero, K_- es la altura mínima que el sistema tiene después de aplicar la entrada de 10° al cuadricóptero, los guiones (---) representan que el cambio es despreciable, en la última fila se expresa si el cuadricóptero es estable con el controlador probado.

El controlador N5 tiene el resultado mejor a lo esperado.

Tabla 5 Resultados controladores Roll/Pitch

	Roll/Pitch				Altura		Estable
N	T_r	M_p	T_p	T_s	K_+	K_-	E
1	0.053	14.26	0.08	ND	108.79	85.11	SI
2	0.056	13.64	0.08	2.6	106.05	89.70	SI
3	0.069	16.09	0.12	1.6	---	97.45	SI
4	0.096	16.79	0.17	1.4	---	98.05	SI
5	0.1	16.64	0.18	1.4	---	98.04	SI
6	0.07	16.89	0.13	1.6	---	97.91	SI

5.3 Resultados de control de orientación de Yaw ψ

Debido a que el sistema es multivariable, una entrada afecta a más de una variable por lo que al igual que en el diseño de controlador de Roll y Pitch se utilizará el índice de rendimiento modificado J_m (ecuación 4.6) en la cual tenga un valor " a_1 " que multiplica el índice de error ángulo Yaw y un valor " a_2 " que multiplica el índice de desempeño de altura. Ambos índices suman 1 y con esto se tiene un índice modificado que evalúa el ángulo Yaw y la altura simultáneamente, para que la aplicación de MatLab sintonice las ganancias sin perder estabilidad en la altura.

$$J_m = a_1 J_h + a_2 J_y \quad (5.2)$$

Donde:

- J_m índice de eficiencia modificado
- J_h índice de eficiencia altura
- J_y índice de eficiencia Yaw
- a_1 y a_2 son los coeficientes del porcentaje de cada índice de desempeño.

Para probar la metodología propuesta de un índice de desempeño combinado el valor correspondiente al índice de Yaw, se varió desde 1 hasta 0.5 con intervalos de 0.1, en la tabla 6 se muestran los resultados obtenidos en la aplicación de las ganancias del controlador PID K_p , K_i y K_d respectivamente, los valores de a_1 , a_2 y los índices de desempeño del ángulo (J_y), altura (J_h) y el índice modificado J_m .

La ecuación 5.2 fue añadida al código donde se parametrizo el ángulo para que ambos tuvieran valor similar, para el uso de la app en MatLab se añadió una entrada de 1m en el tiempo de 0 segundo, y a los cuatro segundos se añade una entrada de 10° de Yaw al cuadricóptero.

Tabla 6 Ganancias controladores yaw

N	A1	a2	Kp	Ki	Kd	Jm	Jy	Jh
1	0	1	20.	0.639	15.247	0.759	0.071	0.071
2	0.1	0.9	7.917	5.526	7.172	0.031	0.021	0.022
3	0.2	0.8	7.56	1.307	7.075	0.029	0.0212	0.022
4	0.3	0.7	6.037	0.821	6.251	0.017	0.022	0.020
5	0.4	0.6	4.058	0.411	4.961	0.005	0.025	0.017
6	0.5	0.5	4.336	0.649	5.3	0.008	0.024	0.016

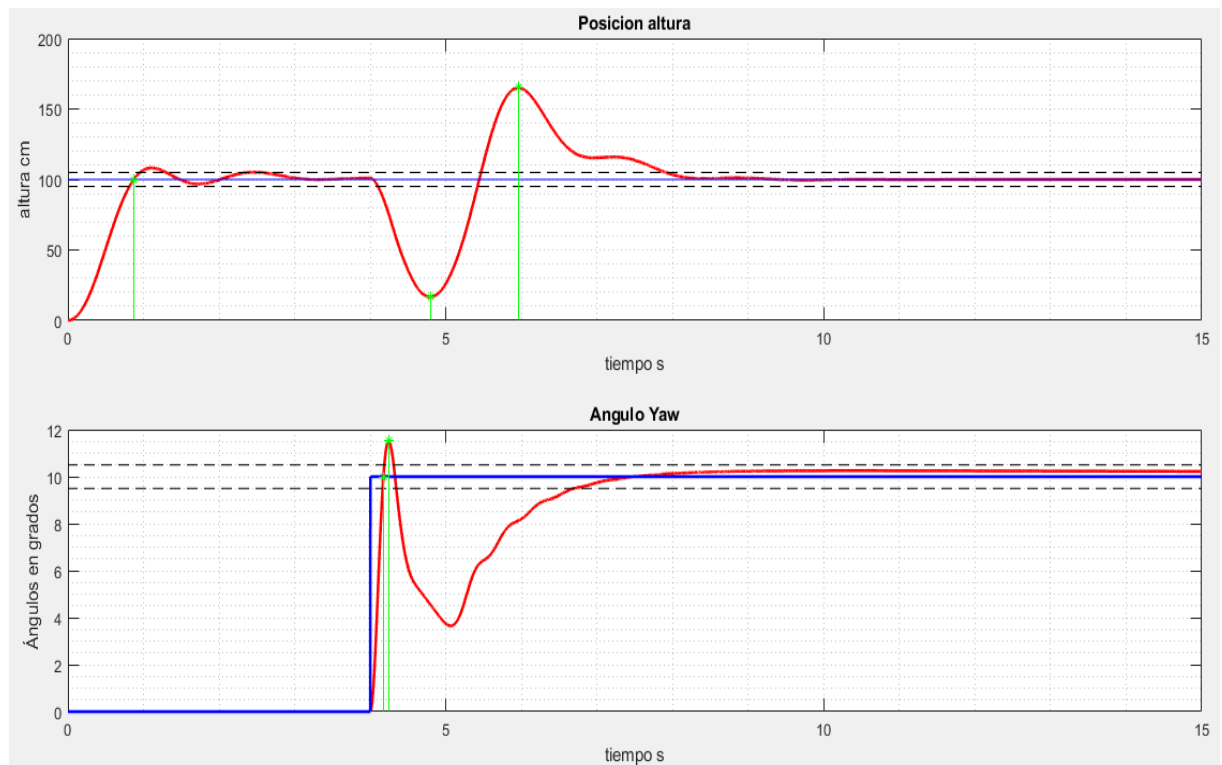


Figura 31. Controlador N1 Yaw

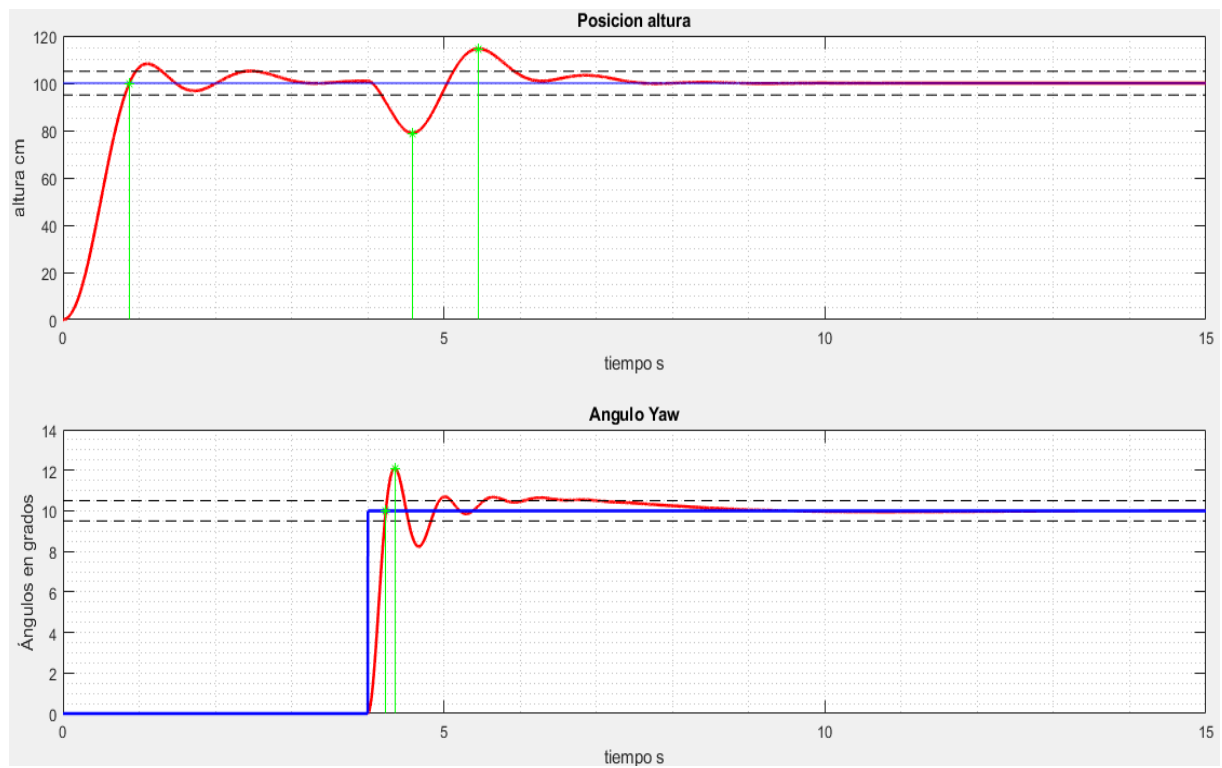


Figura 32. Controlador N2 Yaw.

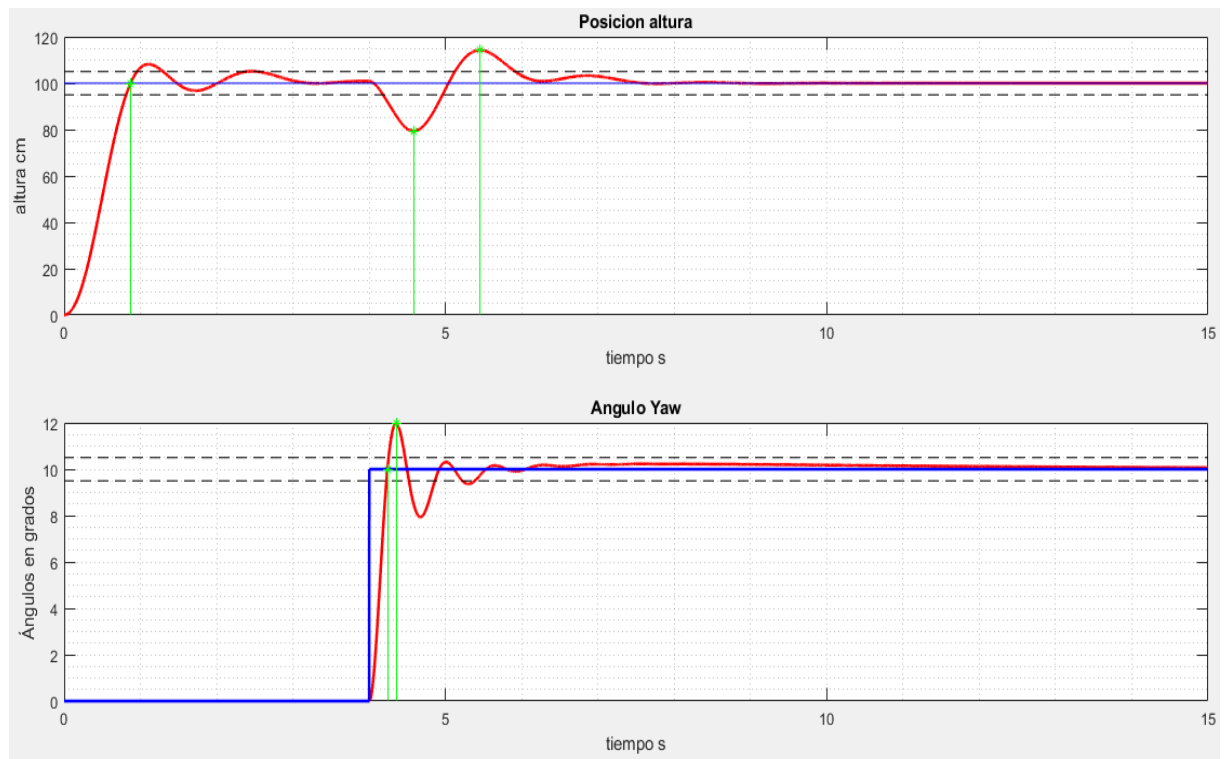


Figura 33. Controlador N3 Yaw

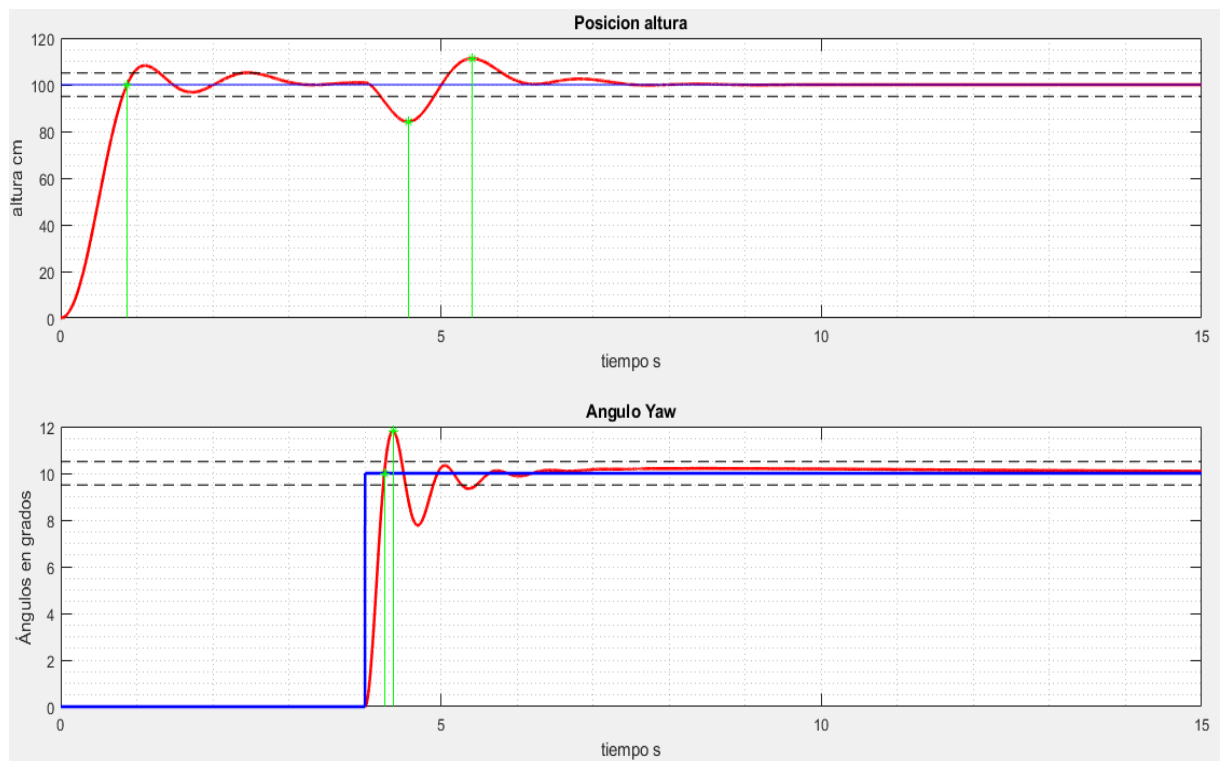


Figura 34. Controlador N3 Yaw

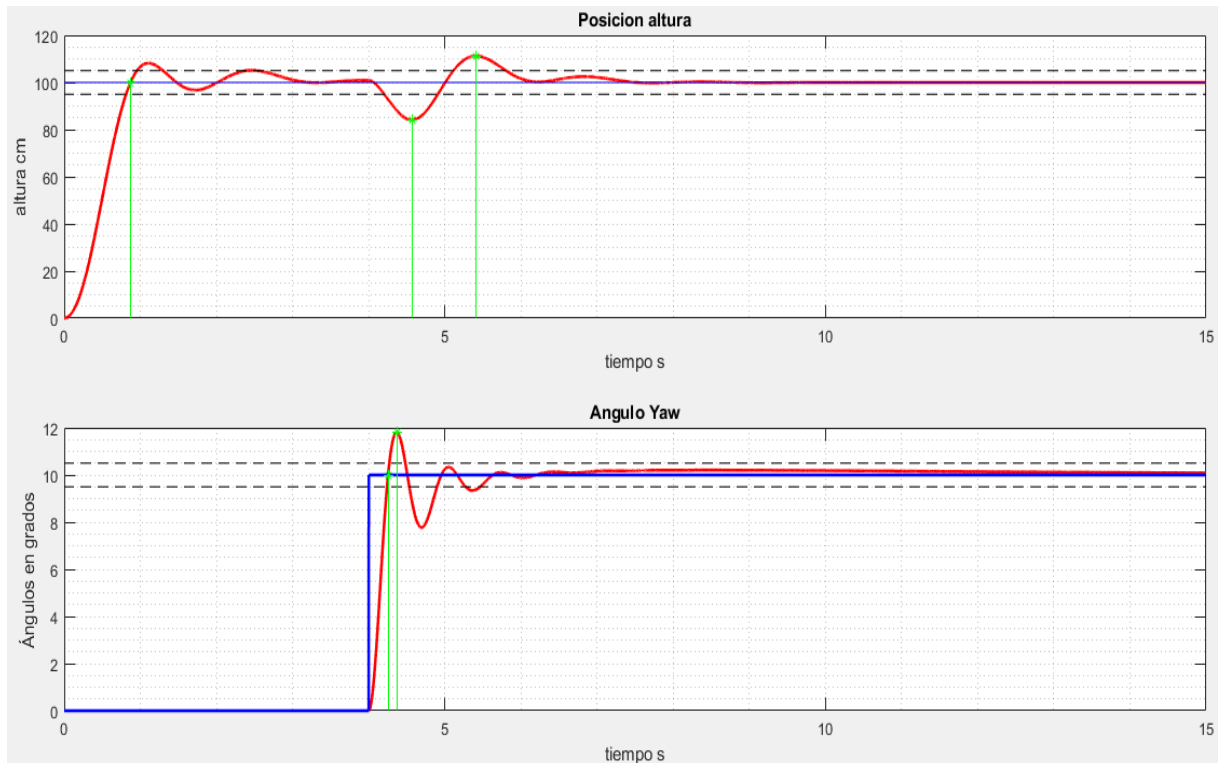


Figura 35. Controlador N4 Yaw

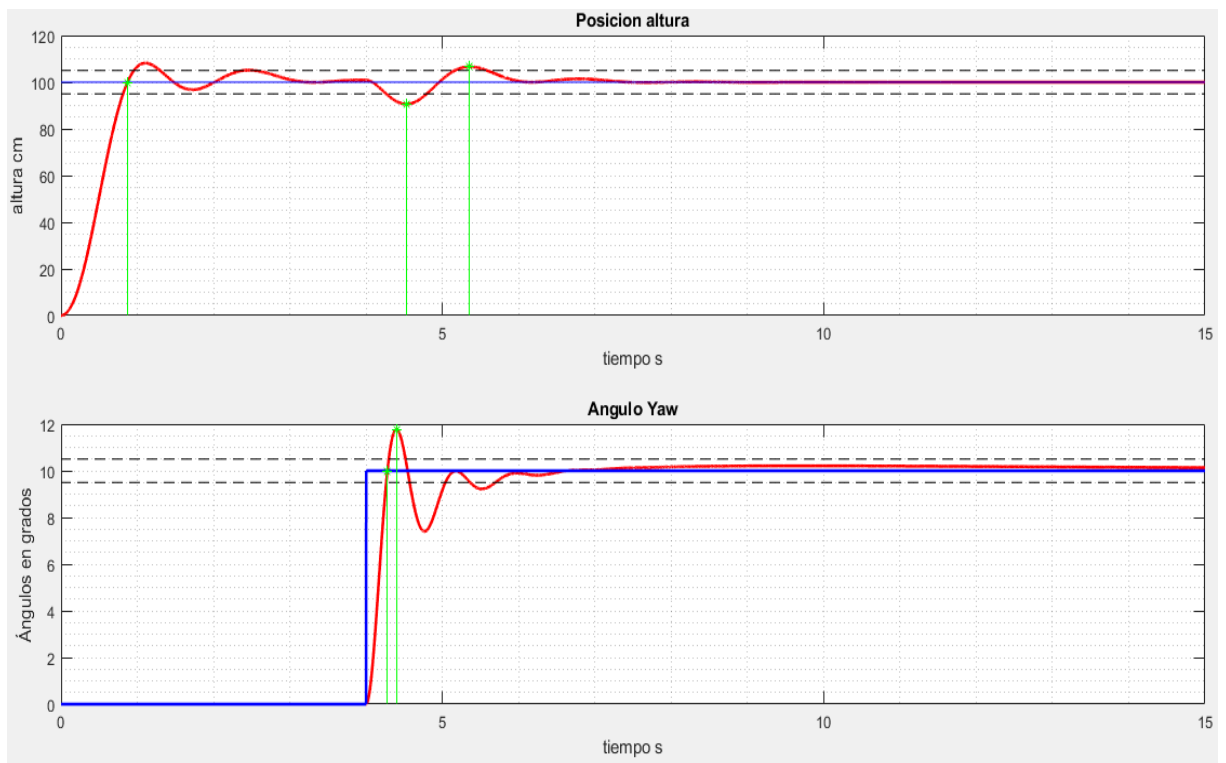


Figura 36. Controlador N5 Yaw

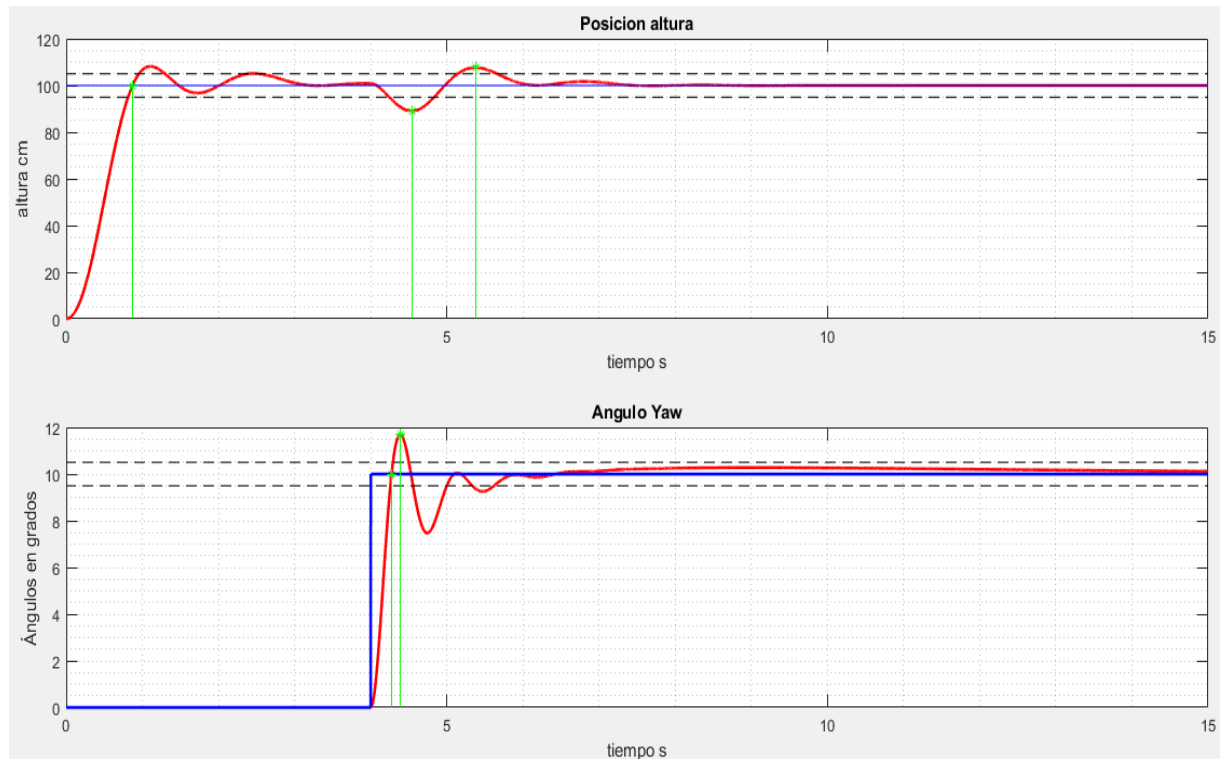


Figura 37. Controlador N6 Yaw

Tabla 7. Resultados de controladores Yaw

	yaw				Altura		Estable
N	Tr	Mp	Tp	Ts	K+	K-	E
1	0.17	11.52	0.24	3	165.04	16.59	SI
2	0.23	12.10	0.35	4	114.54	78.88	SI
3	0.24	11.96	0.35	1.5	114.24	79.34	SI
4	0.25	11.81	0.36	1.5	111.24	84.06	SI
5	0.28	11.17	0.39	1.5	106.61	90.58	SI
6	0.27	11.69	0.38	1.5	107.69	89.04	SI

La tabla 7 muestra los principales resultados de los controladores obtenidos por la aplicación de MatLab, Tr es el tiempo en que el cuadricóptero alcanza la referencia, después de aplicar la entrada de referencia que para este trabajo fue a los 4 segundos, Mp es el máximo sobre impulso que el cuadricóptero tiene, Tp es el tiempo en que el Mp sucede después de aplicar la entrada de referencia que para este trabajo fue a los 4 segundos, Ts es el tiempo aproximado en que el sistema tiene un error igual o menor al 5 % de la referencia después de aplicar la entrada de referencia que para este trabajo fue a los 4 segundos, K+ es la altura máxima que

el sistema tiene después de aplicar la entrada de 10° al cuadricóptero, K- es la altura mínima que el sistema tiene después de aplicar la entrada de 10° al cuadricóptero, en la última fila se expresa si el cuadricóptero es estable con el controlador probado.

El controlador N5 tiene el resultado mejor a lo esperado.

5.4 Pruebas del cuadricóptero.

Para probar el resultado del cuadricóptero se probó el funcionamiento con entradas de distinto valor con los valores elegidos para los controladores Roll, Pitch, Yaw y altura, todo esto con las ganancias seleccionadas con mejor respuesta en el diseño por el método de algoritmos genéticos.

Para todos los controladores se probó el funcionamiento a diferentes entradas, en el caso de control de altura se aplica una entrada de -1 m, 0.5 m, 1 m, 1.5 m, 2 m y 2.5 m, para el caso de la entrada negativa esta solo es un punto de referencia por lo que no representa un impedimento para su traslado, en la figura 38 y 39 se muestra el comportamiento del cuadricóptero y en la tabla 8 se mencionan los resultados.

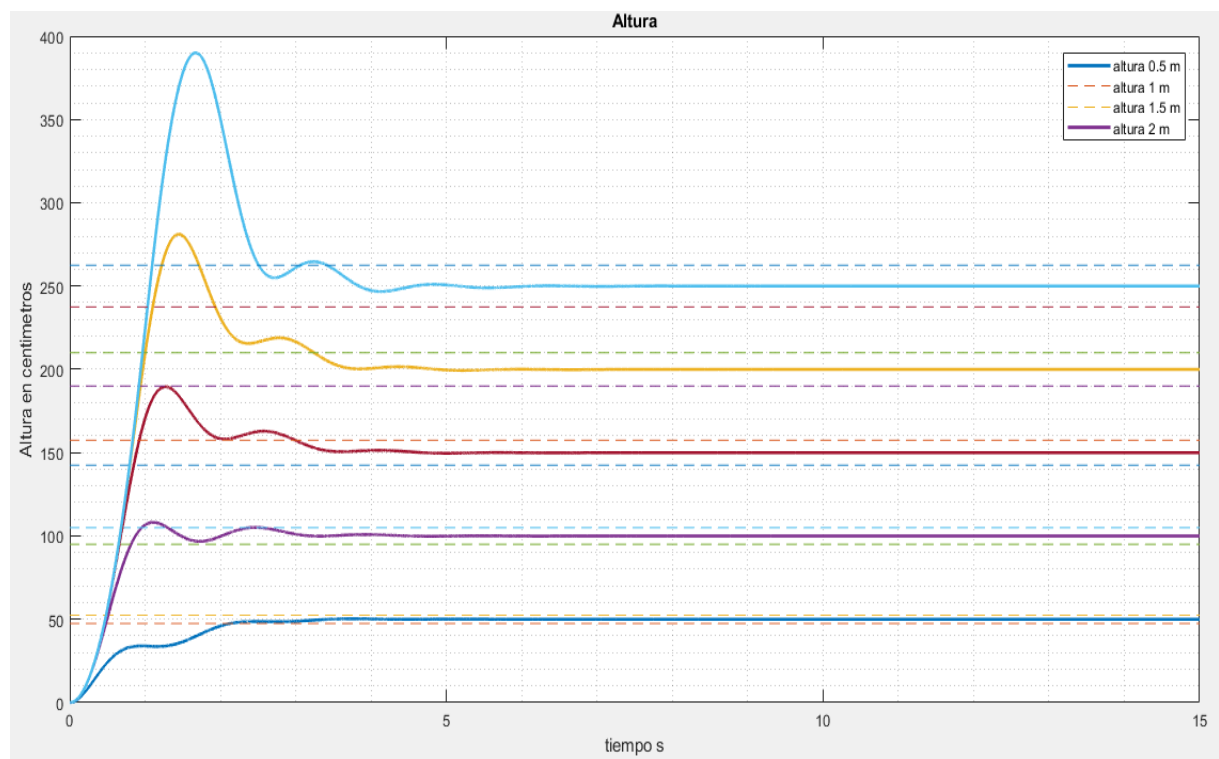


Figura 38. Comportamiento de altura del cuadricóptero a distintas entradas

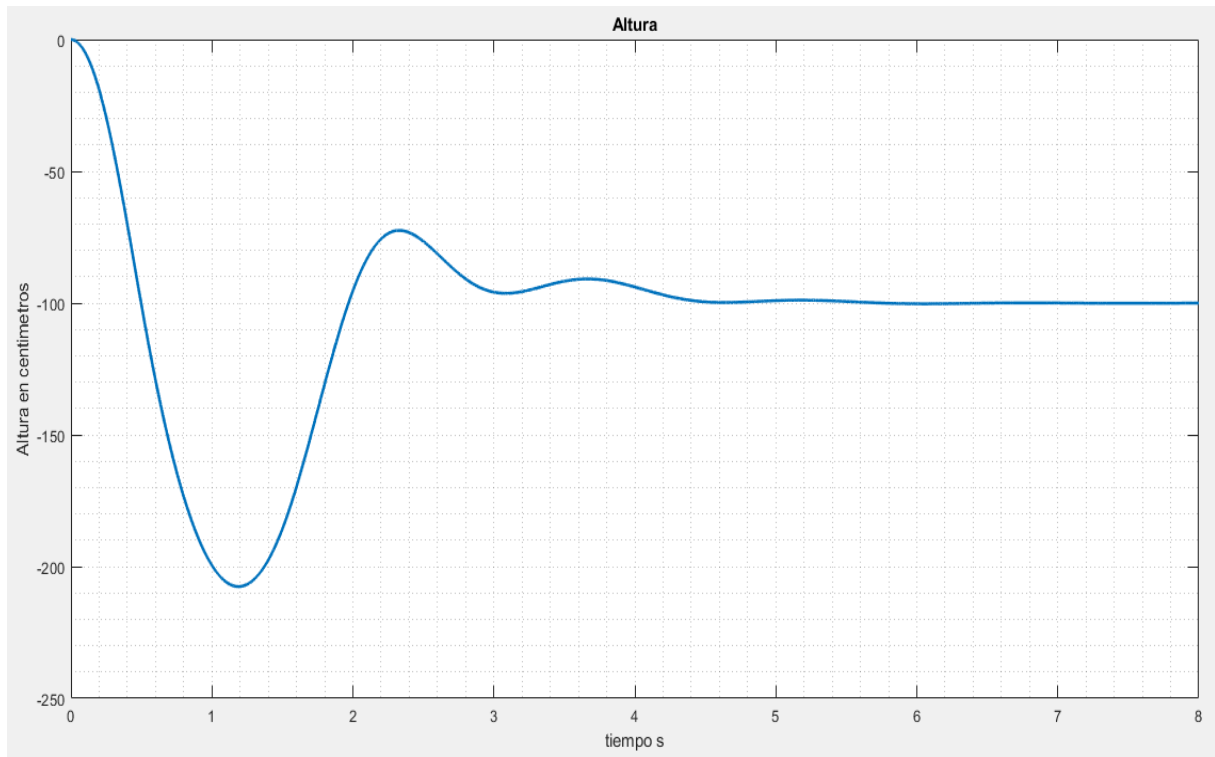


Figura 39. Comportamiento de altura del cuadricóptero a una entrada negativa

Tabla 8. Resultados altura a diferentes entradas

N	Entrada (cm)	Ts(s)	Tr (s)	Mp (cm)	Tp (s)
1	-100	4.0-	0.49	-207.57	1.19
2	50	2.0	3.44	50.3	3.75
3	100	1.25	0.87	108.1	1.10
4	150	3.0	0.88	189.5	1.27
5	200	3.25	0.97	281.0	1.44
6	250	3.5	1.06	389.9	1.67

Donde Tr es el tiempo en que el cuadricóptero llega a la referencia desde el momento de la entrada, Mp es el máximo sobre impulso, donde en la primera columna es el menor valor obtenido debido a que en este caso es negativa la entrada, en el resto son el mayor valor obtenido, Tp es el tiempo en que ocurrió el Mp desde el momento de la entrada.

En la figura 40 se muestra que las variables Roll, Pitch y Yaw no tienen variación alguna con las entradas en altura a cualquier valor.

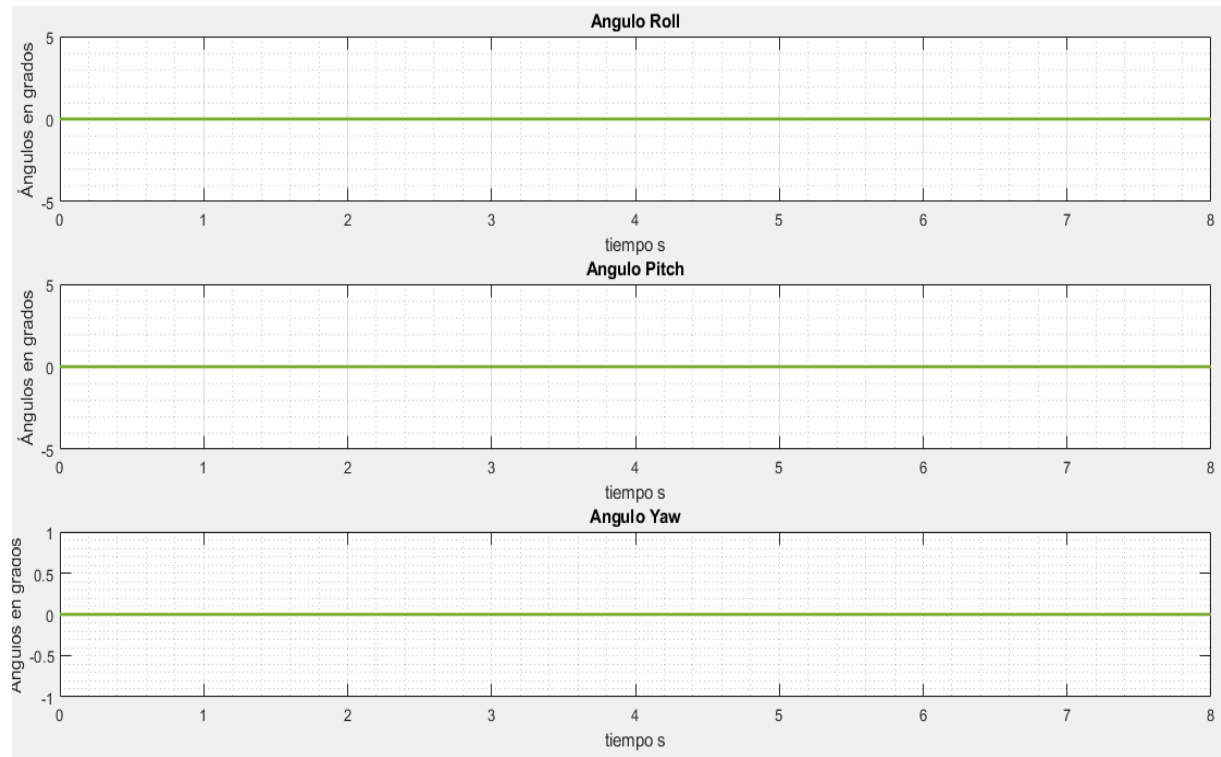


Figura 40. Comportamiento de Roll, Pitch y Yaw a entradas diferentes de altura

Para el caso de las pruebas de los ángulos se tiene que los ángulos negativos y positivos no influyen en el comportamiento solo la magnitud; las entradas se acotaron a valores positivos. Además, los cuadricópteros están diseñados a trabajar en ángulos de -25° a 25° , por lo que se acotaron a valores menores a 25° . Para controladores de Roll, Pitch y Yaw se comprobó el funcionamiento a 5° , 10° , 15° , 20° y 25° , todas estas entradas se probaron con el cuadricóptero a 1 metro de altura.

En la figura 41, 42 se muestra el comportamiento de la altura y el ángulo Roll y en la figura 43 se muestra el comportamiento de los ángulos Pitch y Yaw con las diferentes entradas de ángulo Roll.

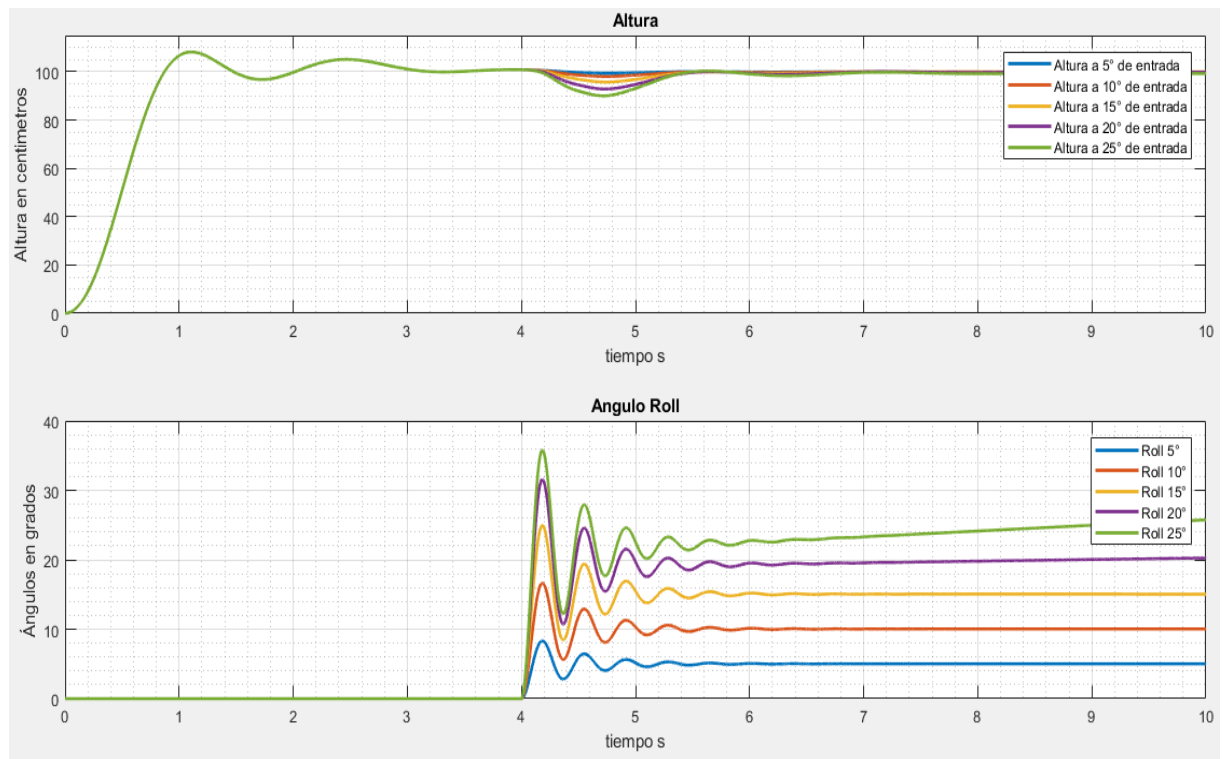


Figura 41 Comportamiento de altura y ángulo Roll con distintas entradas de Roll

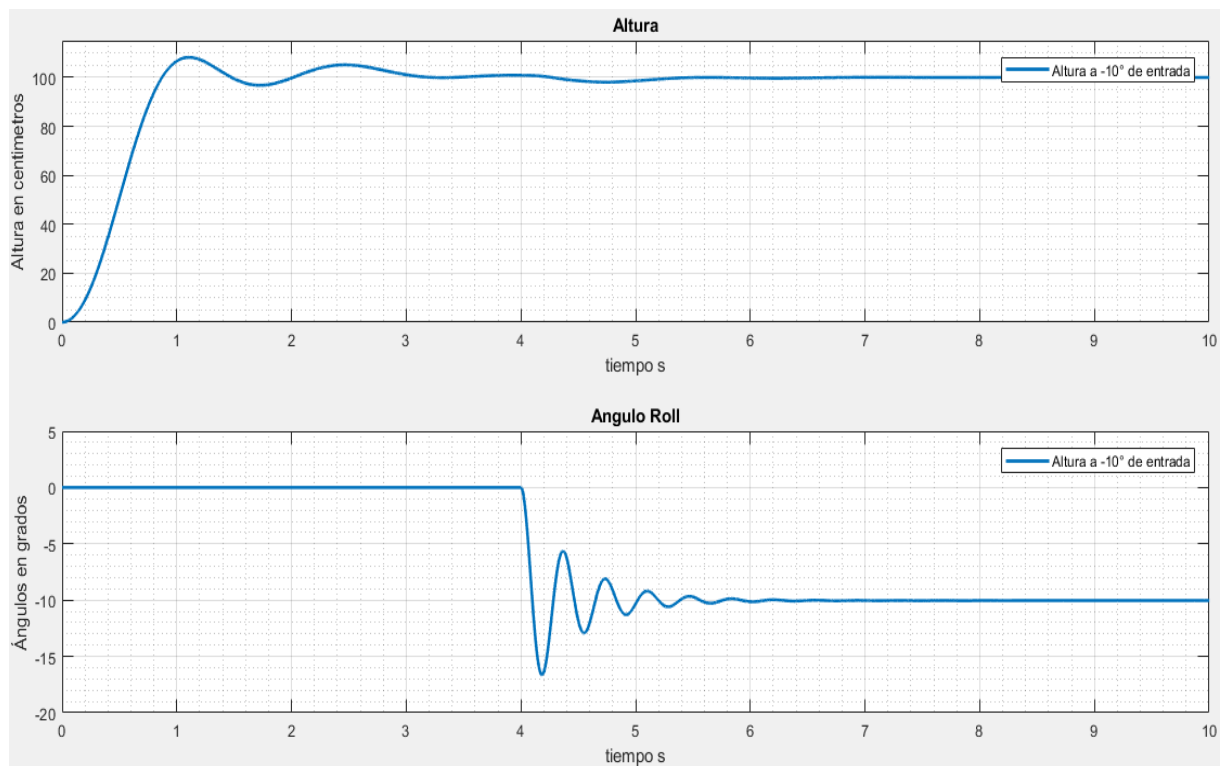


Figura 42. Comportamiento de ángulo Roll del cuadricóptero a una entrada negativa

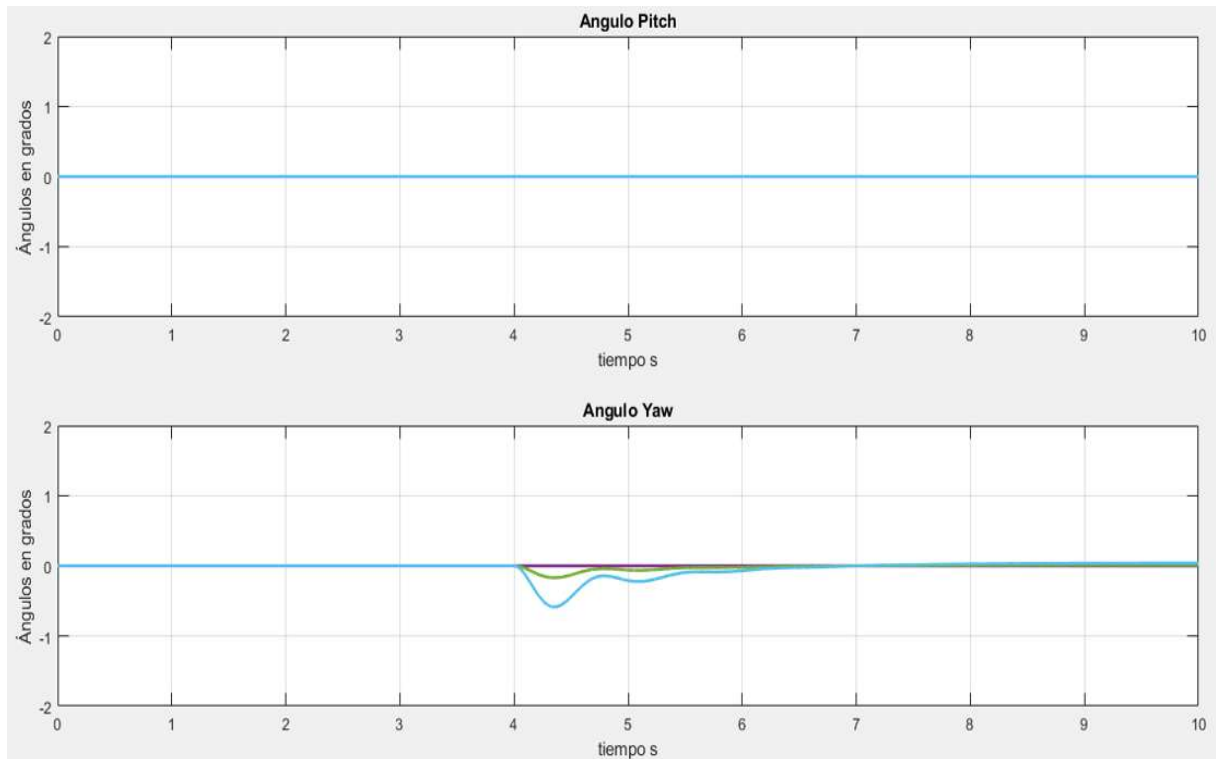


Figura 43. Comportamiento de ángulo Pitch y Yaw con diferentes entradas de Roll

En la figura 43 se muestra que el único de los ángulos que se modifica con el cambio de Roll es Yaw, pero este tiene un cambio muy pequeño en el caso una entrada de Roll de 25° el ángulo Yaw cambia a -0.58°, mientras que una entrada de ángulo Roll provoca un cambio de -0.16° para el resto de entradas no presenta ningún cambio.

Tabla 9. Resultados de Roll y Altura a diferentes entradas de Roll

N	Entrada (°)	Ts(s)	Tr (s)	Mp (°)	Tp (s)	K+	k-
ROLL						ALTURA	
1	-10	1.25	0.10	-16.64	0.185	100.75	98.04
2	5	1.5	0.10	8.32	0.185	100.76	99.40
3	10	1.25	0.10	16.64	0.185	100.75	98.04
4	15	1.25	0.10	24.97	0.185	100.74	95.73
5	20	1.20	0.10	31.60	0.185	100.68	92.83
6	25	ND	0.11	35.80	0.185	100.55	90.02

Donde Tr es el tiempo en que el cuadricóptero llega a la referencia desde el momento de la entrada, Mp es el máximo sobre impulso, Tp es el tiempo en que ocurrió el Mp desde el momento de la entrada, K+ es la posición altura máxima que alcanza el cuadricóptero después de la entrada de Roll y K- es la posición mínima que alcanza el cuadricóptero después de la entrada. Se puede observar que el primer valor del Mp, es el menor valor obtenido debido a

que en este caso es negativa la entrada, en el resto son mayores al valor de referencia, ND significa no disponible debido a que durante el tiempo de simulación este valor no se obtuvo. Para el caso cunado se modifica la referencia del ángulo de Yaw en los mismos valores que las de Roll; las respuestas del comportamiento de la altura y el ángulo Yaw se muestran en la figura 44 y 45; y en la figura 46 se muestra el comportamiento de los ángulos Roll y Pitch.

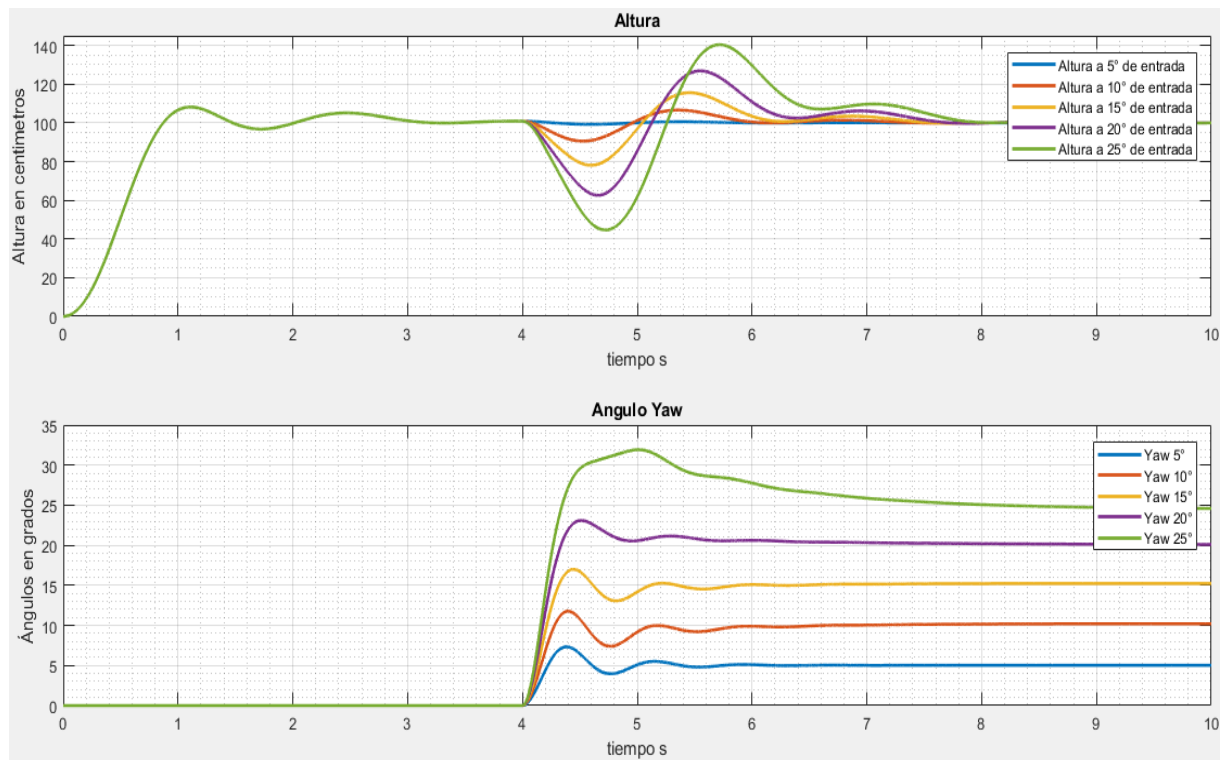


Figura 44. Comportamiento de altura y ángulo Yaw con distintas entradas de Yaw

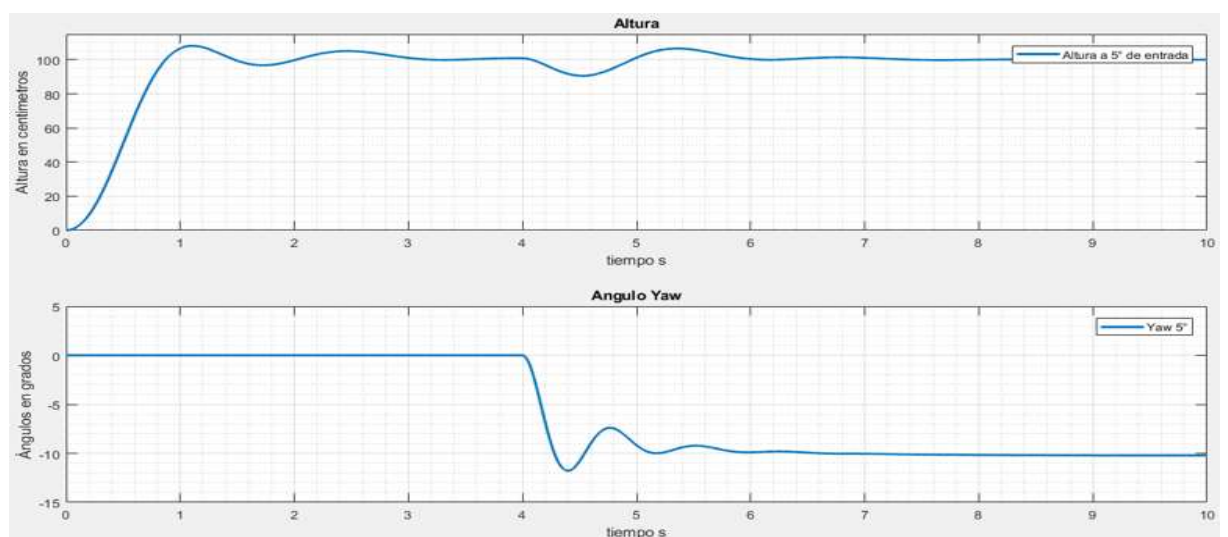


Figura 45. Comportamiento de ángulo Yaw del cuadricóptero a una entrada negativa

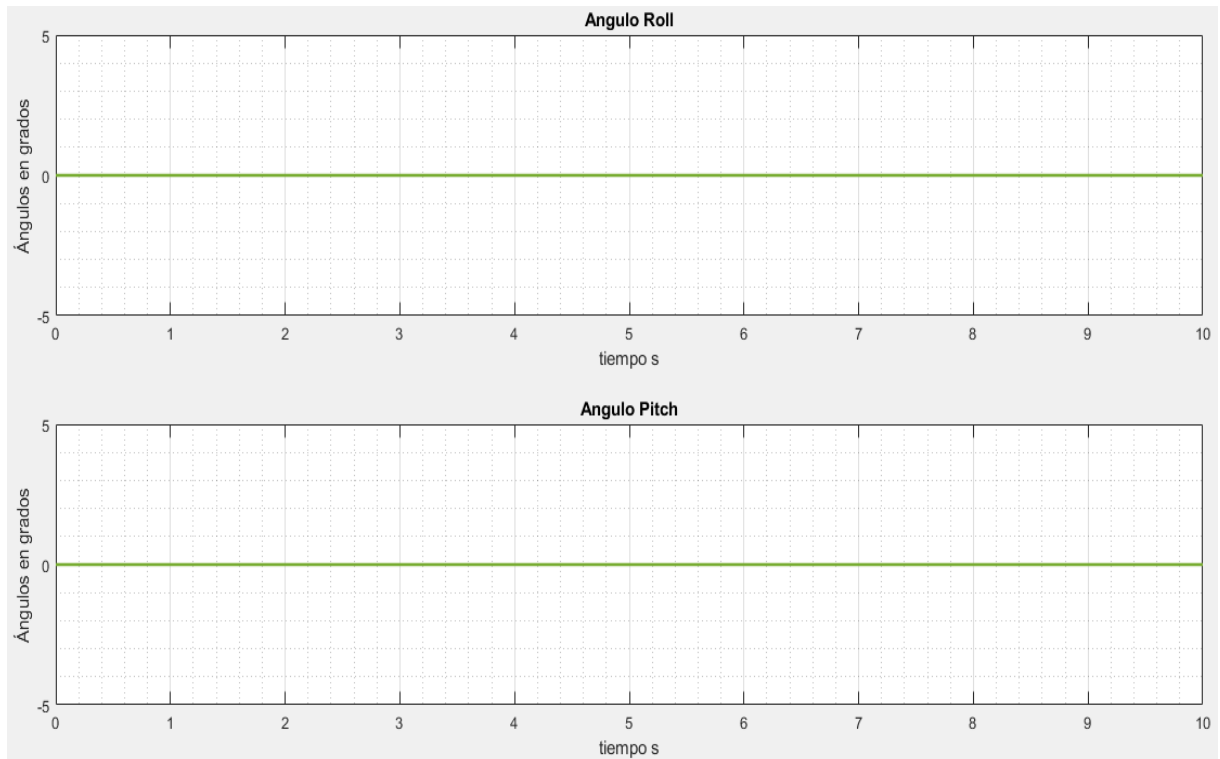


Figura 46. Comportamiento de Angulo Pitch y Roll con diferentes entradas de Yaw

N	Entrada (°)	Ts(s)	Tr (s)	Mp (°)	Tp (s)	K+	k-
	YAW					ALTURA	
1	-10	1.5	0.28	-11.77	0.39	108.18	90.58
2	5	1.2	0.22	7.33	0.38	108.18	99.24
3	10	1.5	0.28	11.77	0.39	108.18	90.58
4	15	1.0	0.31	17.00	0.44	115.58	78.10
5	20	1.5	0.33	23.08	0.51	126.87	62.58
6	25	3.0	0.34	31.92	1.01	140.40	44.65

Donde Tr es el tiempo en que el cuadricóptero llega a la referencia desde el momento de la entrada, Mp es el máximo sobre impulso, Tp es el tiempo en que ocurrió el Mp desde el momento de la entrada, K+ es la posición altura máxima que alcanza el cuadricóptero después de la entrada de Roll y K- es la posición mínima que alcanza el cuadricóptero después de la entrada. Igual que en caso de Roll, se puede observar que el primer valor del Mp, es el menor valor obtenido debido a que en este caso es negativa la entrada, en el resto son mayores al valor de referencia.

5.5 Comparación con trabajos anteriores

Los trabajos realizados por Paul Ortiz 2017 reportan que para el diseño de control de altura es: el tiempo de subida es $T_r = 1.9$ s., el tiempo de establecimiento $T_s = 7.5$ s. y el máximo sobre impulso es $M_p = 25$ % (figura 47).

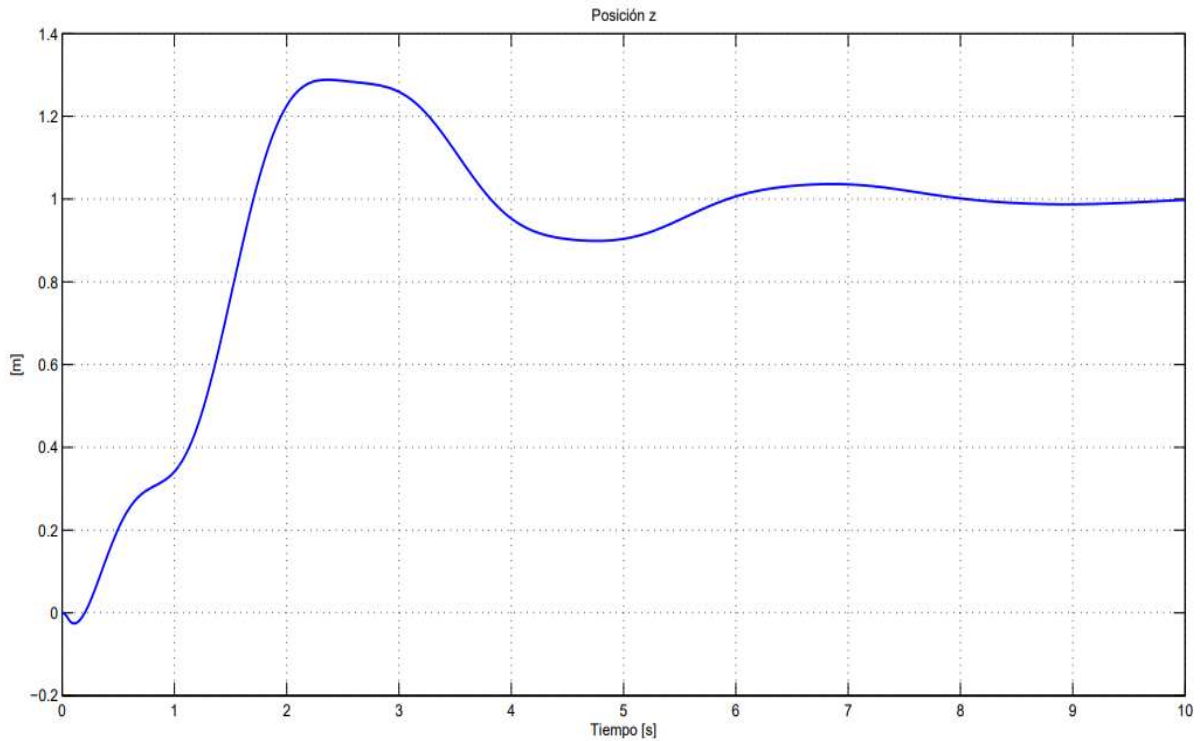


Figura 47. Comportamiento de altura del trabajo de Paul Ortiz 2017

Donde se observa que por medio de algoritmos genéticos se obtiene un máximo sobre impulso de 8.1% en comparación a los 25% de Paul Ortiz, y un tiempo de establecimiento de 7.5 s. en comparación de los 1.25 s. que se obtienen en el presente trabajo.

Los resultados presentados por Paul Ortiz en ángulos son los siguientes. En la figura 48 muestra el comportamiento del controlador en donde se puede observar algunas características como: el sistema es estable a lo largo del tiempo, el tiempo de subida es $T_r = 1$ s., el tiempo de establecimiento es $T_s = 3$ s. para los ángulos de Roll y Pitch, por otro lado, para el ángulo de Yaw $T_s = 10$ s.

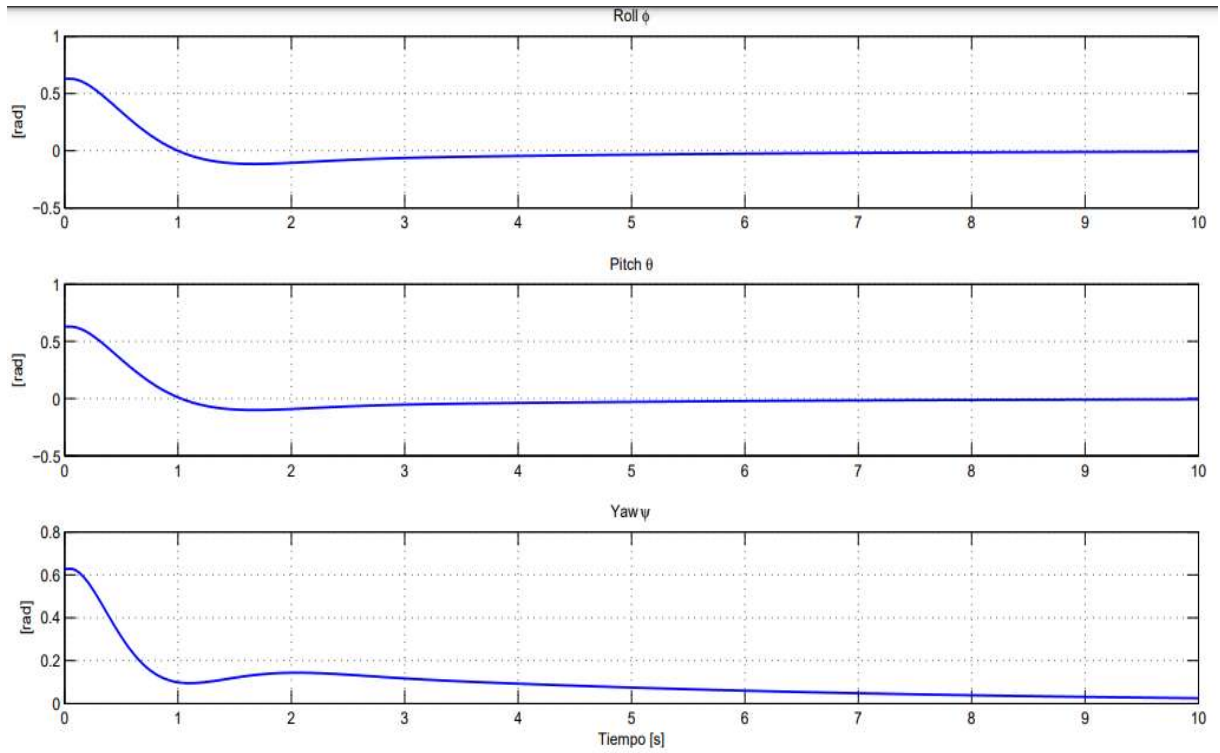


Figura 48. Controles de orientación Paul Ortiz

En su trabajo la prueba del control de orientación es iniciar la simulación con ángulo de $\frac{\pi}{5}$ donde tiene como referencia 0° , por lo que no se pueden comparar trabajos en el ningún ángulo.

Capítulo 6

6.1 Conclusiones

En el actual trabajo se utilizó la metodología de algoritmos genéticos para la obtención de controladores para un cuadricóptero. Tal metodología permite obtener controladores aceptables para su implementación; además el uso de un índice de desempeño modificado al citado por la bibliografía aporta una mejor respuesta al sistema de manera general, aunque este método aún tiene que ser mejorado para que así a su vez se mejore la respuesta obtenida, Una desventaja del procedimiento realizado en esta tesis es que se requiere una ecuación del sistema para cualquier sistema que se intente sintonizar controladores.

En la sintonización del ángulo Roll, Pitch y Yaw se pudo observar que al momento de variar la entrada de cualquier ángulo la altura variaba aunque este estuviera entre -20° y 20° , por lo contrario del trabajo donde se obtuvieron los parámetros del cuadricóptero simulado Paul Ortiz 2014, y distintos trabajos citados (Verónica Ortiz & Pablo Pulla, S. F. Antonio Pico Villalpando, 2012, Michell P. Ehrlich & Luis F. Lupian, 2015, Alejandro Antona Castañares, 2018, Antonio Pico, 2012) donde se hicieron linealizaciones de las fórmulas del cuadricóptero para poder obtener funciones de transferencia y sintonizar estas sin tomar en cuenta que provocan variaciones en el cuadricóptero, en las pruebas simuladas debido a que las funciones habían sido linealizadas no mostraban alguna variación en otra variable pero al momento de implementar el cuadricóptero tenían variaciones en el desempeño de este debido a estas linealizaciones como Antonio Pico en 2012 donde las pruebas al momento de variar la inclinación del cuadricóptero este presentaba perturbaciones así como lo presentado por Luis Romero Paredes en 2014 donde sus pruebas simuladas tenían valores muy buenos y los sistemas llegaban a la referencia en menos de un segundo no obstante al implementarlo tardaba 14 s en estabilizarse el cuadricóptero, para el caso de la tesis citada de Paúl Ortiz Gonzales (2014) los resultados de su controlador PID son similares a los obtenidos en este tesis donde él tiene un sobre impulso del 30% en comparación al casi 9% obtenidos por medio del algoritmos genéticos, para el caso de los ángulos Roll Pitch y Yaw no hay pruebas donde se cambie la referencia solo pruebas donde el cuadricóptero al momento de inicio de la

simulación el ángulo era distinto a 0 y la referencia era 0 y este es llevado a la referencia en aproximadamente 3 s, pero sin oscilaciones.

Referencias

- Alejandro Antona Castaños, Identificación, Modelado y Control de un Cuadricóptero de Bajo Coste, Universidad Técnica Superior de Ingenieros Industriales, Universidad Politécnica de Madrid, 2018.
- Antonio Pico Villalpando, Diseño e implementación de un sistema de control para un cuadricóptero, Centro De Investigación y De Estudios Avanzados Del Instituto Politécnico Nacional, 2012.
- Didier Giraldo Buitrago y Eduardo Giraldo Suárez, Teoría De Control Análogo, Colombia, Edición 1, 2009.
- Eduardo Giraldo Suarez, Pablo Andrés Muñoz Gutiérrez y Jhon Abraham Bonilla Becerra, Identificación y Control de un Vehículo Aéreo no Tripulado tipo Quadcopter, Ingeniería USBMed, Volumen 7, Numero 1, 2016.
- Edwin Dimitri Nieto Guerrero, Fernando Alonso Vaca De La Torre, Desarrollo de un modelo matemático, cinemático y dinámico con la aplicación de software, para modificar el funcionamiento de un dron, para que este realice monitoreo automático. Revista científica mundo de la investigación y el conocimiento, Editorial saberes del conocimiento, paginas 332-343. URL: <http://recimundo.com/index.php/es/article/view/814>, 2020.
- Isidro Ignacio Lázaro Castillo, Ingeniería de sistemas de control continuo, Morelia Michoacán, México, Editorial 2, 2019
- Javier Hernández Benítez. Polinomios Ortogonales, Funciones de Transferencia y Transformaciones Espectrales. Madrid España, Universidad Carlos III de Madrid departamento de matemáticas. Diciembre 2006.
- Jhon Abraham Bonilla Becerra, Control Acoplado de sistemas Multivariables Acoplado a Vehículos Aéreos no Tripulados, Universidad Tecnológica de Pereira, Facultad de ingenierías, 2019.
- JL Mendoza-Soto, Sintonización PID usando Algoritmos Genéticos, URL <https://www.youtube.co2020.m/watch?v=CwI1lNrmYZI>, 2020.
- Katsuhiko Ogata, Ingeniería de control moderna, Madrid, España, Editorial 5, 2019.

- L. Desbrough, R. Miller: Increasing customer value of industrial control performance monitoring – Honeywell's experience. Sixth International Conference on Chemical Process Control, AIChE Symposium Series Number 326, vol. 98, 2002.
- Lucía Cervantes Gómez, Modelización matemática Principios y aplicaciones, Puebla, México, primera edición Benemérita Universidad Autónoma de Puebla, 2015
- Luis Eduardo Romero Paredes, Diseño y construcción de un módulo de control PID para la estabilización de un cuadricóptero, 2014.
- Martín Medina Diaz, Algebra lineal numérica, Universidad de Tarapaca, Facultad de ciencias, departamento de matemáticas. S.F.
- MatLab. MathWorks. URL <https://la.mathworks.com/products/matlab.html>, 2022.
- Othon Colorado Arellano, Norberto Hernández Romero, Juan Carlos Seck Tuoh Mora, Joselito Medina Marín, Algoritmo genético aplicado a la sintonización de un controlador PID para un sistema acoplado de tanques, Boletín Científico del Instituto de Ciencias Básicas e Ingeniería, Número 10(2018)50-55, 2018.
- Paúl Ortiz Gonzales, Modelado y Control Robusto de un vehículo Aéreo no Tripulado Quadotor, En ambientes cerrados, Universidad Politécnica Salesiana. 2014.
- Roger Miranda Colorado, Rubén Alejandro Garrido Moctezuma, Luis Tupak Aguilar Bustos & José Ernesto Herrero Brito, Drones, Modelado y Control de drones, México, Ed Alf Omega, 1º edición, 2018.

Apéndice 1. Toolbox de optimización de MatLab

Matlab es un lenguaje técnico computacional y un ambiente interactivo para el desarrollo de algoritmos, visualización y análisis de datos, y computación numérica. Es usado en una gran variedad de aplicaciones, como el procesamiento de señales e imágenes, comunicaciones, modelado y análisis financiero, entre otros. Se pueden añadir toolboxes, los cuales son colecciones de funciones para un propósito en especial, que extienden las funcionalidades de Matlab (Iván Cabezas & Juan David Pérez, 2010).

Uno de estos, es el toolbox de Optimización, el cual provee algoritmos para resolver diferentes tipos de problemas de optimización, con restricciones o sin ellas; los cuales incluyen programación lineal, programación cuadrática, minimización (maximización) de funciones de una o varias variables, solución de sistemas de ecuaciones no lineales, entre otros. El toolbox de optimización de Matlab es un conjunto de funciones (almacenadas en archivos del tipo m-files) que proveen algoritmos para solucionar diferentes tipos de problemas. Además, posee una interfaz gráfica que facilita a los usuarios menos experimentados usar fácilmente el toolbox, sin necesidad de hacer uso de código (Iván Cabezas & Juan David Pérez, 2010).

Optimization Toolbox™ proporciona funciones para hallar parámetros que minimicen o maximicen los objetivos y respeten las restricciones. Esta toolbox incluye funciones (solvers) para programación lineal (LP), programación lineal entera mixta (MILP), programación cuadrática (QP), programación cónica de segundo orden (SOCP), programación no lineal (NLP), mínimos cuadrados lineales con restricciones, mínimos cuadrados no lineales y ecuaciones no lineales (Mathworks, 2022).

Puede definir su problema de optimización con funciones y matrices, o bien especificando expresiones de variables que reflejen la matemática subyacente. Puede utilizar la diferenciación automática de funciones objetivo y de restricción para obtener soluciones más rápidas y precisas (Mathworks, 2022).

Puede utilizar los solvers de la toolbox, para hallar las soluciones óptimas a problemas continuos y discretos, realizar análisis de compensación e incorporar métodos de optimización en algoritmos y aplicaciones. La toolbox permite realizar tareas de optimización de diseños, incluidas la estimación de parámetros, la selección de componentes

y el ajuste de parámetros Permite hallar las soluciones óptimas en aplicaciones tales como optimización de carteras, gestión y trading de energía, y planificación de producción (Mathworks, 2022).

Para el funcionamiento del toolbox de MatLab se requiere conocer los siguientes conceptos (Mathworks, 2022):

- La función fitness: Es la función que se desea optimizar, también conocido como la función objetivo. El objetivo del toolbox es encontrar el mínimo de la función objetivo.
- Individuos: Es cualquier punto que se pueda aplicar en la función fitness y cada individuo tiene un puntaje o valor, el cual se trata de que sea el mínimo posible.
- Población y generaciones: es un arreglo de individuos, por ejemplo, si el tamaño de la población es de 100 y el número de variables 3 en la función fitness, la población es representada por una matriz 100×3 . El mismo individuo puede aparecer más de una vez en la población. En cada iteración el algoritmo genético realiza una serie de cálculos en la población actual para producir una nueva población.
- Diversidad: Se refiere al promedio de distancia entre los individuos en una población. Una población tiene gran diversidad si el promedio de distancia es largo, de otra manera tiene una baja diversidad si el promedio de distancia es corto.
- Valores fitness y mejores valores fitness: El valor fitness es el valor de la función para esos individuos, debido a que la toolbox encuentra el valor mínimo, el mejor valor fitness para una población es el valor más pequeño fitness para cualquier individuo en la población.
- Padres e hijos: Para crear la siguiente generación, el algoritmo genético selecciona ciertos individuos en la población actual, llamados padres, y los usa para crear individuos en la siguiente generación, llamados hijos.

Algoritmo de funcionamiento:

1. El algoritmo inicia por crear una población inicial
2. El algoritmo después de crear una secuencia de población nueva. A cada paso, el algoritmo usa los individuos en la generación actual para crear la siguiente población. Para crear la siguiente población el algoritmo realiza los siguientes pasos:

- 2.1 Califica cada miembro de la población actual por cálculos su valor fitness. Esos valores son llamados fila de calificaciones fitness.
- 2.2 Escala la fila de calificaciones fitness para convertir los en un rango en valores más usables. Esos valores escalados son llamados valores expectación.
- 2.3 Selecciona los miembros, llamados padres, basados en su expectación
- 2.4 Algunos de los individuos en la población actual que tienen menor fitness son elegidos elite. Esos individuos elite son pasados a la siguiente población.
- 2.5 Los hijos producidos de los padres. Hijos son producidos ya sea por hacer cambios aleatorios a un simple padre (mutación) o por combinación de vectores entradas de un par de padres (crossover).
- 2.6 Remplaza la población actual con los hijos de la siguiente generación.
3. El algoritmo se detiene cuando uno de los criterios es conocido.
4. El algoritmo toma modificaciones en los pasos para restricciones lineales y enteros.
5. El algoritmo es ampliamente modificado para restricciones no lineales.

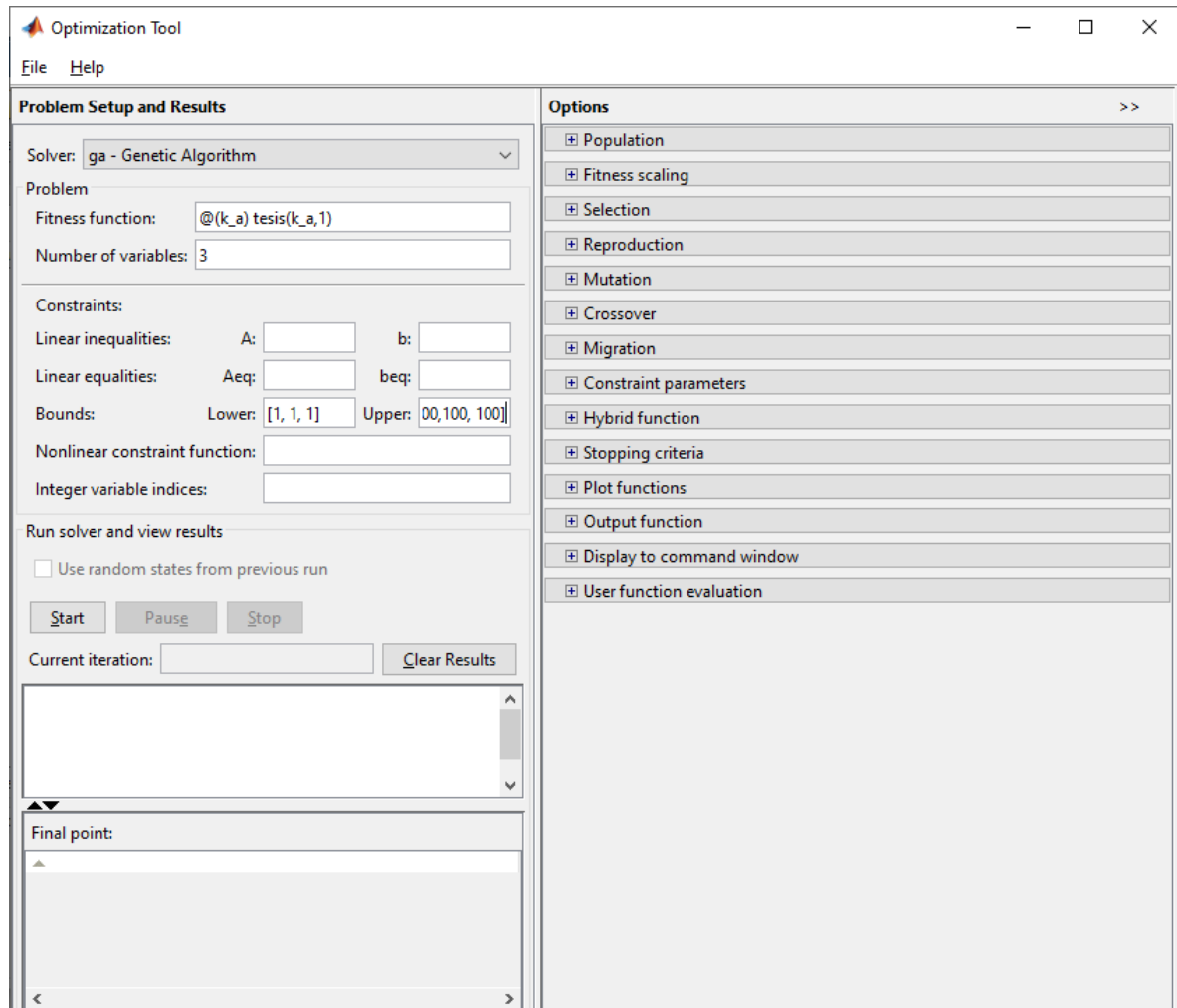
Uso del Toolbox

Para el uso del toolbox se basó en el video en Youtube Sintonización PID usando Algoritmos Genéticos, del usuario JL Mendoza-Soto publicado en 2020.

El cual describe el proceso para la utilización del toolbox, los pasos son los siguientes

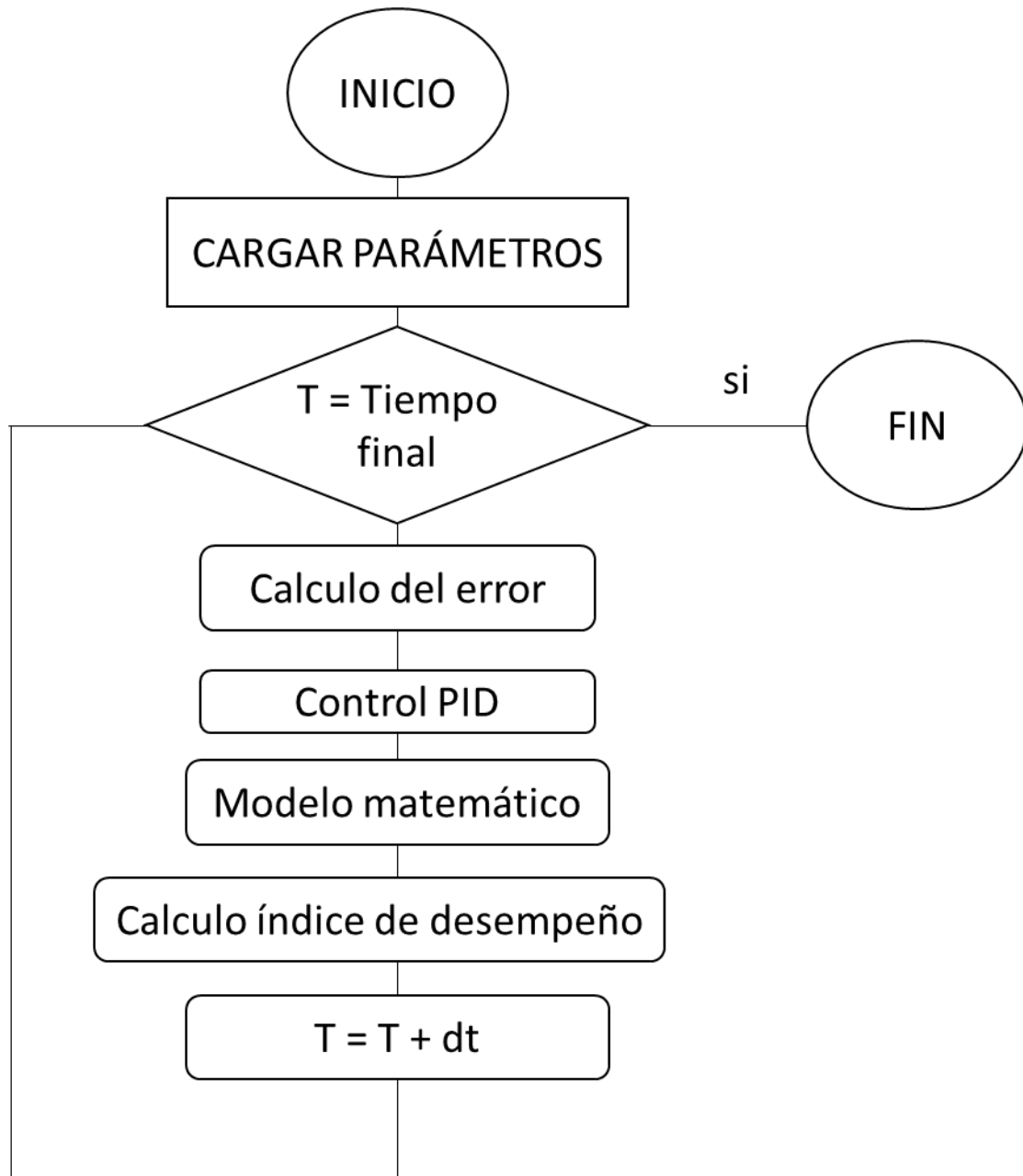
1. Ejecutar MatLab
2. Abrir la aplicación en el menú>aplicaciones>optimization toolbox
3. En la sección solver elegir GA- genethic algoritm
4. En fitnes function colocar @ y entre paréntesis colocar el nombre de la variable o vector de las variables a encontrar, y enseguida de un espacio el nombre de la función entre paréntesis los parámetros para que la función trabaje, en estos parámetros debe estar el nombre de la variable a encontrar.
5. En numero de variables colocar el número de variables a encontrar.
6. En la parte constraints colocar en bounds las ganancias mínimas y máximas consecutivamente.
7. En la parte de run solver and views results presionar el botón de start para iniciar el algoritmo de GA

En la parte derecha del toolbox está la parte de options esta sección está diseñada para colocar un parámetro del algoritmo específico, pero también tiene unos predeterminados.



Anexo 1 Función de MatLab

Diagrama de flujo de función de MatLab



```

function [J_h] = tesis3(k_a, h)

%% % variables de entrada%%
rad = 180/pi; max_f = 6.0; masa_motor = .0776; masa_helice = .01367;
masa_cuerpo = 1.02525; masa_bateria = .28462;
m = 4*masa_motor + 4*masa_helice + masa_bateria + masa_cuerpo;
tq = 0.0636; km = tq; I_xx = 0.0116; I_yy = 0.0116; I_zz = 0.0232;
L = 0.28; g = 9.81;
a1 = 0.1;
a2 = 0.9;
%%Ganancias controladores
K = zeros(3, 4);
K(1,:) = [1, 1, 1, 1];
K(:,h) = k_a; % control a sintonizar
u = zeros(1,4); %señal de control

%%Condiciones iniciales
J= [0, 0, 0, 0]'; error_a = [0, 0, 0, 0]';
Xl_ini = [0, 0, 0, 0]'; Xl_i = [0, 0, 0, 0]';
M_u=[0, 0, 0, 0]'; %%fuerzas u1 u2 u3 y u4
y = [0,0,0,0]'; n = 1;
y_a=[0, 0, 0, 0]'; y_an = [0, 0, 0, 0]';
vel_a = [0, 0, 0, 0]'; vel_an = [0, 0, 0, 0]';
posicion_a = [0, 0, 0, 0]';
%%tiempo de trabajo
t = 8;
dt = 0.001;
ti = 0;
resultados = zeros(4, t/dt);
resultados_vel = zeros(4, t/dt);
errores = zeros(4, t/dt);
ref = [ ];
%% señal de entrada
in = [1, 0, 0, 0]'; error_ai = [0, 0, 0, 0]';
error_h = [0, 0, 0, 0];
for tiempo=ti:dt:t

    error = in - y;
    errores(:,n) = error;
    error_ai = error_ai + error.*dt;
    %%accion de contro u
    for q = 1: 4
        u(q) = K(:,q)'*[error(q), error_ai(q), ((error(q)-
            error_a(q))/dt)]';
    end
    u(1) = 0.25*((u(1)) + g);
    if u(1)<0.5
        u(1) = 0.5;
    end
    %%COMBINACION CONTROL
    U = [(u(1)+u(2)+u(4)), (u(1)+u(3)-u(4)), (u(1)-u(2)+u(4)), (u(1)-
u(3)-u(4))]'';
    %%indice de desempeño
    if tiempo>=4
        J = J + (error.^2 + error_a.^2).*dt./2;
    end
end

```

```

error_a = error;
%%modelo de motores
x1p = 4.875.*U - 5*X1_i;
X1_i = X1_i + (X1_ini + x1p)*dt/2;
X1_ini = x1p;
% % Saturador de motores
for a = 1:4
    if X1_i(a) > 6
        X1_i(a) = 6;
    end
end
% % combinación fuerza de motores
M_u(1) = [1, 1, 1, 1]* X1_i;
M_u(2) = L*([1, 1]*[X1_i(1), -X1_i(3)]');
M_u(3) = L*([1, 1]*[X1_i(2), -X1_i(4)]');
M_u(4) = L*tq*([1, -1, 1, -1]*X1_i);
% % fuerzas resultantes en matriz
resultados_vel(1,n) = M_u(1);
resultados_vel(2,n) = M_u(2);
resultados_vel(3,n) = M_u(3)';
resultados_vel(4,n) = M_u(4)';
%%MODELO CUADRICOPTERO
%%aceleraciones
y_a(1) = ((1/m)*cos(posicion_a(2))*cos(posicion_a(3))*M_u(1)) - g +
error_h(1);
y_a(2) = ((I_yy-I_zz)*vel_a(3)*vel_a(4)/I_xx + vel_a(3)*vel_a(4) +
M_u(2)/I_xx + error_h(2));
y_a(3) = (I_zz-I_xx)*vel_a(2)*vel_a(4)/I_yy - vel_a(2)*vel_a(4) +
M_u(3)/I_yy + error_h(3);
y_a(4) = (I_xx-I_yy)*vel_a(2)*vel_a(3)/I_zz - vel_a(2)*vel_a(3) +
M_u(4)/I_zz + error_h(4);
%%correcciones
vel_a = vel_a + ((y_a+ y_an) * dt/2);
y_an = y_a;
posicion_a = posicion_a + ((vel_a + vel_an)*dt/2);
vel_an = vel_a;
%%velocidades
% % posicioes
resultados(:,n) = posicion_a;
y = posicion_a;
n = n+1;
end
J_h = a1*J(1)+a2*J(h)/in(h);
end

```