

ESCUELA POLITÉCNICA NACIONAL

**FACULTAD DE INGENIERÍA ELÉCTRICA Y
ELECTRÓNICA**

**DISEÑO E IMPLEMENTACION DE MINIDRONES (MINI UAVS)
DESARROLLO Y SIMULACIÓN DE ALGORITMOS DE DESPEGUE
Y ATERRIZAJE DE MINIDRONES**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERA EN
ELECTRÓNICA Y AUTOMATIZACIÓN**

EVELYN ALEXANDRA IZA BARRIONUEVO

evelyn.iza@epn.edu.ec

DIRECTOR: ING. JORGE ANDRÉS ROSALES ACOSTA, PhD

andres.rosales@epn.edu.ec

DMQ, octubre 2022

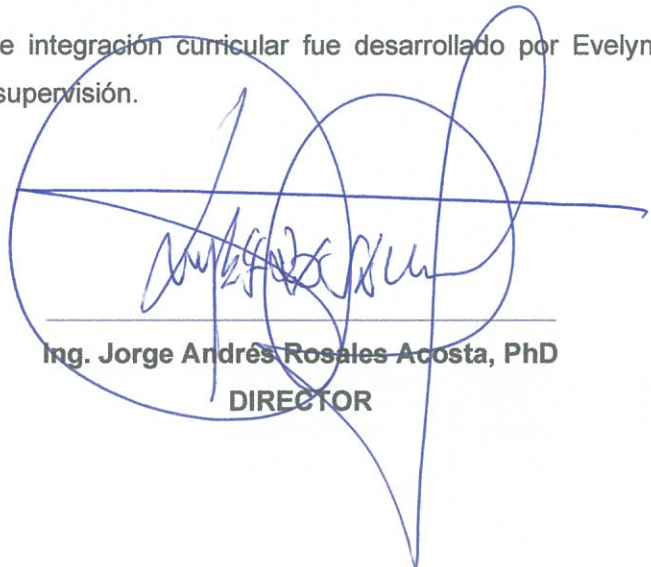
CERTIFICACIONES

Yo, Evelyn Alexandra Iza Barrionuevo declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



Evelyn Alexandra Iza Barrionuevo

Certifico que el presente trabajo de integración curricular fue desarrollado por Evelyn Alexandra Iza Barrionuevo, bajo mi supervisión.



Ing. Jorge Andrés Rosales Acosta, PhD
DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

Evelyn Alexandra Iza Barrionuevo

Ing. Jorge Andrés Rosales Acosta, PhD

DEDICATORIA

A mis padres Carlos y Carmen, que durante toda mi vida universitaria y hasta ahora me han sabido apoyar, entender y guiar mi camino de tal forma que no decaiga ante ninguna adversidad, sino que siga adelante con más fuerzas siendo ellos mismos quienes me dan la fortaleza y ganas para luchar.

A mis hermanas Carla y Valeria que de igual forma han estado presente con sus consejos y ayuda en todos los ámbitos universitarios y diarios.

AGRADECIMIENTO

Agradezco a Dios y a mis padres, Carlos y Carmen por ser el pilar fundamental en mi vida, que, con sus enseñanzas, paciència, consejos y lucha constante, he podido alcanzar esta meta y llegar a ser persona que soy.

A mis hermanas agradezco su paciència y ayuda que ha sido recibido en los momentos indicados y muchas veces han sido un gran ejemplo.

A mis amigos, que siempre han sabido darme su apoyo en todo momento.

A mis ingenieros, que con sus enseñanzas he podido llegar a este momento y en especial al Ing. Andrés Rosales que me ha orientado para la elaboración de este trabajo

ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN	VII
ABSTRACT	VIII
1. INTRODUCCIÓN.....	1
1.1 OBJETIVO GENERAL	2
1.2 OBJETIVOS ESPECÍFICOS	2
1.3 ALCANCE	2
1.4 MARCO TEÓRICO.....	3
1.4.1 VEHICULOS AÉREOS NO TRIPULADOS (UAVS).....	3
1.4.2 CLASIFICACION DE LOS UAVS	4
1.4.3 CUADRICOPTERO	6
1.4.4 ALGORITMOS DE CONTROL	9
1.4.5 INDICES DE DESEMPEÑO PARA CONTROLADORES	11
2. METODOLOGÍA.....	12
2.1 IMPLEMENTACION DEL MODELO MATEMATICO DEL CUADRICOPTERO.....	13
2.2 DISEÑO DEL CONTROLADOR PID.....	17
2.3 DISEÑO DEL CONTROLADOR PD.....	19
2.4 DISEÑO DEL CONTROLADOR BASADO EN LOGICA DIFUSA	21
3. RESULTADOS, CONCLUSIONES Y RECOMENDACIONES.....	27
3.1 RESULTADOS.....	27
3.1.1 REFERENCIA DE ANGULO Y ALTURA CONSTANTE	27
3.1.2 REFERENCIA DE ANGULO CONSTANTE Y ALTURA VARIABLE. 31	
3.1.3 REFERENCIA DE ANGULO VARIABLE Y ALTURA CONSTANTE. 36	
3.2 CONCLUSIONES.....	40
3.3 RECOMENDACIONES	41

4. REFERENCIAS BIBLIOGRÁFICAS	42
5. ANEXOS.....	44
ANEXO I. ECUACIONES DEL MODELO MATEMÁTICO EN MATLAB FUNCTION	45
ANEXO II. ESQUEMA DE CONTROL PID	46
ANEXO III. ESQUEMA DE CONTROL PD	47
ANEXO IV. ESQUEMA DE CONTROL DIFUSO	48

RESUMEN

El presente Trabajo de Integración Curricular muestra el diseño, implementación y simulación de tres algoritmos para el control de aterrizaje y despegue de mini UAV (Unmanned Aerial Vehicle), específicamente del cuadricóptero debido a que presenta mayor estabilidad y facilidad de control, para lo cual se hace uso de un modelo matemático conformado por el subsistema de rotación que corresponde a los ángulos roll, pitch y yaw; y el subsistema de translación que corresponde al movimiento en el eje x, y, z.

Al interesarnos principalmente la variable de altura, se toma en cuenta solo las ecuaciones del subsistema de rotación y del movimiento en el eje Z, por lo tanto, se van a implementar cuatro controladores. Los controladores por implementar son PID, PD y controlador basado en lógica difusa.

Para la simulación de todo el trabajo se utilizará el Software MATLAB/Simulink donde se realizan pruebas ante diferentes cambios de referencia de ángulo y altura, así también como se obtendrá los índices de desempeño ISE e ISCO para los controladores de la altura.

PALABRAS CLAVE: altura, aterrizaje , despegue, lógica difusa, PID, PD,

ABSTRACT

This document shows the design, implementation, and simulation of three algorithms for the control of landing and takeoff of mini UAV (Unmanned Aerial Vehicle), specifically the quadcopter because it presents greater stability and ease of control, for this reason the selected mathematical model consists of two subsystems: the rotation subsystem corresponding to the roll, pitch and yaw angles; and the translation subsystem corresponding to the movement in the x, y, and z axis.

Since the focus of the work is the height variable, only the equations of the rotation subsystem and the Z-axis motion are considered, therefore, four controllers are implemented. The implemented controllers are PID, PD and fuzzy logic controller.

For the simulation of the mathematical model and controller, MATLAB/Simulink software is used, and tests are performed for different changes of angle and altitude reference, as well as to obtain the ISE and ISCO performance indexes for the altitude controllers.

KEYWORDS: altitude, landing, takeoff, fuzzy logic , PID, PD

1. INTRODUCCIÓN

Actualmente los mini UAVS o también conocidos como drones son vehículos aéreos con la capacidad de vuelo sin tripulación. Estos han ido ganando mayor campo e interés en las diferentes líneas de investigadores e industrias gracias a sus características de tamaño, rapidez, infraestructura, capacidad de movilidad en lugares cerrados, algunos silenciosos, bajo costo; permitiendo su uso en diversas aplicaciones en áreas como: construcción, agricultura, medio ambiente, militar, seguridad vial, inspecciones de seguridad industrial, entre otras. [1]

Existe diversas formas de clasificar a los drones, sin embargo, el que más sobresale de entre todos es el cuadricóptero, que es un vehículo conformado por cuatro rotores en forma de cruz o x, cuenta con la capacidad de realizar el despegue y aterrizaje de forma vertical, tienen seis grados de libertad lo que permiten mayor maniobrabilidad, control y modelado. El uso del cuadricóptero es amplio y si nos referimos al aspecto de seguridad, es importante que el despegue sea preciso y rápido para poder seguir una trayectoria de monitoreo o se realice un aterrizaje preciso para conocer de forma rápida alguna emergencia. El aterrizaje y despegue se logra cuando los cuatro rotores giran a una misma velocidad, es decir al aumentar la velocidad en la misma magnitud el cuadricóptero se eleva, si una vez elevado se mantienen a una velocidad constante el dron se queda suspendido en el aire y si por el contrario empiezan a disminuir la velocidad en la misma magnitud el dron empieza a descender. En estas etapas el problema radica en cuando se producen cambios bruscos de velocidad ocasionan desestabilización del cuadricóptero y retardos para alcanzar una altura deseada o un aterrizaje suave sin que sufra daños.

El presente trabajo se enfoca en controlar las variables que influyen en el aterrizaje y despegue del cuadricóptero mediante el diseño e implementación de tres controladores a aplicarse el modelo matemático, el cual se obtiene a partir del modelo dinámico del cuadricóptero. Los controladores por implementar son el PID, PD y el basado en lógica difusa.

El modelo matemático cuenta con dos subsistemas: rotación (ϕ, θ, ψ) y translación (x, y, z) y de este último solo nos interesa la respuesta en el eje Z, por lo tanto, se implementan cuatro controladores PID para el ángulo de roll, pitch, yaw y la altura (eje Z). También se implementa cuatro controladores PI para las mismas variables mencionadas anterior y finalmente se diseña dos controladores basados en lógica difusa para roll y pitch y dos controladores para yaw y la altura. El controlador basado en lógica difusa se establecen

dos entradas con cinco funciones de membresía cada una y una salida con cinco funciones de membresías y tendrán una base de veinte y cinco reglas.

Una vez diseñados los controladores, tanto estos como el modelo matemático serán implementados y simulados en Matlab Simulink mediante bloques donde se realizarán pruebas ante valores variados del ángulo pitch, roll, yaw y altura. Se obtendrá los índices de desempeño ISE e ISCO del controlador de la altura para realizar un análisis del controlador mas óptimo.

1.1 OBJETIVO GENERAL

Diseñar, simular y comparar diferentes algoritmos, para el despegue y aterrizaje de un mini drone tipo cuadricóptero con el fin de obtener un sistema de control para aplicaciones de seguridad.

1.2 OBJETIVOS ESPECÍFICOS

1. Estudiar los diferentes tipos de mini drones existentes y principalmente las variables que intervienen durante el despegue y aterrizaje de este.
2. Estudiar el funcionamiento y las variables de un cuadricóptero y posteriormente seleccionar un modelo matemático.
3. Diseñar tres algoritmos de despegue y tres algoritmos de aterrizaje: PID, PI y controlador basado en lógica difusa.
4. Simular el modelo matemático del cuadricóptero e implementar los controladores en el software Simulink de Matlab y así comparar y analizar los resultados antes diferentes pruebas y mediante los índices de desempeño.

1.3 ALCANCE

Se realizará el estudio del funcionamiento y variables de un cuadricóptero para así entender el modelo matemático seleccionado.

Se realizará el estudio de los algoritmos PID y PI aplicados al control de mini drones basándose en la sintonización en lazo abierto.

Se realizará el estudio de la lógica difusa aplicada al control de mini drones y como realizar su implementación.

Se revisará y validará el modelo matemático del mini drone (cuadricóptero) variando las condiciones iniciales.

Se desarrollarán los controladores propuestos, mediante los métodos de sintonización en lazo abierto y posteriormente se utilizará tablas de sintonización para obtener los parámetros.

Se diseñará el controlador basado en lógica difusa.

Se utilizará el software de simulación Matlab – Simulink para realizar la implementación del modelo matemático con sus respectivos controladores.

Se realizarán pruebas para todos los controladores con distintas condiciones de funcionamiento, es decir, por ejemplo, diferentes condiciones de altura y aplicando diferentes ángulos de movimiento en lo que respecta a roll, pitch y yaw.

Se evaluarán índices de desempeño como el ISE e ISCO.

1.4 MARCO TEÓRICO

Este capítulo va a constar de una explicación sobre los vehículos aéreos no tripulados y en general sobre los mini UAVS, una clasificación de estos y principalmente se va a explicar sobre los cuadricópteros así también como algoritmos de control que se va a aplicar al mini UAV.

1.4.1 VEHICULOS AÉREOS NO TRIPULADOS (UAVS)

Los vehículos aéreos no tripulados, por sus siglas UAVS, también conocidos como drones, se pueden definir como una aeronave que realizan su vuelo sin ningún tripulante a bordo. Generalmente estos suelen ser manejados de forma remota desde una base mediante joystick o actualmente se están implementando algoritmos que permitan controlar el despegue, aterrizaje y seguimiento de trayectoria sin necesidad de la presencia humana durante todo su vuelo.

Al mencionar los UAVS, ingresamos a un área de estudio extensa donde se puede identificar gran variedad de estos, iniciando por las diferentes dimensiones en la que se los puede encontrar y aquí es donde se identifica a los conocidos mini drones, los cuales se dice deben tener un peso base menor a 600 gramos y tener la capacidad para colocarlos

en la palma de la mano para ser considerados dentro de este grupo. Hablando comercialmente ya existe diversos mini drones tales como: Tech rc Drone, Atoyx Mini Dronde, RC Drone, Parrot Mambo y más. [2]

Los mini drones tienen características similares a las de un drone común, por lo que están contruidos con materiales de bajo peso, capacidad de largos periodos de vuelo, bajo costo y facilita la vida del ser humanos en algunas aplicaciones. A continuación, se presenta algunas ventajas y desventajas.

Ventajas de usar de mini drones en diferentes aplicaciones:

- Al tener un tamaño reducido tienen la capacidad de realizar su vuelo en lugares cerrados y de difícil acceso y así evitar riesgos para el ser humano.
- Debido a su tamaño su transporte o infraestructura no es muy necesaria debido a que fácilmente se lo puede lanzar desde una mano.
- Su despegue no genera mayor ruido.
- Contiene cámara que permitan tener visión diurna y nocturna
- Bajo costo
- Debido a su gran avance tecnológico ya se cuenta con una gran variedad de plataformas y aplicaciones donde se puede simular su vuelo y de esta forma entender los factores que afectan su vuelo de tal forma que más adelante realizar mediante programación o algoritmos de control una mejoría en su trayectoria.

Desventajas de usar de mini drones en diferentes aplicaciones:

- La mayor desventaja del mini drone es que al ser de menor peso son más propensos a desestabilización ante presencia de fuerte vientos.
- Tienden a tener baterías de poco almacenamiento por lo que en muchas ocasiones su tiempo de vuelo puede ser limitado.

1.4.2 CLASIFICACIÓN DE LOS UAVS

El siguiente trabajo en general trata sobre los mini drones, pero como antes se mencionó estos presentan características similares que un drone, por lo tanto, se puede tener la misma clasificación y esta puede ser de acuerdo con la aplicación que vaya a ser usado, según el método de control o la más común es de acuerdo con su tipo de ala. [3]

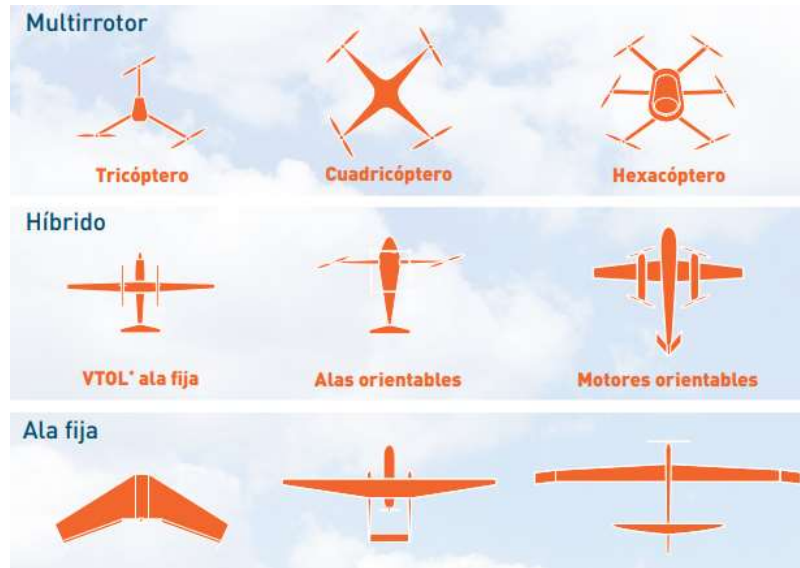


Figura 1.1. Clasificación de los drones de acuerdo con sus alas [1]

Dentro de la clasificación según el tipo de ala se tiene tres grandes grupos.

1.4.2.1 Ala fija

Este tipo de drones presenta en su estructura alas estáticas, siendo así muy similar en diseño y vuelo a un avión, por lo tanto, presentan mayor autonomía dando beneficios como mayor tiempo de vuelo y mayor altura, sin embargo, su mayor desventaja es que necesita de una pista para su despegue y aterrizaje ya que no puede realizarlo de forma vertical.

1.4.2.2 Multirrotor

Este tipo de drone se caracteriza por presentar en su estructura varias hélices o alas que giran en diferentes sentidos permitiendo así generar fuerza de empuje que nos permite mantener en el aire. Al tener varias alas va a tener varios rotores. Entre sus principales ventajas es que puede realizar un despegue y aterrizaje de forma vertical y tiene la capacidad de permanecer en un punto fijo, sin embargo, su mayor desventaja es que presenta poca autonomía.

En este grupo de acuerdo con su número de rotores se puede dividir en: tricópteros, cuadricópteros, hexacópteros u octacópteros.

1.4.2.3 Híbrido

Este tipo de dron es el menos conocido y en su estructura presenta tanto un ala fija, así como también como rotores, es básicamente una unión de las dos clasificaciones anteriormente mencionadas. Entonces entre sus ventajas se tiene que puede aterrizar y despegar de forma vertical y puede alcanzar elevadas alturas.

Otras clasificaciones mencionadas antes son según su uso que puede ser militar o civil y según su control: autónomo, control remoto o monitorizado.

1.4.3 CUADRICOPTERO

El cuadricóptero o también conocido como quadrotor pertenece al grupo de los multirrotores debido a que en su estructura presenta cuatro (4) rotores que permiten su vuelo, de estos rotores: 2 van a girar en el mismo sentido y 2 en sentido contrario, mediante esta primicia del giro de los rotores y la velocidad que se emplee se va a poder controlar la rotación, torque, desplazamiento y elevación del UAV. [4]

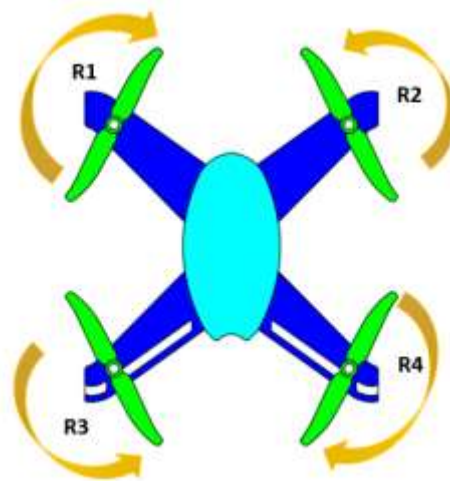


Figura 1.2. Cuadricóptero y el sentido de giro de sus alas

Estos UAVs presentan 4 movimientos básicos.

1.4.3.1 Movimiento de alabeo

Este movimiento también conocido como roll, consiste en que el cuadricóptero se desplazara de izquierda a derecha. Esta acción se va a producir cuando R1 y R3 se encuentren a baja velocidad y R2 Y R4 a alta velocidad o viceversa.

1.4.3.2 Movimiento de cabeceo

Este movimiento también conocido como pitch, consiste en que el cuadricóptero tendrá movimiento hacia adelante y atrás. Esta acción se va a producir cuando R1 y R2 se encuentren a baja velocidad y R3 Y R4 a alta velocidad o viceversa.

1.4.3.3 Movimiento de guiñada

Este movimiento también conocido como yaw, consiste en que el cuadricóptero podrá girar su eje vertical. Esta acción se va a producir cuando R1 y R4 se encuentren a baja velocidad y R2 Y R3 a alta velocidad o viceversa.

1.4.3.4 Movimiento de empuje o Throttle

Este movimiento consiste en que el cuadricóptero podrá ascender o descender de forma vertical. Esta acción se va a producir cuando todos los rotores aumenten o disminuyan su velocidad en igual magnitud.

1.4.3.5 Modelo Matemático

El modelo matemático en este trabajo es tomado de la tesis “Dynamic Model and Control of Quadrotor in the Presence of Uncertainties”, el mismo se basa en el modelo dinámico del cuadricóptero que presenta dos subsistemas, siendo el subsistema de rotación donde se tiene los ángulos de roll (ϕ), pitch (θ) y yaw (ψ); y el sistema de translación, el cual representa la posición del drone en el eje x-y-z.

Ambos subsistemas actúan conjuntamente

Para obtener el subsistema de rotación se parte de la ecuación del movimiento rotacional y aplicando el método de Newton-Euler se puede utilizar la siguiente formula generalizada:

$$M_b = (J\dot{w} + w \times Jw + M_g) \quad (1.1)$$

De acuerdo con la ecuación 1.1 se tiene:

J : representa la Matriz de inercial del cuadricóptero, siendo esta una matriz diagonal debido a que el drone es construido simétricamente con respecto a los sistemas de coordenadas, esta matriz es:

$$J = \begin{bmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{bmatrix} \quad (1.2)$$

w : representa la velocidad angular

M_g : es el momento giroscópico que se genera por la inercia de los rotores.

M_b : esta representa los momentos a aplicarse en el cuadricóptero y es representado por la siguiente ecuación:

$$M_b = \begin{bmatrix} U_1 l \\ U_2 l \\ U_3 l c \end{bmatrix} \quad (1.3)$$

Para simplificar la ecuación 1.1 se puede despreciar el momento giroscópico porque la inercia generada por el cuadricóptero es mayor que la inercia generada por el rotor. De tal forma que se va a obtener la siguiente ecuación:

$$M_b = (Jw + w \times Jw) \quad (1.4)$$

Reemplazando la ecuación 1.4 con la matriz de inercia y con las velocidades angulares se obtiene:

$$\begin{bmatrix} U_1 l \\ U_2 l \\ U_3 l c \end{bmatrix} = \begin{bmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (1.5)$$

Al reescribir la ecuación 1.5 se va a obtener las ecuaciones del subsistema de rotación, siendo las siguientes:

$$\ddot{\phi} = \frac{l}{J_x} U_1 + \frac{J_y}{J_x} \dot{\psi} \dot{\theta} - \frac{J_z}{J_x} \dot{\theta} \dot{\psi} \quad (1.6)$$

$$\ddot{\theta} = \frac{l}{J_y} U_2 + \frac{J_z}{J_y} \dot{\psi} \dot{\phi} - \frac{J_x}{J_y} \dot{\theta} \dot{\psi} \quad (1.7)$$

$$\ddot{\psi} = \frac{l}{J_z} U_3 + \frac{J_x}{J_z} \dot{\theta} \dot{\phi} - \frac{J_y}{J_z} \dot{\phi} \dot{\theta} \quad (1.8)$$

Para el subsistema de translación se obtiene a partir de la ecuación de movimiento de translación y junto con la segunda ley de Newton, que se deriva del marco inercial de la tierra se tiene la siguiente formula generalizada:

$$m\ddot{d} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + R \times F_{ng} \quad (1.9)$$

Donde: m es la masa, g es la gravedad $9.8 \frac{m}{s^2}$

F_{ng} : es la ubicación física dentro del sistema fijo terrestre y está dado por:

$$F_{ng} = \begin{bmatrix} 0 \\ 0 \\ -U_4 \end{bmatrix} \quad (1.10)$$

R es la matriz rotacional que se obtiene a partir de la matriz rotacional de cada uno de los ejes, obteniendo lo siguiente:

$$R = \begin{bmatrix} \sin\psi\cos\theta & \cos\psi\sin\theta\sin\phi & \sin\psi\sin\phi - \cos\psi\cos\phi\sin\theta \\ \sin\psi\cos\theta & \cos\psi\cos\phi + \sin\psi\sin\theta\sin\phi & -\cos\psi\sin\phi + \sin\theta\sin\psi\cos\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\phi\cos\theta \end{bmatrix} \quad (1.11)$$

Reemplazando la ecuación 1.9 se obtiene los siguiente:

$$m \begin{bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + \begin{bmatrix} \sin\psi\cos\theta & \cos\psi\sin\theta\sin\phi & \sin\psi\sin\phi - \cos\psi\cos\phi\sin\theta \\ \sin\psi\cos\theta & \cos\psi\cos\phi + \sin\psi\sin\theta\sin\phi & -\cos\psi\sin\phi + \sin\theta\sin\psi\cos\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\phi\cos\theta \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ -U_4 \end{bmatrix} \quad (1.12)$$

Reescribiendo la ecuación 1.12 se va a obtener la aceleración con respecto al eje x-y-z:

$$\ddot{X} = \frac{-U_4}{m} (-\cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi)) \quad (1.13)$$

$$\ddot{Y} = \frac{-U_4}{m} (\cos(\phi)\sin(\theta)\sin(\psi) + \sin(\phi)\cos(\psi)) \quad (1.14)$$

$$\ddot{Z} = \frac{-U_4}{m} (\cos(\phi)\cos(\theta)) - g \quad (1.15)$$

[5]

1.4.4 ALGORITMOS DE CONTROL

1.4.4.1 Controlador PID

Los controladores PID son los clásicos y más usados a nivel general. Estos están compuestos por tres bloques principales que corresponden a la parte proporcional, derivativa e integral.

Mediante la Figura 1.3, se observa el esquema del controlador en bloques y este compuesto por tres acciones

- Proporcional (P): Mediante esta acción el controlador busca una respuesta del sistema donde se tenga un menor error en el régimen estacionario así también como se tenga menor tiempo de reacción del sistema.
- Integral (I): Mediante esta acción se busca mayor estabilidad de la respuesta del sistema.
- Derivativa (D): Con esta se busca disminuir el error de la respuesta del sistema en régimen estacionario, sin embargo, tiende a aumentar la inestabilidad de la salida del sistema.

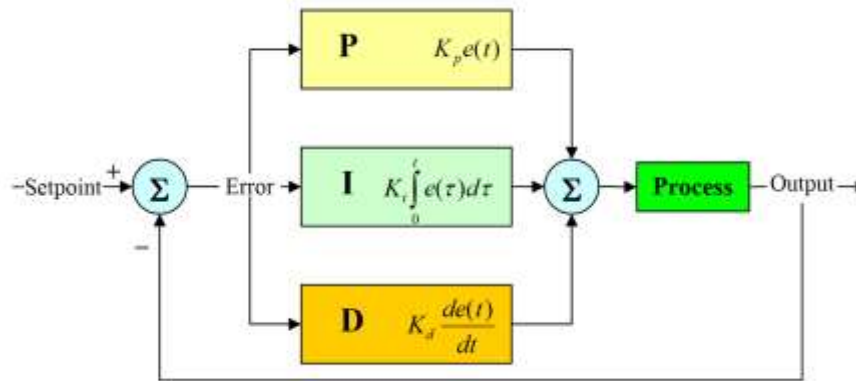


Figura 1.3. Esquema de un controlador PID

Mediante las tres acciones que se tiene, a estas se la puede combinar obteniendo más controladores como P, PI, PD y PID.

De todos los controladores mencionados para el presente trabajo se va a trabajar con los controladores PID y PD que se representan con las siguientes ecuaciones:

$$PID(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad (1.16)$$

$$PD(t) = K_p e(t) + K_d \frac{de(t)}{dt} \quad (1.17)$$

Donde:

K_p : es la ganancia proporcional

K_i : es la ganancia integral

K_d : es la ganancia derivativa

Estos parámetros son los que se deben calcular o ajustar para obtener un sistema de control óptimo.

1.4.4.2 Controlador Basado en Lógica Difusa

La lógica difusa o fuzzy actualmente es muy utilizada debido a que se considera una técnica de la Inteligencia Artificial, la cual tiene el propósito de manifestar el conocimiento común a través de una sintaxis lingüística para a su vez esta expresarse mediante un lenguaje matemático permitiendo a si mayor precisión a un sistema.

Este control se puede considerar interesante debido a que es útil tanto para modelos de plantas lineales como no lineales, siendo este último su mayor ventaja ante procesos complejos con presencia de datos no precisos.

En lógica difusa se puede implementar dos tipos de sistemas de control: Takagi Sugeno y Mamdani, siendo este último el más utilizado y su estructura es el de la Figura 1.4, donde se puede observar que está formado por cuatro bloques.

Fusificador: en esta etapa la interfaz de fusificación es la encargada de adquirir los valores de entrada y transformarlos a valores difusos y a su vez designa a cada variable un grado de pertenencia de acuerdo con cada conjunto difuso establecido.

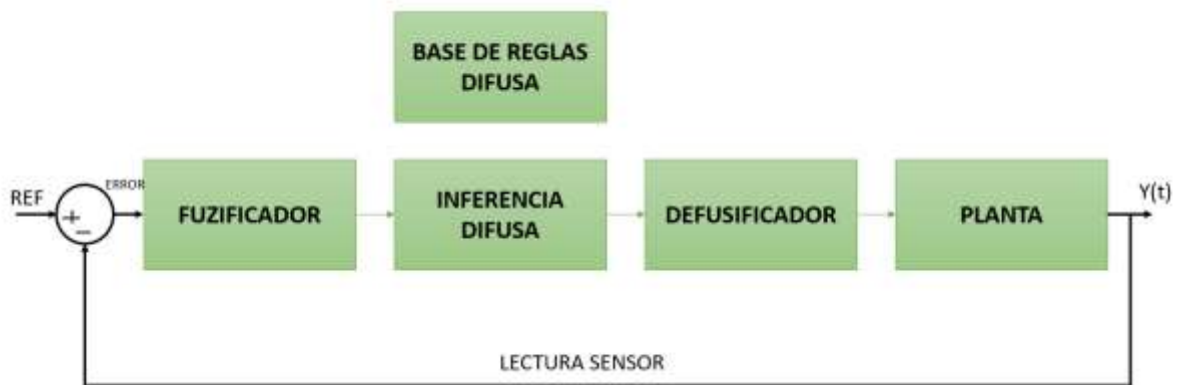


Figura 1.4. Estructura clásica de un control difuso

Inferencia difusa: esta etapa es la encargada de interpretar los valores de un vector de entrada y juntamente con las reglas lingüísticas se asigna un vector de salida.

Defusificador: esta etapa se considera el proceso inverso de la fusificación, es decir transforma los conjuntos difusos de salida en un valor real aceptable para la planta. Para esta etapa se utilizan diferentes métodos como: centroide, membresía del medio máximo o promedio ponderado.

Base de reglas difusas: Este bloque se refiere básicamente al conocimiento que se tenga de la planta y a partir de esto se va a establecer las reglas lingüísticas. [6]

1.4.5 INDICES DE DESEMPEÑO PARA CONTROLADORES

Al implementar los controladores en un sistema para comprobar su correcto funcionamiento se suele usar los índices de desempeño, los cuales determinan la optimización y su desempeño en el sistema. Estos suelen ser los índices ISE e IAE.

1.4.5.1 Integrador de error cuadrático ISE

El ISE determina la optimización del controlador con respecto al error del sistema en estado transitorio.

Su fórmula es la siguiente:

$$ISE = \int_0^{\infty} e(t)^2 dt \quad (1.18)$$

1.4.5.2 Integrador del valor absoluto del error IAE

El IAE determina de forma cuantitativa los mínimos errores presentes al implementar un controlador.

La fórmula para determinar es:

$$IAE = \int_0^{\infty} |e(t)| dt \quad (1.19)$$

2. METODOLOGÍA

Este capítulo presenta la implementación del modelo matemático en Simulink y el diseño de los tres algoritmos de control para el aterrizaje y despegue del mini dron.

En esta etapa al trabajar con MATLAB – SIMULINK se cuenta con una amplia librería donde en especial se van a usar 3 bloques que nos van a permitir la implementación de las ecuaciones del modelo matemático, así como también la simulación de los controladores. Estos bloques son los siguientes:

Bloque de MATLAB Function, donde se va a poder ingresar las ecuaciones del modelo matemático sin necesidad del uso de otros bloques.



Figura 2.1 Bloque MATLAB Function

Bloque de PID Controller, permite ingresar los valores de ganancia proporcional, integral y derivativo, generando automáticamente el controlador PID sin la necesidad de otros bloques. También permite obtener un controlador PI, PD, P, D, I.



Figura 2.2 Bloque MATLAB Function

Bloque de Fuzzy Logic Controller, este bloque permite generar un controlador basado en lógica difusa siempre y cuando primero se haya ingresado los valores del controlador mediante el toolbox Fuzzy Logic.

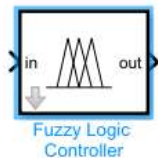


Figura 2.3 Bloque Fuzzy Logic Controller

2.1 IMPLEMENTACIÓN DEL MODELO MATEMATICO DEL CUADRICOPTERO

Para la implementación del modelo matemático del cuadricóptero se va a usar las ecuaciones del subsistema de rotación (1.6, 1.7, 1.8) y la ecuación de aceleración respecto al eje Z del subsistema de traslación (1.15), debido a que el presente trabajo trata sobre algoritmos de despegue y aterrizaje interesándonos principalmente el movimiento del cuadricóptero en el eje z.

Los parámetros para utilizar en las ecuaciones son:

Tabla 2.1. Parámetros del Cuadricóptero

Símbolo	Descripción	Valor
m	Masa cuadricóptero [kg]	0.6
l	Longitud del brazo del cuadricóptero [m]	0.25
Jx=Jy	Momento de inercia en X-Y [kgm^2]	0.02
Jz	Momento de inercia en Z [kgm^2]	0.015
c	Relación entre la velocidad del rotor y su efecto en la rotación del cuadricóptero	0.02
g	Gravedad [m/s^2]	9.81

Tanto la masa, longitud del brazo del cuadricóptero como el valor de c son tomados de [5], sin embargo, el momento de inercia en X-Y-Z se pueden calcular. Para esto se asume que el cuadricóptero es un cuerpo rígido y uniforme, de tal forma que la masa se distribuye por igual para cada brazo del vehículo y se utiliza la siguiente fórmula: [7]

$$J_i = \left(\frac{1}{3} \times ml^2\right) \quad (2.1)$$

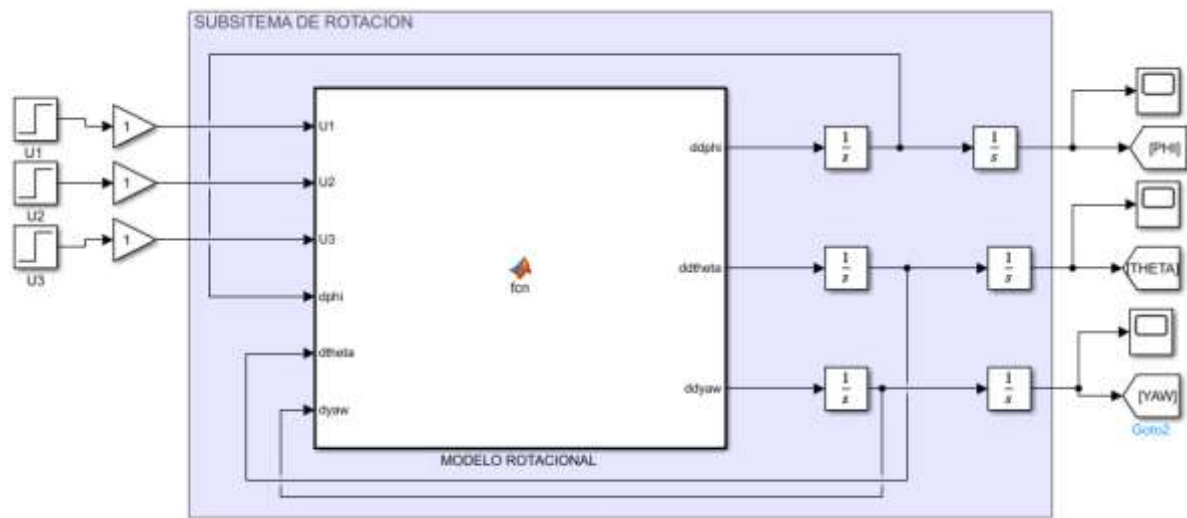


Figura 2.4 Implementación del Subsistema de Rotación

Determinados los parámetros se procede a la implementación del modelo como se observa en la Figura 2.4 y 2.5 que corresponden respectivamente al subsistema de rotación y al subsistema de altitud que como solo tiene la ecuación de aceleración respecto al eje Z se lo llamara subsistema de altitud. El primero tiene como entrada los pares U1, U2, U3 y de salida tiene los ángulos de pitch, roll y yaw. El segundo tiene como entradas ángulos pitch, roll y U4 (fuerza de empuje) y como salida tiene la posición en el eje Z.

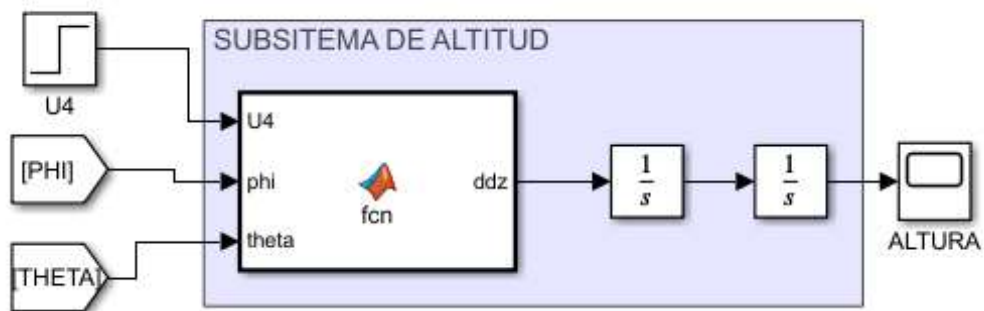


Figura 2.5 Implementación del Subsistema de Altitud

Para la simulación del modelo sin controladores tanto para los pares U1, U2, U3 como para el empuje U4 se va a utilizar una entrada paso.

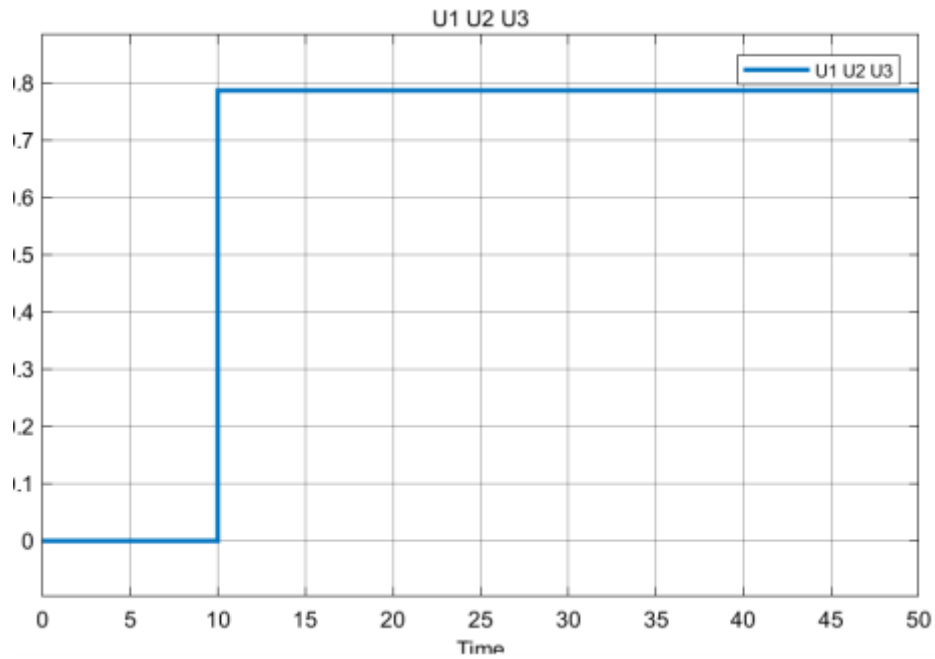


Figura 2.6 Entrada paso señal U1, U2, U3

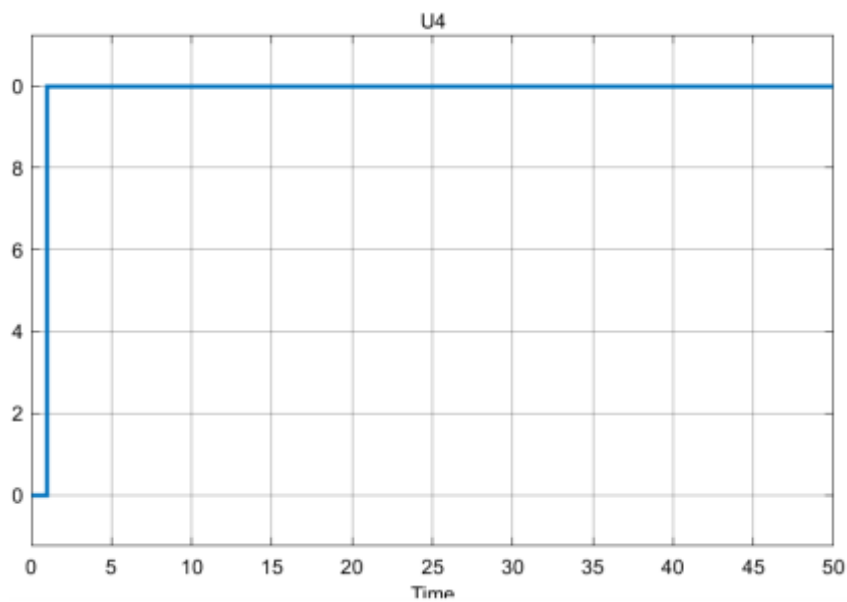


Figura 2.7 Entrada paso señal U4

En la Figura 2.8 se puede observar las salidas del subsistema de rotación, donde la señal de roll descende sin control, mientras tanto las señales de pitch y yaw aumente sin control.

Para la señal de salida del subsistema de altitud que representa la posición en el eje Z y se presenta en la Figura 2.9 se tiene un descenso sin control.

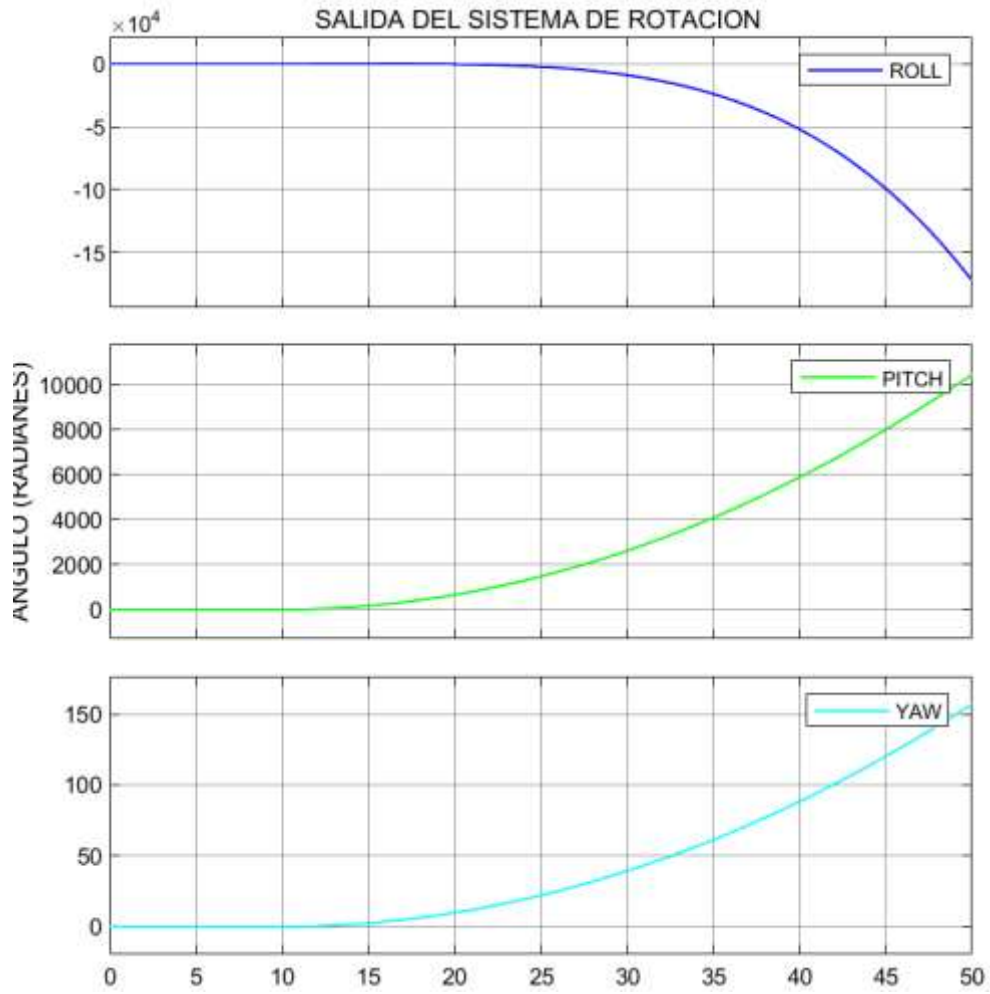


Figura 2.8 Señales de salida del subsistema de rotación

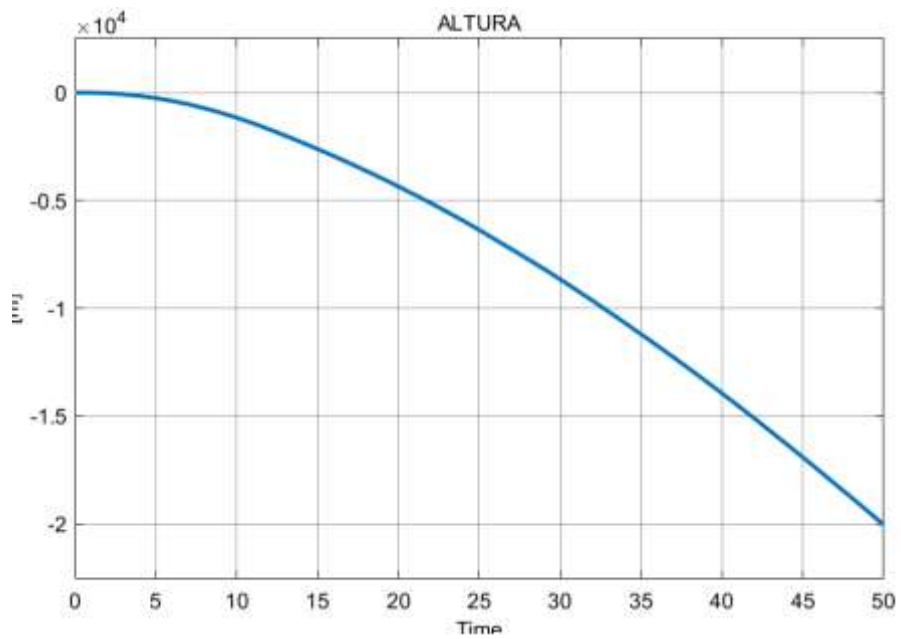


Figura 2.9 Señales de salida del subsistema de altitud

2.2 DISEÑO DEL CONTROLADOR PID

Para el diseño del controlador PID en el modelo propuesto se va a diseñar 4 controladores PID, que van a permitir controlar roll, pitch, yaw y la altura.

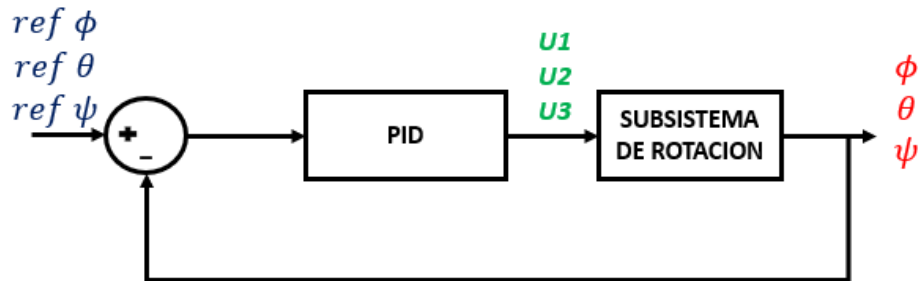


Figura 2.10 Control PID para el subsistema de rotación

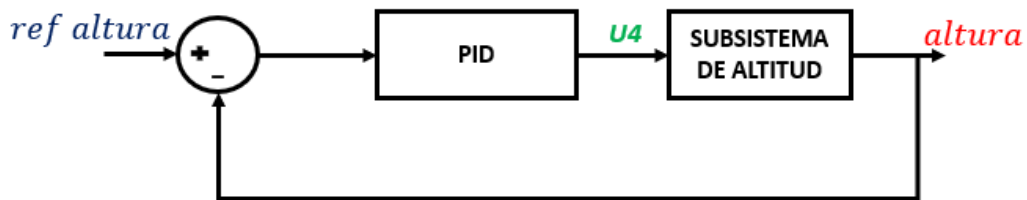


Figura 2.11 Control PID para el subsistema de altitud

Para el diseño del controlador PID para ROLL se va a tener como entrada del controlador la diferencia entre el valor deseado del ángulo de roll y el valor de roll obtenido del subsistema de rotación y como salida del controlador el par $U1$.

Para el diseño del controlador para pitch y yaw se tiene una relación similar a la mencionada en el anterior párrafo.

Para el diseño del controlador PID para el subsistema de altitud como entrada se tiene la diferencia entre la altura deseada y la altura del sistema y como salida el empuje $U4$.

Establecidas las entrada y salidas del controlador se procede con el diseño, para esto como al tener un modelo no lineal se dificulta la sintonización de los controladores, sin embargo, se lo puede realizar empíricamente mediante prueba y error o en este caso al utilizar Matlab/Simulink se cuenta con el PID Tuner que nos permite obtener fácilmente la ganancia proporcional, integral y derivativa.

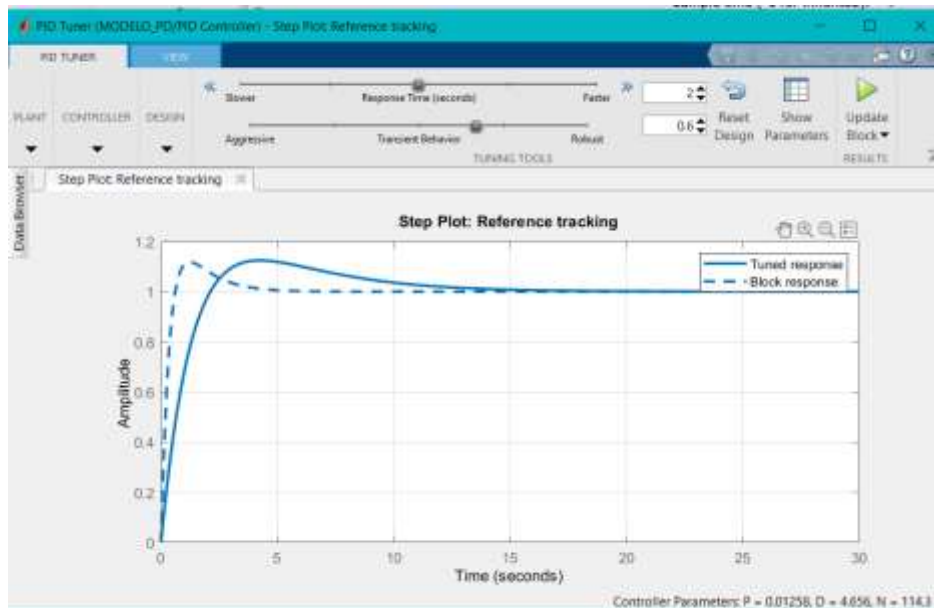


Figura 2.12 PID Tuner de MATLAB / Simulink

Se inicia con el diseño del controlador de ROLL, para lo cual tanto para las entradas de U2 y U3 se van a establecer en cero y posteriormente se inicia con el tuneo del controlador.

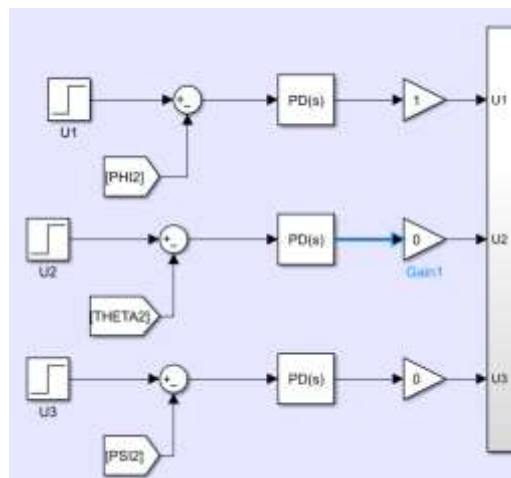


Figura 2.13 Entrada U2 y U3 en cero

En la Tabla 2.2 se expone los valores PID para el controlador de roll.

Tabla 2.2. Valores PID para roll

P	I	D
0.099	0.046	4.794

Para el diseño del controlador de PITCH, las entradas U1 y U3 se establecen en cero y en la Tabla 2.3 se expone los valores PID para el controlador.

Tabla 2.4. Valores PID para pitch

P	I	D
0.06	0.029	3.76

Mediante la obtención de estos dos primeros controladores se puede identificar que los valores de sintonización son muy parecidos, por lo cual para ambos controladores se podría tener los mismos valores.

Para el diseño del controlador de YAW, las entradas U1 y U2 se establecen en cero y en la Tabla 2.4 se expone los valores PID para el controlador.

Tabla 2.5. Valores PID para yaw

P	I	D
6.53	0.126	1.92

Para el diseño del controlador del subsistema de altura es necesario que ya estén implementados los controladores del subsistema de rotación porque de acuerdo con la ecuación 1.4-15, los valores de pitch y roll influyen en la obtención de la altura.

Mediante la Tabla 2.4 se expone los valores PID para el controlador de altura

Tabla 2.5. Valores PID para altura

P	I	D
4.16	0.147	0.814

2.3 DISEÑO DEL CONTROLADOR PD

El diseño de controladores PD es muy similar al diseño de los controladores PID, se tiene la misma estructura de bloques:

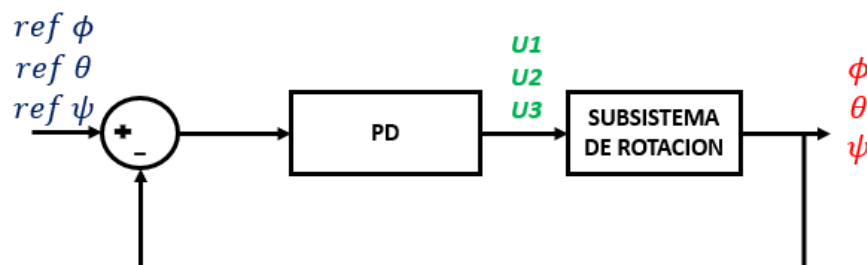


Figura 2.14 Control PD para el subsistema de rotación

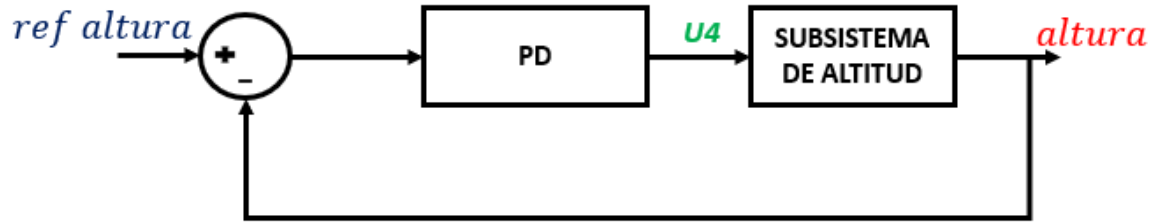


Figura 2.15 Control PD para el subsistema de rotación

Entonces se puede decir que se realiza el mismo procedimiento mencionado anteriormente para poder obtener los controladores, en este caso solo se va a obtener ganancia proporcional y derivativa.

En las siguientes tablas se expresa los valores para cada controlador:

Tabla 2.6. Valores PD para roll

P	D
0.137	1.478

Tabla 2.7. Valores PD para pitch

P	D
0.01	82.78

Tabla 2.8. Valores PD para yaw

P	D
0.11	62.798

Para el subsistema de rotación se puede observar que la ganancia proporcional y derivativa obtenida son completamente diferente entre ellas.

Tabla 2.9. Valores PD para altura

P	D
4.68	0.83

Para el subsistema de altitud se puede observar que los valores de la ganancia proporcional y derivativa son parecidos a los del controlador PID.

2.4 DISEÑO DEL CONTROLADOR BASADO EN LÓGICA DIFUSA

Para el diseño del controlador basado en lógica difusa se van a generar 2 controladores con la misma lógica que van a ser implementados para el control de los ángulos de roll y pitch, esto debido a que estos ángulos tienen mayor sustentación al momento de controlar.

El método por trabajar para obtener el controlador es el de Mamdani y para la defusificación se va a trabajar por el método de centroide.

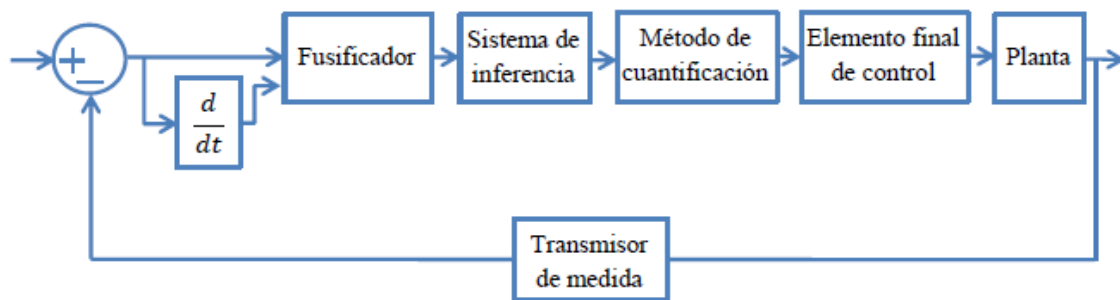


Figura 2.16 Arquitectura de control difuso [6]

De acuerdo con la arquitectura presentada en la Figura 2.16 el controlador difuso va a tener 2 entradas y 1 salida, entonces se inicia definiendo la entrada y salida del controlador

Entradas:

- $e(t)$: la diferencia entre el valor deseado y la salida del sistema (esto sería tanto para roll como para pitch)
- $\frac{d}{dt}e(t)$: la variación del error en el tiempo

Salida:

- U1 para el controlador difuso de roll y U2 para el controlador difuso de pitch

Posteriormente se determina los valores lingüísticos para las variables de entrada y salida, esto se presenta en las Tablas 2.10 – 2.11 – 2.12. Mediante los valores lingüísticos establecidos se tiene 5 conjuntos difusos que es lo más recomendable, debido a que si solo se utiliza 3 valores tal vez no se abarque todo el rango de datos del proceso y en cambio sí se establece mayor número de conjuntos también aumenta el número de reglas que se tengan que definir.

Tabla 2.10. Valores lingüísticos para el error

DEFINICION	SIMBOLOGIA
Error grande negativo	EGN
Error pequeño negativo	EPN
Error cero	EZ
Error pequeño positivo	EPP
Error grande positivo	EGP

Tabla 2.11. Valores lingüísticos para la derivada del error

DEFINICION	SIMBOLOGIA
Derivada Error grande negativo	DEGN
Derivada Error pequeño negativo	DEPN
Derivada Error cero	DEZ
Derivada Error pequeño positivo	DEPP
Derivada Error grande positivo	DEGP

Tabla 2.12. Valores lingüísticos para la salida del controlador

DEFINICION	SIMBOLOGIA
U negativo grande	TNG
U pequeño negativo	TN
U cero	TZ
U pequeño positivo	TP
U positivo grandes	TPG

Una vez determinado los valores lingüísticos se procede a crear las funciones de membresía para la entrada y salida del controlador. Hay que mencionar que el universal discurso de entrada viene dado de $[-3.16 \ 3.16]$ y los conjuntos a crear se van a encontrar dentro de este rango.

Tabla 2.13. Funciones de membresía para el error

SIMBOLOGIA	FUNCION DE MEMBRESIA
EGN	$[-3.16 \ -3.16 \ -1.56 \ -0.7825]$
EPN	$[-1.56 \ -0.7825 \ 0]$
EZ	$[-0.7825 \ 0 \ 0.7825]$
EPP	$[0 \ 0.7825 \ 1.56]$
EGP	$[0.7825 \ 1.56 \ 3.16 \ 3.16]$

Las funciones de membresía correspondiente a la derivada del error serán las mismas de las funciones de membresía del error, debido a que ambas tienen un comportamiento de forma similar, la única diferencia está la forma de obtener el dato, que en este caso es la

derivada del error, este último también se puede decir que actúa como tendencia del error dando así una perspectiva del funcionamiento de este. [8]

El discurso universal de salida viene dado de [-0.1 0.1] y dentro de este se van a crear las 5 funciones de membresía para la salida.

Tabla 2.14. Funciones de membresía para U

SIMBOLOGIA	FUNCION DE MEMBRESIA
TNG	[-0.1 -0.1 -0.08 -0.04]
TN	[-0.08 -0.04 0]
TZ	[-0.04 0 0.04]
TP	[0 0.04 0.08]
TPG	[0.04 0.08 0.1 0.1]

Establecida la base de datos se procede a crear una base de reglas, las cuales son elaboradas de acuerdo con la experiencia y conocimiento sobre el funcionamiento del cuadricóptero. El conjunto de reglas generalmente tiene una sintaxis 'si – entonces', sin embargo, en muchos casos también se hace presente el conectivo 'AND' y es lo que se presentara en el trabajo. El número de reglas a construir son de acuerdo con el número de entradas que tenga el controlador. [9]

Por ejemplo: si se tiene una entrada (error) y una salida (temperatura) se va a tener solo 5 reglas ya que cada valor lingüístico de entrada va a corresponder a un valor lingüístico de salida y esta se puede evidenciar en la Table 2.15.

Tabla 2.15. Base de reglas para 1 entrada 1 salida

error	ENG	ENP	EZ	EPP	EPG
temperatura	TNG	TNP	TZ	TPP	TPG

La base de reglas escritas en sintaxis 'si – entonces' queda:

- Si el error es negativo grande ENTONCES la temperatura es negativa grande
- Si el error es negativo pequeño ENTONCES la temperatura es negativa pequeña

Para el controlador a diseñar en este trabajo se tiene 2 entradas de 5 funciones de membresía cada una por lo tanto se va a tener 25 reglas, con la sintaxis 'si – y – entonces'.

La construcción de las reglas se realiza en base a la siguiente lógica, donde tiene mayor importancia el error.

Si se tiene un error grande negativo (ENG) y la derivada del error también es negativa grande (DENG), quiere decir que la señal de salida (Roll o Pitch) se encuentra alejada del punto de referencia, por lo tanto, es necesario una acción de control que aplique una fuerza grande negativa (TNG) para que la salida alcance el punto de referencia.

Si se tiene un error grande negativo (ENG) y la derivada del error es pequeña positiva (DEPP), quiere decir que la señal de salida (Roll o Pitch) se encuentra un poco más arriba que el punto de referencia, por lo tanto, es necesario una acción de control que aplique una fuerza pequeña negativa (TN) para que la salida alcance el punto de referencia.

Si se tiene un error grande positivo (EGP) y la derivada del error es pequeño positivo (DEPP), quiere decir que la señal de salida (Roll o Pitch) se encuentra mucho más abajo que el punto de referencia, por lo tanto, es necesario una acción de control que aplique una fuerza positiva grande (TPG) para que la salida alcance el punto de referencia.

Si se tiene un error cero (EZ) y la derivada del error es grande positivo (DEGP), quiere decir que la señal de salida (Roll o Pitch) se encuentra mucho más abajo que el punto de referencia, por lo tanto, es necesario una acción de control que aplique una fuerza positiva pequeña (TPG) para que la salida alcance el punto de referencia.

Mediante la Tabla 2.16 se presenta como quedan determinado las reglas.

Tabla 2.16. Base de reglas para controlador difuso

$de \setminus e$	ENG	ENP	EZ	EPP	EGP
DENG	TNG	TNG	TN	TZ	TP
DENP	TNG	TN	TZ	TZ	TP
DEZ	TNG	TZ	TZ	TZ	TPG
DEPP	TN	TZ	TZ	TP	TPG
DEGP	TNG	TZ	TP	TPG	TPG

La base de reglas para el controlador escritas en sintaxis 'si – y - entonces' son:

- Si el error es ENG y la derivada del error es DENG entonces U1 es TNG
- Si el error es ENG y la derivada del error es DENP entonces U1 es TNG
- Si el error es ENG y la derivada del error es DEZ entonces U1 es TNG
- Si el error es ENG y la derivada del error es DEPP entonces U1 es TN
- Si el error es ENG y la derivada del error es DEGP entonces U1 es TNG

- Si el error es ENP y la derivada del error es DENG entonces U1 es TNG
- Si el error es ENP y la derivada del error es DENP entonces U1 es TN
- Si el error es ENP y la derivada del error es DEZ entonces U1 es TZ
- Si el error es ENP y la derivada del error es DEPP entonces U1 es TZ
- Si el error es ENP y la derivada del error es DEGP entonces U1 es TZ
- Si el error es EZ y la derivada del error es DENG entonces U1 es TN
- Si el error es EZ y la derivada del error es DENP entonces U1 es TZ
- Si el error es EZ y la derivada del error es DEZ entonces U1 es TZ
- Si el error es EZ y la derivada del error es DEPP entonces U1 es TZ
- Si el error es EZ y la derivada del error es DEGP entonces U1 es TP
- Si el error es EPP y la derivada del error es DENG entonces U1 es TZ
- Si el error es EPP y la derivada del error es DENP entonces U1 es TZ
- Si el error es EPP y la derivada del error es DEZ entonces U1 es TZ
- Si el error es EPP y la derivada del error es DEPP entonces U1 es TP
- Si el error es EPP y la derivada del error es DEGP entonces U1 es TPG
- Si el error es EGP y la derivada del error es DENG entonces U1 es TP
- Si el error es EGP y la derivada del error es DENP entonces U1 es TP
- Si el error es EGP y la derivada del error es DEZ entonces U1 es TPG
- Si el error es EGP y la derivada del error es DEPP entonces U1 es TPG
- Si el error es EGP y la derivada del error es DEGP entonces U1 es TPG

Para la implementación del controlador difuso en Matlab/Simulink se lo realiza utilizando el Toolbox Fuzzy Logic, donde se va a ingresar todos los datos del controlador.

- Al ingresar al toolbox primero hay que definir el número de entradas, en este caso dos y se le coloca un nombre a cada variable lo mismo con la salida (Figura2.17).
- Ingresar las funciones de membresía para el error

- Ingresar las funciones de membresía para la derivada del error, que es igual a la del error.
- Ingresar las funciones de membresía para la salida U. (Figura 2.18)

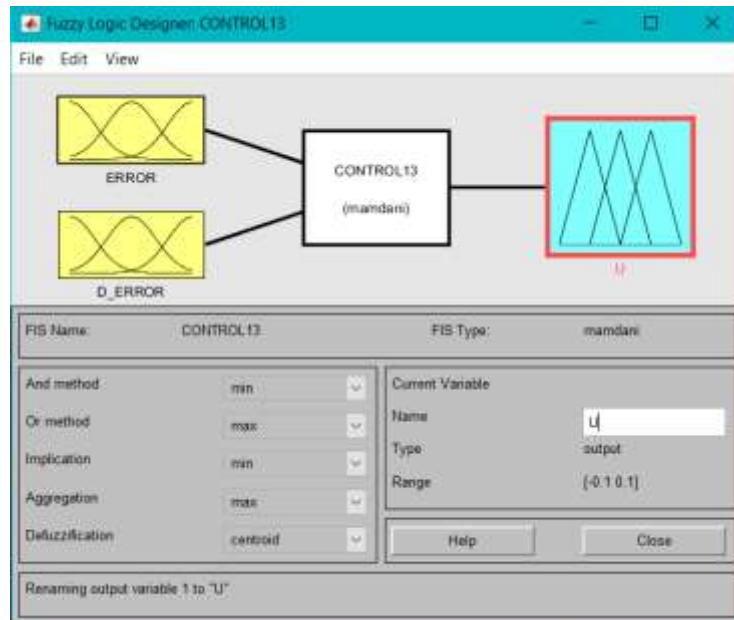


Figura 2.17 Fuzzy Logic Designer

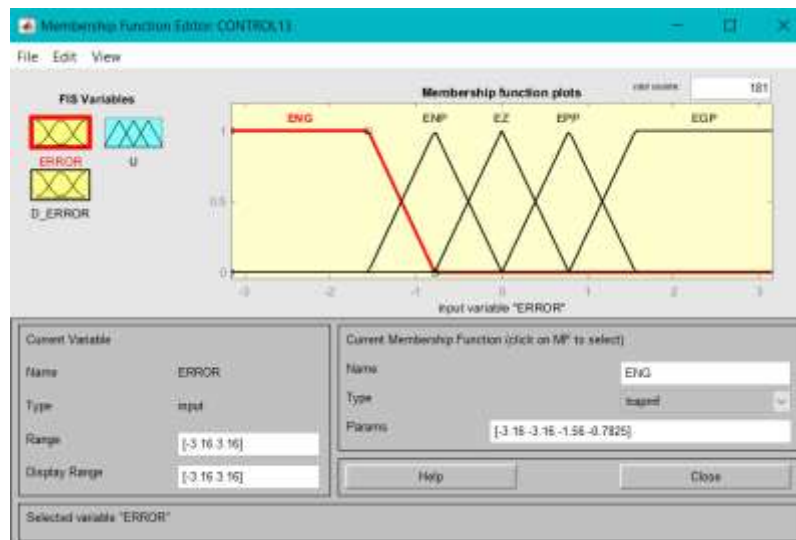


Figura 2.18 Ingreso de las variables de membresía para el error

- Ingresar la base de reglas del controlador (Figura 2.19)

Una vez realizado todo lo mencionado anteriormente se exporta el controlador ya sea a una carpeta o al workspace, al realizar esto se debe colocar un nombre al controlador y este nombre es el que se debe configurar en el bloque de Fuzzy Logic Controller.

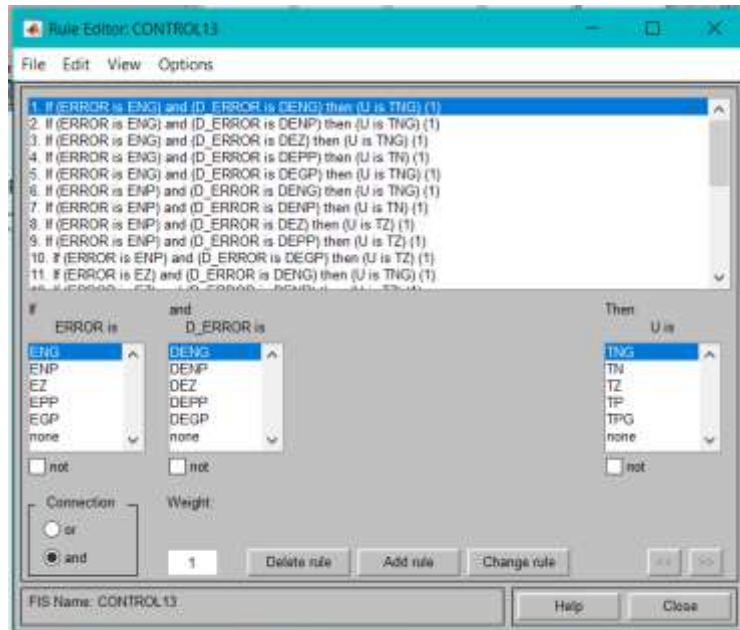


Figura 2.18 Ingreso de las variables de membresía para el error

3. RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

3.1 RESULTADOS

En esta etapa se presenta los resultados obtenidos al implementar los controladores diseñados en el capítulo anterior, para comprobar su funcionalidad se realizará pruebas a un valor constante y a valores variados de referencia de altura, valores constantes y variados de ángulos de roll, pitch y yaw. Se obtendrá los índices de desempeño ISE e ISCO para los controladores de altura, se observa las acciones de control para cada caso y también se procederá a calcular el valor de sobreimpulso.

Las gráficas presentadas van a ser una comparación de las señales obtenidas del subsistema de rotación y del subsistema de altitud ante la implementación de los controladores diseñados e identificar el más preciso.

3.1.1 REFERENCIA DE ÁNGULO Y ALTURA CONSTANTE

Para comprobar el funcionamiento de los controladores, inicialmente para las tres entradas del subsistema de rotación se tendrá un valor constante de $\frac{\pi}{4}$ [rad] y para la referencia de altura se tendrá un valor de 14 [m]

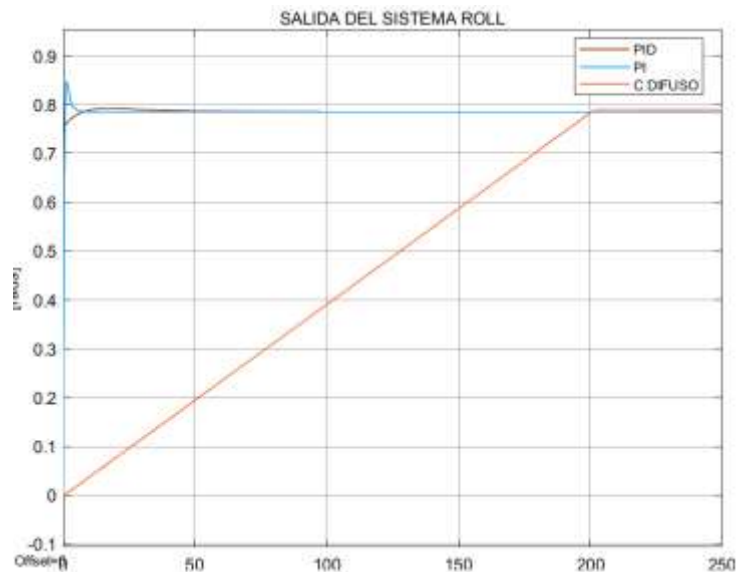


Figura 3.1 Señal de salida de ROLL

Mediante la siguiente Tabla 3.1 se realiza una comparación del tiempo de establecimiento y sobreimpulso que presenta el ángulo roll ante una entrada constante, obteniendo que se obtiene una mejor respuesta al implementar un controlador PID, ya que ante un controlador PD se presenta un elevado sobreimpulso y ante un controlador difuso se presenta mayor tiempo de establecimiento.

Tabla 3.1. Resultados de las pruebas realizadas a Roll

Controlador	Sobre impulso [%]	Tiempo de establecimiento [s]
PID	<0.1	55
PD	8.25	20
DIFUSO	0	200

Tabla 3.2. Resultados de las pruebas realizadas a Pitch

Controlador	Sobre impulso [%]	Tiempo de establecimiento [s]
PID	5.12	100
PD	<0.1%	100
DIFUSO	0	180

Ante los resultados presentados en la Tabla 3.2 para las pruebas realizadas para pitch se analiza que al implementar un controlador PD se va a obtener mejor tiempo de establecimiento y un valor casi nulo de sobreimpulso. En la Tabla 3.3 se analiza los

resultados para Yaw donde mediante los parámetros obtenidos el implementar un controlador PD es lo mejor.

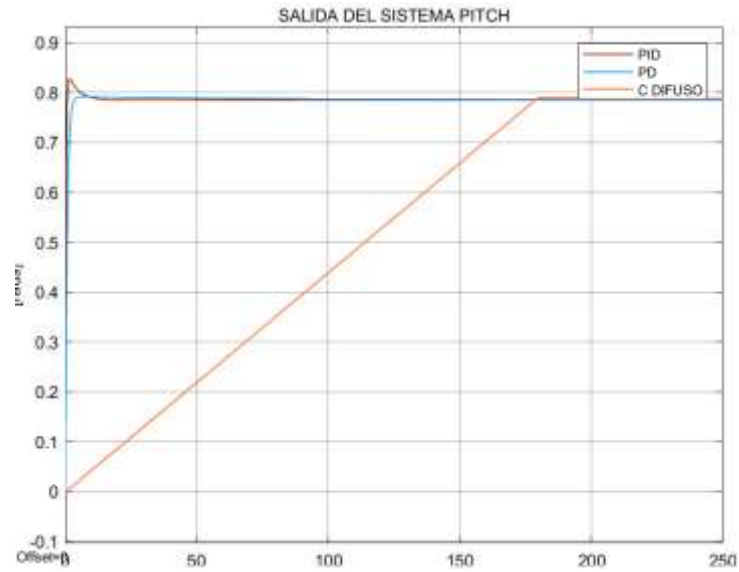


Figura 3.2 Señal de salida de PITCH

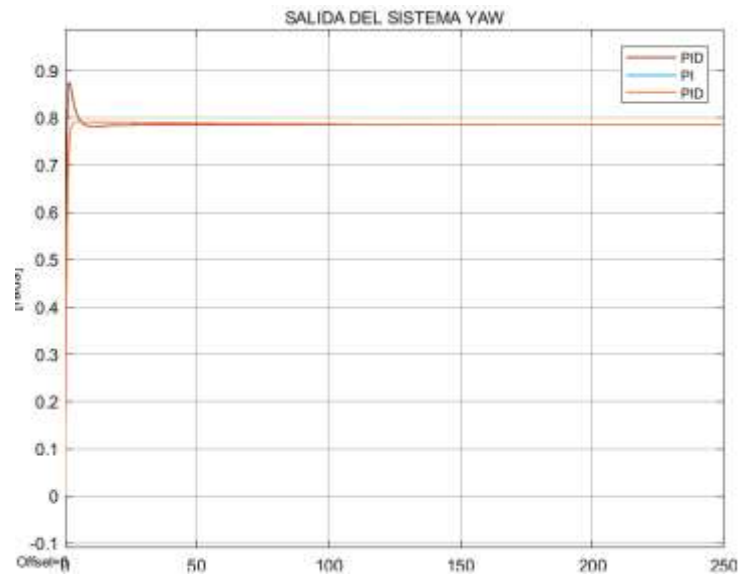


Figura 3.3 Señal de salida YAW

Tabla 3.3. Resultados de las pruebas realizadas a Yaw

Controlador	Sobre impulso [%]	Tiempo de establecimiento [s]
PID	12.82	100
PD	<0.1%	25

Para esta primera prueba donde se analizó las respuestas del subsistema de rotación (ϕ, θ, ψ) , se puede determinar que se tuvieron mejores resultados al implementar el controlador PD, ante el controlador difuso se presenta elevados tiempos de establecimiento y para el controlador PID existe presencia de sobreimpulso.

En este caso no se va a determinar los índices de desempeño ISE e ISCO debido a que el presente trabajo se enfoca en el estudio del movimiento del cuadricóptero en el eje Z por lo tanto se profundiza su estudio en este.

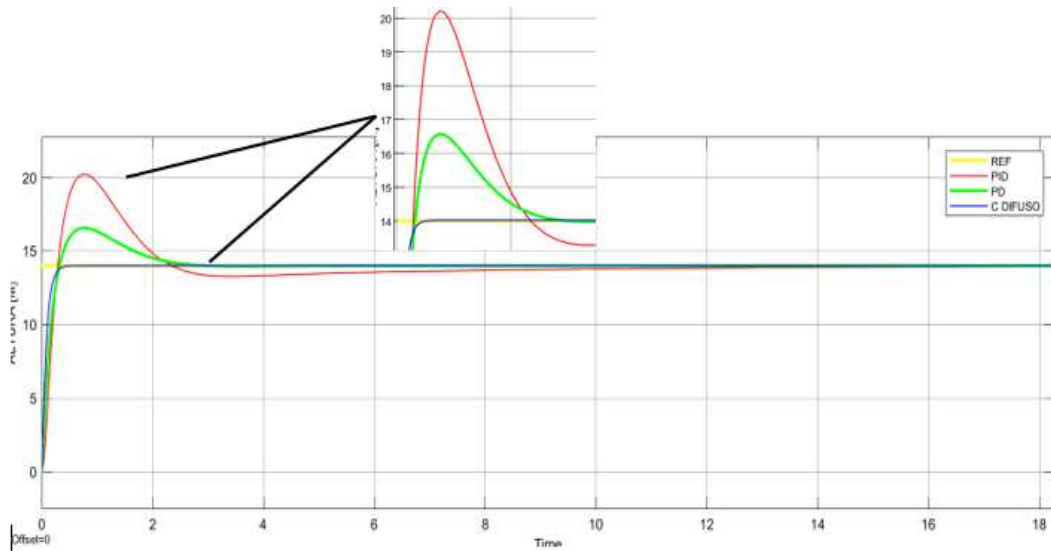


Figura 3.4 Señal de salida del subsistema de altitud

Para el subsistema de altitud también se realizó el diseño de controladores y mediante la Tabla 3.4 se puede determinar que al implementar los controladores y compararlos se tiene mejores resultados ante un controlador de lógica difusa y se tiene peores condiciones ante un controlador PID.

Mediante la obtención de los índices de desempeño para la altitud se observó que se obtiene un valor ISE menor para un controlador Difuso y un valor ISCO menor el PID seguido por el valor del controlador difuso, por lo tanto, para señales de entradas constantes un controlador basado en lógica difusa es más robusto y óptimo.

Tabla 3.4. Resultados de las pruebas realizadas a Altura

Controlador	Sobre impulso [%]	Tiempo de establecimiento [s]
PID	42.85	15
PD	17.14	3
DIFUSO	0	0.5

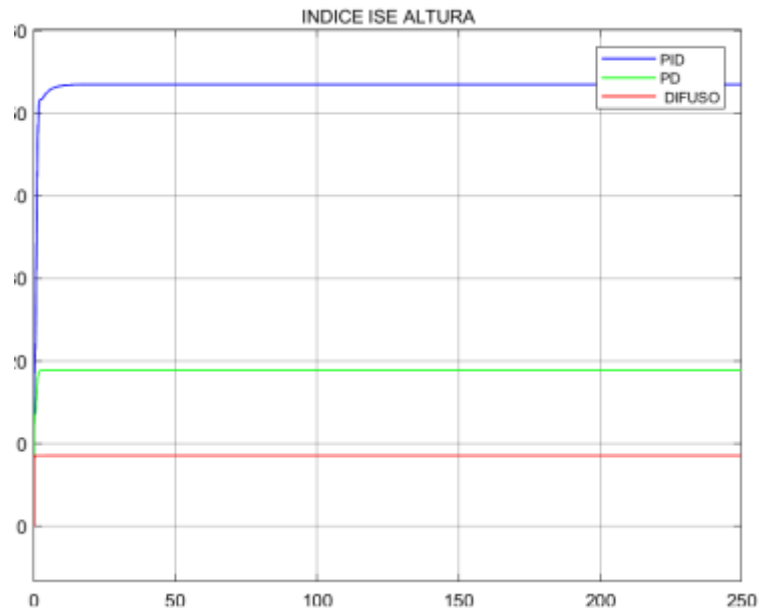


Figura 3.5 Índice ISE para controladores de altura

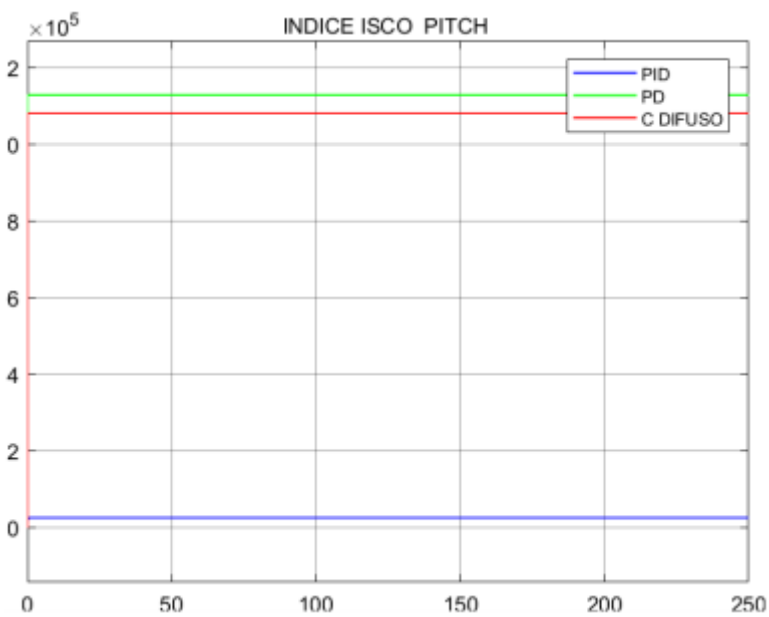


Figura 3.6 Índice ISCO para controladores de altura

Tabla 3.5. Resultados de los índices de desempeño para Altura

Controlador	ISE	ISCO ($\times 10^5$)
PID	53	100
PD	8	100
DIFUSO	18	180

3.1.2 REFERENCIA DE ÁNGULO CONSTANTE Y ALTURA VARIABLE

Esta prueba se va a realiza con un valor de referencia constante para las tres entradas del subsistema de rotación, siendo este valor π [rad] y para la referencia de altura se variará la altura entre 0 y 15 [m] en un tiempo de 0 a 300 [s].

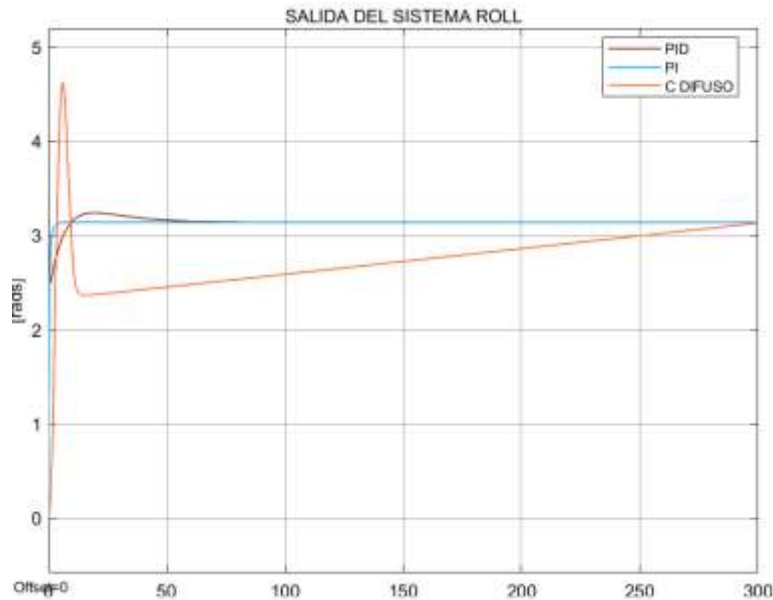


Figura 3.7 Señal de salida de ROLL

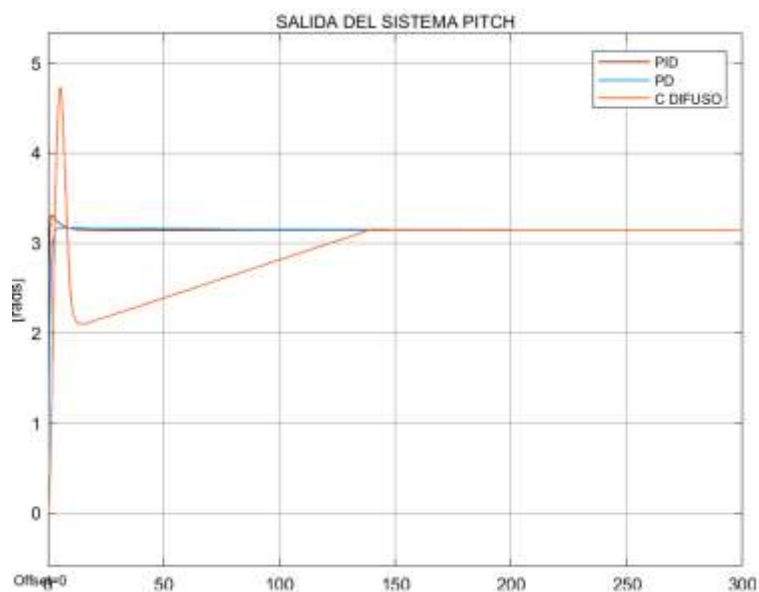


Figura 3.8 Señal de salida de PITCH

Al analizar las Figuras de las señales obtenidas se puede decir que ante una entrada de referencia de π tanto para roll como pitch, el controlador de lógica difusa actúa de forma brusca presentando un elevado sobreimpulso y mayor tiempo de establecimiento, esto debido a que el universal discurso de entrada está en un rango de $[- 3.16 \ 3.16]$, que son los valores máximos que puede ser la referencia de entrada por lo tanto siempre es

recomendable evitar trabajar con los extremos. Al referirnos a los controladores implementados PID y PD tanto para roll y pitch se puede decir que el más óptimo es el PD ya que presenta menor tiempo de establecimiento y casi nada de sobreimpulso.

Tabla 3.6. Resultados de las pruebas realizadas a Roll

Controlador	Sobre impulso [%]	Tiempo de establecimiento [s]
PID	3.49	70
PD	0	5
DIFUSO	46.49	300

Tabla 3.7. Resultados de las pruebas realizadas a Pitch

Controlador	Sobre impulso [%]	Tiempo de establecimiento [s]
PID	1.91	70
PD	<0.1%	5
DIFUSO	49.58	140

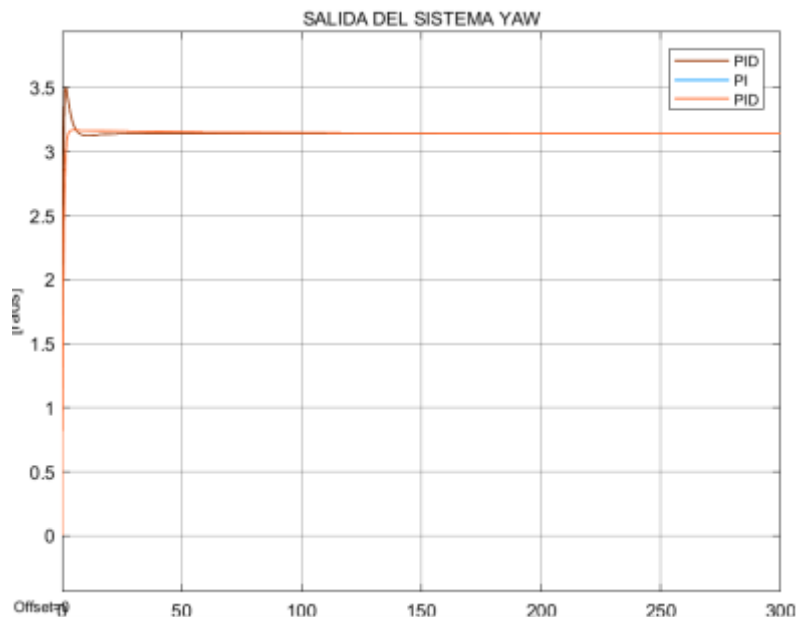


Figura 3.9 Señal de salida de YAW

Para la variable yaw (ψ) se diseñó solo controladores PID y PD, debido a que este no influye en la ecuación de altitud. Pero igual se realiza un breve análisis que ante un cambio de referencia la acción de control funciona correctamente, teniendo una mejor respuesta ante un control PD ya que con el PID se tiene mayor tiempo de establecimiento y sobreimpulso.

Tabla 3.8. Resultados de las pruebas realizadas a Yaw

Controlador	Sobre impulso [%]	Tiempo de establecimiento [s]
PID	11.46	60
PD	<0.1%	5

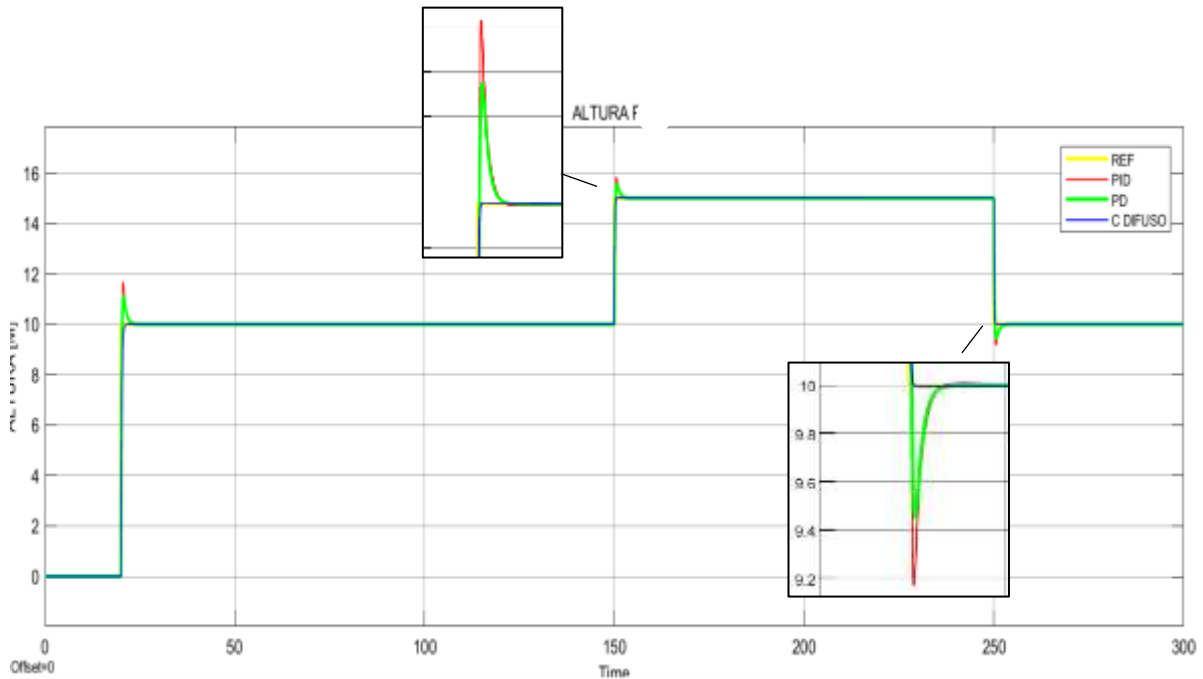


Figura 3.10 Señal de salida del subsistema de altitud

Ante un cambio de referencia de la variable altitud se puede determinar que al aplicar un control PID se tiene mayor tiempo de establecimiento y mayor sobreimpulso, que es diferente ante un control PD donde disminuye el tiempo de establecimiento y sobreimpulso y cambia totalmente ante la implementación de controladores difusos en los ángulos pitch y roll que influyen en la altitud, para este último caso se determinó que el tiempo de establecimiento es pequeño y es imperceptible el sobreimpulso.

Entonces se puede decir que los controladores difusos aplicados en roll y pitch si ayudan en la intención de mantener un despegue rápido y preciso y un aterrizaje suave.

Tabla 3.9. Resultados de las pruebas realizadas para ascenso

Controlador	Sobre impulso [%]	Tiempo de establecimiento [s]
PID	8.2	15
PD	0	10
DIFUSO	0	2

Tabla 3.10. Resultados de las pruebas realizadas para descenso

Controlador	Sobre impulso [%]	Tiempo de establecimiento [s]
PID	8.2	15
PD	2.6	10
DIFUSO	0	2

Tabla 3.11. Resultados de los índices de desempeño para Altura (10m)

Controlador	ISE	ISCO ($\times 10^5$)
PID	12.5	0.2
PD	7.6	5.8
DIFUSO	11.3	5.5

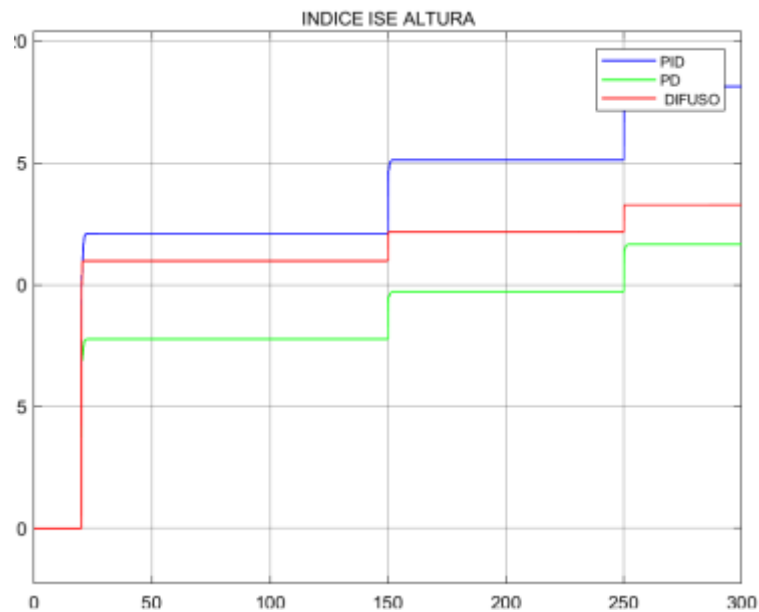


Figura 3.11 Índice ISE para controladores de altura

A simple vista se determinó que el mejor controlador es el difuso, sin embargo, de acuerdo al índice ISE, el controlador más óptimo es el PD y el peor el PID y de acuerdo al índice ISCO se va a tener una acción menos brusca a la salida del control, al implementar una acción PID y una acción más brusca al implementar un PD, pero tanto el valor ISE e ISCO del controlador difuso se encuentran en una posición intermedia sin llegar a ser mejor o peor en los índices de desempeño dando a entender que si cumple con las condiciones para ser un control robusto y óptimo.

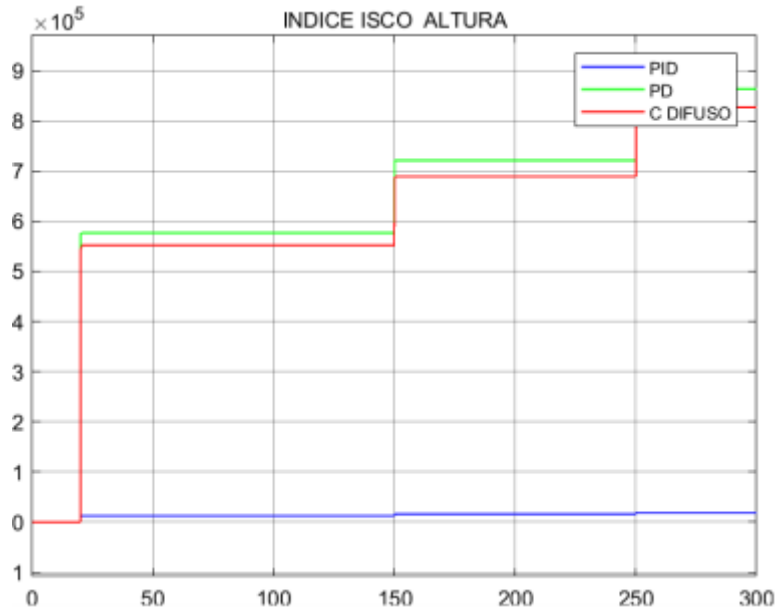


Figura 3.12 Índice ISCO para controladores de altura

3.1.3 REFERENCIA DE ÁNGULO VARIABLE Y ALTURA CONSTANTE

Los valores de referencia para las tres entradas del subsistema de rotación se tendrá un valor variable mayor de π [rad] y menor de $\frac{\pi}{3}$ [rad] y para la referencia de altura se tendrá un valor constante de 10[m].



Figura 3.13 Señal de salida de ROLL

Con la variación del ángulo de referencia de roll (ϕ), pitch (θ) y yaw (ψ), las respuestas de roll ante la implementación de los controladores se observa en la Figura 3.13, donde para la acción de control PID al iniciar con un ángulo de $\frac{\pi}{3}$ [rad] se tiene un pequeño sobreimpulso, al cambiar la referencia a π [rad] se tiene un mayor porcentaje de sobreimpulso y al cambiar nuevamente la referencia a $\frac{\pi}{3}$ [rad] se tiene un sobreimpulso brusco y con respecto al tiempo de establecimiento se promedia que es 30s; para la acción de control PD, cuando se producen los cambios antes mencionados, presentan sobreimpulsos imperceptibles y un tiempo promedio de establecimiento de 10s; para el controlador difuso ante la referencia inicial de $\frac{\pi}{3}$ [rad] no presenta sobreimpulso pero si un elevado tiempo de establecimiento, al cambia la referencia a π [rad] se observa un gran sobreimpulso y un elevado mayor tiempo de establecimiento ya que, aunque haya pasado 300s no se alcanza la referencia y cuando vuelve a caer la referencia se tiene un sobreimpulso negativo elevado. Mediante toda esta explicación se puede determinar que los controladores con mejor respuesta para la variación de referencia angular para roll son PID y PD, siendo este último el mejor



Figura 3.14 Señal de salida de PITCH

El funcionamiento de los controladores implementados para la variable pitch es similar a la de la variable de roll, por lo que se llega a la misma conclusión de que los controladores más robustos serian PID y PD.

Para el subsistema de altitud y ante los controladores implementados se tiene una visión diferente a la del subsistema de rotación, debido a que en este caso a simple vista se tiene

mejor respuesta con la implementación de controladores difusos dando un menor tiempo de establecimiento y sobreimpulso.

Tabla 3.12. Resultados de las pruebas realizadas para altitud

Controlador	Sobre impulso [%]	Tiempo de establecimiento [s]
PID	8	22
PD	2.5	6
DIFUSO	0	< 1

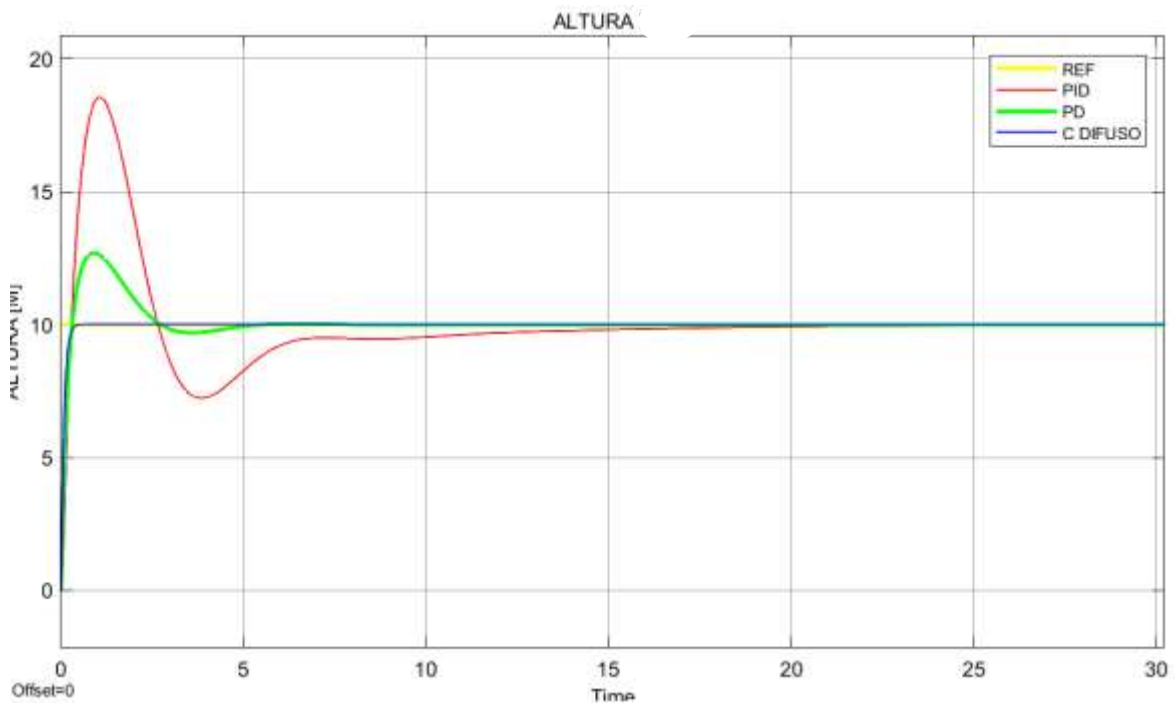


Figura 3.15 Señal de salida del subsistema de altitud

Tabla 3.13 Resultados de los índices de desempeño para Altura (10m)

Controlador	ISE	ISCO ($\times 10^5$)
PID	110	0.2
PD	17	5.5
DIFUSO	5	5.75

De acuerdo con el índice de desempeño ISE se tiene un valor elevado para PID y un mínimo para el controlador difuso, siendo este el más robusto, sin embargo, de acuerdo con el índice ISCO el controlador difuso presenta un mayor valor dando a entender que necesita una mayor acción de control para alcanzar la referencia.

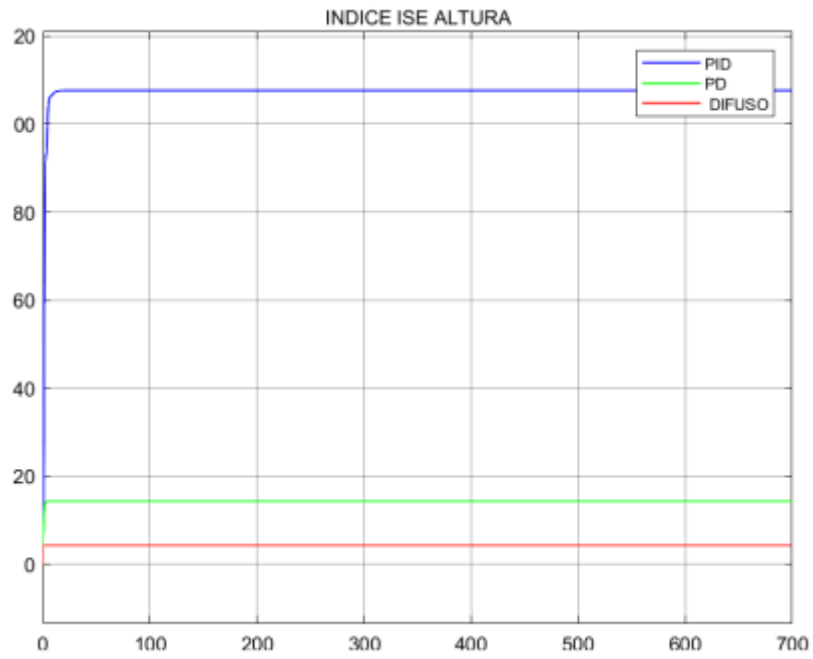


Figura 3.11 Índice ISE para controladores de altura

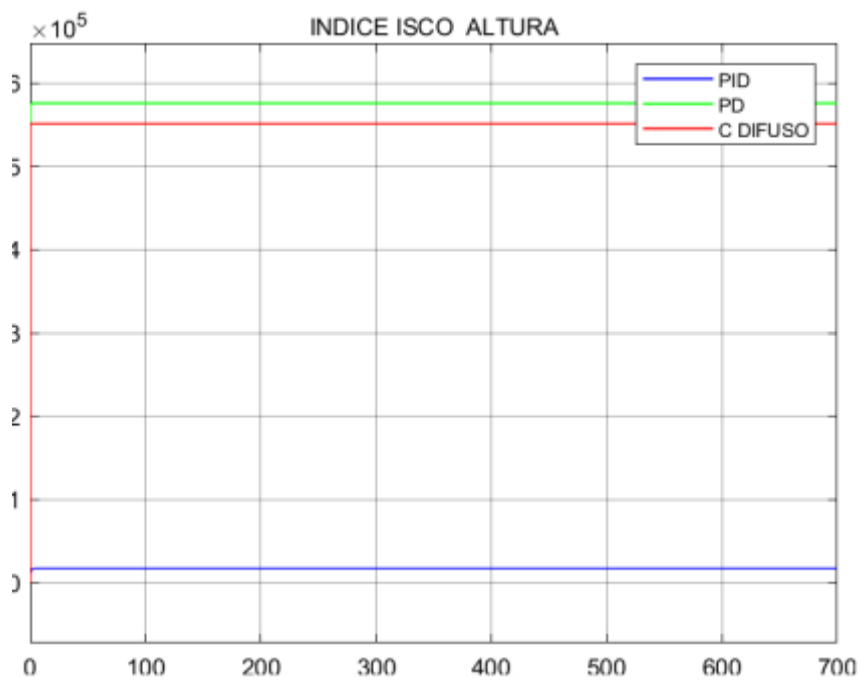


Figura 3.12 Índice ISCO para controladores de altura

3.2 CONCLUSIONES

El despegue y aterrizaje de un cuadricóptero hace referencia al movimiento que se tendrá en el eje Z, sin embargo, de acuerdo al modelo matemático escogido también influye los valores de roll y pitch por tanto se diseñó e implementó controladores PID, PD y DIFUSO para roll, pitch y para la altura y a simple vista se determina que mientras los controladores PID y PD dan una mejor respuesta de roll y pitch, no pasa lo mismo con la altura debido a que presenta mayor sobreimpulso, pero al implementar el controlador DIFUSO, las salidas de roll y pitch presentan mayor sobre impulso y tiempo de establecimiento y la altura no presenta sobreimpulso y tiene menor tiempo de establecimiento, dando a entender que al referirse completamente a la altura el controlador basado en lógica difusa da una mejor respuesta. Sin embargo, si nos referimos a los índices de desempeño ISO e ISCO, para el primero se determina que el controlador difuso sería el más óptimo, pero para el segundo este controlador presenta un valor elevado dando a entender que necesita una mayor acción de control para alcanzar la referencia ocasionando un mayor consumo de energía, no siendo lo mejor. Por lo que se puede concluir que la implementación de controladores PD sería lo más adecuado.

Las variables que intervienen en el aterrizaje y despegue del cuadricóptero son los valores de pitch y roll y de acuerdo con la ecuación para determinar el movimiento del UAV en el eje z, se determina que mientras más pequeño sea el valor de pitch y roll, el movimiento dependerá del empuje (U_4) que se esté aplicando y de la gravedad.

El cuadricóptero se considera el vehículo aéreo más estable y fácil de controlar por lo que puede aplicarse en diferentes aplicaciones siendo su principal uso para la seguridad y en el presente trabajado al implementar los controladores PID, PD y DIFUSO para el aterrizaje y despegue, se logró obtener en unos de los casos un tiempo de establecimiento de dos segundos siendo esta una variable importante al momento de hablar de seguridad ya que ante una emergencia se desea alcanzar una altura de forma rápida para comenzar a seguir una trayectoria o se desea un aterrizaje rápido para poder enfocar un peligro en un área determinada.

En la lógica difusa, la base de reglas que se construyen se considera un factor importante para tener un mejor rendimiento del controlador, ya que de nada sirve que se tenga correctamente establecidos las funciones de membresía, debido a que mediante el conjunto de reglas se obtiene un valor difuso a la salida para ser desdifusado.

Al realizar las pruebas para diferentes referencias se puede concluir que mientras los controladores PID y PD tienen la capacidad de actuar ante cualquier referencia, no pasa

los mismo con los controladores difusos debido a que estos cuentan con un universal discurso de entrada y la referencia deseada debe encontrarse dentro de este rango. Esto fue muy notorio al momento de la simulación porque si se escogía una referencia cercana a los extremos del universal discurso se presentaba un tiempo de simulación elevado y peor si se escogía una referencia fuera del rango, la simulación daba error.

3.3 RECOMENDACIONES

El entender el funcionamiento del cuadricóptero nos permitirá también entender el modelo matemático del mismo y el funcionamiento de sus variables de tal forma que se nos facilite el diseño de sus controladores.

El modelo matemático presentando está compuesto por dos subsistemas y en total son seis ecuaciones, sin embargo, en este caso se hacen el uso de cuatro ecuaciones, tres del subsistema de rotación y una ecuación del subsistema de translación que corresponde a la ecuación de aceleración del eje Z, para futuros trabajos donde sea necesario controlar en su totalidad la trayectoria del cuadricóptero se puede hacer uso de las seis ecuaciones.

El diseño del controlador basado en lógica difusa requiere un completo entendimiento del funcionamiento del cuadricóptero para que en trabajos futuros se pueda tener mayor número de funciones de membresía de entrada y salida y más importante se puede construir la base de reglas de tal forma que se consiga mayor precisión.

4. REFERENCIAS BIBLIOGRÁFICAS

- [1] J. C. Bajo, «El uso profesional de los drones y la prevención de riesgos laborales,» 27 Abril 2021. [En línea]. Available: https://prevencion.mc-mutual.com/articulos/-/asset_publisher/gPV7bp1C7xJS/content/el-uso-profesional-de-los-drones-y-la-prevencion-de-riesgos-laborales. [Último acceso: 18 julio 2022].
- [2] J. Martos, «Control Automático con simulink de un mini dro. Aplicación al seguimiento de trayectorias mediante realimentacion visual,» Universidad Politecnica de Valencia, 2021. [En línea]. Available: <https://riunet.upv.es/handle/10251/174861>. [Último acceso: 21 Junio 2022].
- [3] P. Ruiperez, «Diseño y fabricación de un dron,» Universida Politecnica de Valencia, 2015. [En línea]. Available: <https://riunet.upv.es/bitstream/handle/10251/73170/RUIP%C3%89REZ%20-%20Dise%C3%B1o%20y%20fabricaci%C3%B3n%20de%20un%20dron%20media%20impresi%C3%B3n%203D.pdf?sequence=5>. [Último acceso: 4 Julio 2022].
- [4] B. Castelo y B. Mosquera, «Diseño e implementación de un sistema para la detección y seguimiento de una persona a trav{es de un cuadricóptero de tamaño reducido empleando visión artificial,» Escuela Politecnica Nacional, 2022. [En línea]. Available: <https://bibdigital.epn.edu.ec/bitstream/15000/22126/1/CD%2011618.pdf>. [Último acceso: 15 Enero 2022].
- [5] A. Courage, «Dynamic model and control of quadrotor in the presence of uncertainties,» University of South Carolina, 2017. [En línea]. Available: <https://scholarcommons.sc.edu/cgi/viewcontent.cgi?article=5091&context=etd>. [Último acceso: 5 Mayo 2022].
- [6] F. Gutierrez, «Modelado y simulación de control de una aeronave tipo cuadricóptero,» Universidad Tecnológica de Pereira, 2018. [En línea]. Available: <https://repositorio.utp.edu.co/server/api/core/bitstreams/c443b4a3-64c4-48a4-8b59-dbb7f2d701a1/content>. [Último acceso: 21 Junio 2022].
- [7] M. Usman, «Quadcopter Modelling and Control with MATLAB/Simulink Implementation,» LAB University of Applied Sciences, 2022. [En línea]. Available: https://www.theseus.fi/bitstream/handle/10024/333992/Usman_Muhammad.pdf?sequence=2. [Último acceso: 18 Julio 2022].
- [8] J. Zambrano, «Control Difuso Aplicado a un Vehículo Aéreo no Tripulado,» Universidad EAFIT, 2012. [En línea]. Available: https://repository.eafit.edu.co/bitstream/handle/10784/1300/JuanN._ZambranoCaviedes_2012.pdf?sequence=1&isAllowed=y. [Último acceso: 26 Julio 2022].
- [9] M. Awais, «Adaptive Fuzzy Control of Quadrotor,» RIT Scholta Works, 2017. [En línea]. Available:

<https://scholarworks.rit.edu/cgi/viewcontent.cgi?article=10776&context=theses>.
[Último acceso: 31 Julio 2022].

- [10] L. Andrango, «Modelado y control clásico de un cuadricóptero para el despegue y aterrizaje suave en interior,» ESPE, 2021. [En línea]. Available: <http://repositorio.espe.edu.ec/xmlui/bitstream/handle/21000/25629/T-ESPE-044684.pdf?sequence=1&isAllowed=y>. [Último acceso: 26 Julio 2022].
- [11] F. Guerrero y M. Oswaldo, «Modelación, simulación y control de sistemas aéreos no tripulado utilizando inteligencia artificial,» Escuela Politecnica Nacional, 2013. [En línea]. Available: <https://bibdigital.epn.edu.ec/bitstream/15000/6755/1/CD-5130.pdf>. [Último acceso: 1 Agosto 2022].
- [12] A. Vázquez, «Control Syatem for Autonomous Landind on Moing Platforms,» Universidad Carlos III de Madrid, 2017. [En línea]. Available: <https://core.ac.uk/download/pdf/288500278.pdf>. [Último acceso: 26 Julio 2022].
- [13] M. Gil, «Desarrollo de un sistema de aterrizaje autónomo sobre una plataforma de carga para un UAV tipo Parrot AR.Drone 2,» Universidad Politecnica de Madrid, 2016. [En línea]. Available: https://oa.upm.es/43909/1/TFG_MARCOS_ENRIQUE_GIL_FERNANDEZ.pdf. [Último acceso: 3 Agosto 2022].
- [14] V. Salcedo, «Aterrizaje automatico de un vehiculos aéreo no tripulado basado en seguimiento de puntos de interés para superficies móviles,» ESPE, 2018. [En línea]. Available: <http://repositorio.espe.edu.ec/xmlui/bitstream/handle/21000/14006/T-ESPE-057593.pdf?sequence=1&isAllowed=y>. [Último acceso: 3 Agosto 2022].

5. ANEXOS

ANEXO I. ECUACIONES DEL MODELO MATEMÁTICO EN MATLAB FUNCTION

ANEXO II. ESQUEMA DE CONTROL PID

ANEXO III. ESQUEMA DE CONTROL PD

ANEXO IV. ESQUEMA DE CONTROL BASADO EN LOGICA DIFUSA

ANEXO I. ECUACIONES DEL MODELO MATEMÁTICO EN MATLAB FUNCTION

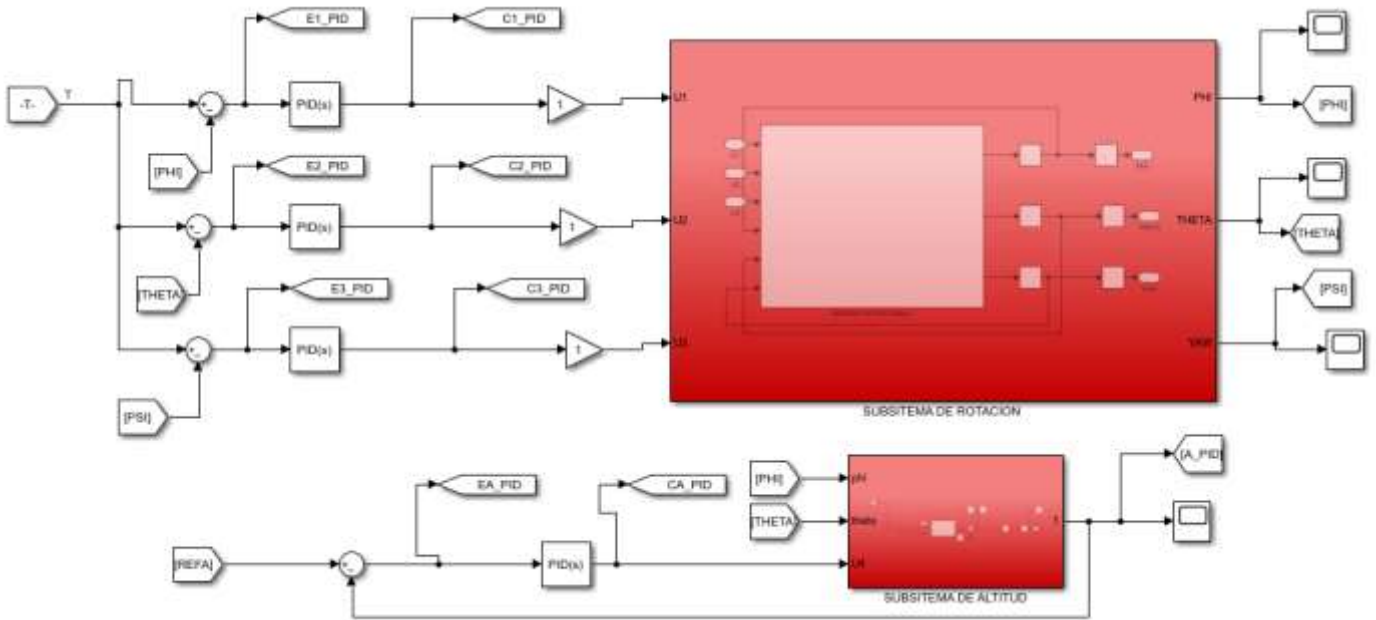
SUBSISTEMA DE ROTACIÓN

```
function [ddphi, ddtheta, ddyaw] =  
fcn(U1, U2, U3, dphi, dtheta, dyaw)  
l=0.25;  
c=0.02;  
jx=0.015;  
jy=0.015;  
jz=0.02;  
ddphi=( (U1*l/jx) + (dtheta*dyaw*jy/jx) -  
(dtheta*dyaw*jz/jx) );  
ddtheta=( (U2*l/jy) + (dphi*dyaw*jz/jy) - (dphi*dyaw*jz/jx) );  
ddyaw=( (U3*l*c/jz) + (dphi*dtheta*jx/jz) -  
(dphi*dtheta*jy/jz) );
```

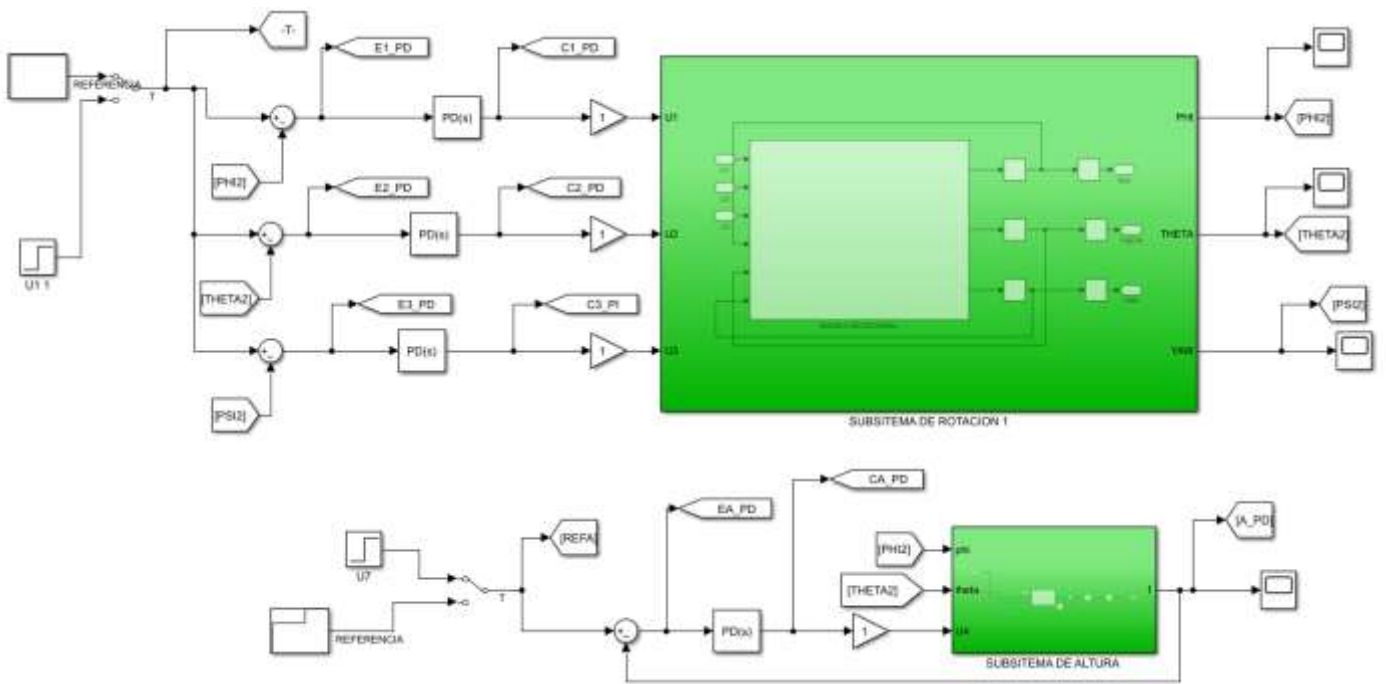
SUBSISTEMA DE ALTITUD

```
function ddz = fcn(U4, phi, theta)  
m=0.6;  
g=9.8;  
ddz = ( (-U4/m) * (cos(theta) * cos(phi)) ) - g ;
```

ANEXO II. ESQUEMA DE CONTROL PID



ANEXO III. ESQUEMA DE CONTROL PD



ANEXO IV. ESQUEMA DE CONTROL DIFUSO

