



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

**“DISEÑO Y ANÁLISIS DEL DESEMPEÑO DE
CONTROLADORES PID Y PID-FUZZY PARA CONTROL
DE ALTITUD EN UN CUADRICÓPTERO”**

TRABAJO DE TITULACIÓN

Previo a la obtención del Título de:

**MAGISTER EN AUTOMATIZACIÓN Y CONTROL
INDUSTRIAL**

ING. CESAR RONALD FLOR ROLDAN

ING. LUIS ANTONIO REYES RIZO

GUAYAQUIL – ECUADOR

2020

AGRADECIMIENTO

Agradezco a Dios, a mis Padres: Soledad y Patricio por creer y confiar en mí y en mis decisiones, y por todo el apoyo y optimismo que me muestran para no dejar a un lado mis metas.

A nuestro profesor y amigo el PhD. Douglas Plaza G., por el apoyo y direccionamiento durante todo el ciclo de aprendizaje, desarrollo de la maestría, y por toda la paciencia para ayudarnos a alcanzar la meta.

Luis Reyes R.

AGRADECIMIENTO

A Dios, por haberme dado la oportunidad de culminar con éxito esta meta.

A mi madre, Rina Rosario Roldán Rodríguez, por haber confiado completamente en mis capacidades, desde el día en que nací.

A mi familia y amigos de la MACI VI cohorte, por su cariño y respaldo en todo momento.

Al Gobierno Nacional de la República del Ecuador, por haberme otorgado la beca para financiar mis estudios por medio del Instituto de Fomento al Talento Humano (IFTH) y la Secretaría Nacional de Educación Superior, Ciencia y Tecnología (SENESCYT).

A mis maestros, porque siempre dieron lo mejor de ellos para enseñarnos y guiarnos durante esta maestría.

César Flor R.

DEDICATORIA

Dedico este trabajo a Dios que me da la oportunidad de demostrar que puedo más cada día, a la familia Reyes Rizo por siempre estar a mi lado acompañándome en todos los pasos que doy, a mis compañeros de la MACI VI por ser verdaderos luchadores que a pesar de las dificultades y distancias siempre llegaban a sus clases. Y de manera especial a todos los héroes del 2020, que con sus profesiones lucharon contra el COVID-19 salvando muchas vidas.

Luis Reyes R.

Dedico esta tesis a mi hija, Fiorella Victoria Flor Carrión, porque es mi mayor motivación para superarme día a día, y a todas las víctimas del Covid-19 para demostrarles que de la mano de Dios, podemos alcanzar nuestros sueños a pesar de todo lo malo que pueda ocurrir.

César Flor R.

TRIBUNAL DE SUSTENTACIÓN

SUB-DECANO FIEC

**PHD. DOUGLAS PLAZA G.
DIRECTOR**

VOCAL PRINCIPAL

DECLARACIÓN EXPRESA

“La responsabilidad y la autoría del contenido de este Trabajo de Titulación, nos corresponde exclusivamente; y damos nuestro consentimiento para que la ESPOI realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”.

Ing. César Ronald Flor Roldan

Ing. Luis Antonio Reyes Rizo

RESUMEN

El presente trabajo describe el diseño y análisis de desempeño de dos tipos de controladores de altitud para un cuadricóptero. En primer lugar se desarrolló un controlador PID clásico y luego se desarrolló un controlador PID-Fuzzy. Ambos controladores fueron implementados sobre el modelo matemático simplificado de un cuadricóptero, el cual se fundamenta en las ecuaciones de Newton-Euler.

Empezamos definiendo los índices de desempeño como son: sobrenivel porcentual (OS), tiempo de estabilización (Tss), y error de estado estable (Ess) alrededor de un punto de operación, el cual permitió evaluar el comportamiento de los controladores PID clásico y Fuzzy PID.

En el controlador PID, se aplicó las técnicas de control regulatorio para desarrollar un controlador clásico tipo Proporcional Integral Derivativo (PID). De esta manera se obtuvo un controlador Cz para el control de altura, el cual cumplió con los parámetros establecidos de; OS, Tss y, Ess. Luego se aplicó la lógica de Mamdani para el desarrollo de un controlador Fuzzy-PID, donde se definió las reglas lingüísticas basadas en la experiencia lógica de funcionamiento de un cuadricóptero. Con la ayuda de la herramienta Fuzzy Logic Designer de Matlab® se definió dos variables de entradas; error de la posición y error de la velocidad y una variable de control de salida. Con la herramienta Rules del Fuzzy Logic Designer, simulamos los valores de entrada para verificar la salida deseada. Con los dos controladores desarrollados y con la ayuda de Simulink®, se realizó la simulación del modelo de la planta junto con el controlador PID clásico y, con el controlador Fuzzy-PID por separado, para su análisis.

Finalmente se realizó la comparación del rendimiento de ambos controladores. De los resultados obtenidos, tenemos que el controlador Fuzzy-PID tiene mejor respuesta OS y Tss, que el controlador PID clásico. También como parte de la comparativa, se realizó la simulación de los controladores ante la respuesta de una perturbación. En conclusión, el controlador Fuzzy-PID tiene mejor rendimiento para el control de altura ante los cambios de referencia y una mejor respuesta antes las perturbaciones, en comparación a un controlador PID clásico.

Palabras claves: Cuadricóptero, Control PID, Control Fuzzy-PID.

ÍNDICE GENERAL

AGRADECIMIENTO.....	II
AGRADECIMIENTO.....	III
DEDICATORIA.....	IV
TRIBUNAL DE SUSTENTACIÓN.....	V
DECLARACIÓN EXPRESA.....	VI
RESUMEN.....	VII
ÍNDICE GENERAL.....	VIII
ÍNDICE DE FIGURAS.....	XII
ÍNDICE DE TABLAS.....	XVI
CAPÍTULO 1 PLANTEAMIENTO DE LA PROBLEMÁTICA.....	1
1.1 <i>Identificación del problema.....</i>	<i>1</i>
1.2 <i>Justificación.....</i>	<i>1</i>
1.3 <i>Solución Propuesta.....</i>	<i>3</i>
1.4 <i>Objetivos de la Tesis.....</i>	<i>3</i>
1.4.1 <i>Objetivos Generales.....</i>	<i>3</i>
1.4.2 <i>Objetivos Específicos.....</i>	<i>3</i>
1.5 <i>Metodología.....</i>	<i>3</i>
1.6 <i>Alcance.....</i>	<i>4</i>
CAPÍTULO 2 ESTADO DEL ARTE.....	5
2.1 <i>Fundamentos de un cuadricóptero.....</i>	<i>5</i>
2.2 <i>Modelo matemático de un cuadricóptero.....</i>	<i>8</i>
2.3 <i>Controlador PID clásico aplicado al control de un cuadricóptero.....</i>	<i>14</i>

2.4 Controlador PID-Fuzzy aplicado al control de un cuadricóptero	17
CAPÍTULO 3 DISEÑO DE CONTROLADORES PID Y PID-FUZZY	25
3.1. Diseño y sintonización basada en modelo del controlador PID	25
3.2. Diseño de reglas de lógica difusa para el controlador FUZZY- PID.....	39
3.3. Ajuste y sintonización del controlador FUZZY - PID	41
CAPÍTULO 4	48
ANÁLISIS COMPARATIVO DEL DESEMPEÑO DE LOS CONTROLADORES.....	48
4.1 Descripción de los escenarios de pruebas.....	48
4.2 Resultados de la simulación.....	49
4.3 Análisis de resultados.....	58
4.4 Respuesta ante perturbaciones.....	63
CONCLUSIONES.....	68
RECOMENDACIONES Y TRABAJOS FUTUROS	69
BIBLIOGRAFÍA	70
ANEXOS	73
PROGRAMA DE MATLAB	73

ABREVIATURAS Y SIMBOLOGÍA

J	Matriz de inercia diagonal del cuadricóptero
ω	Velocidad angular del cuerpo
MB	Momento actuante en el centro del cuerpo.
Jr	Inercia de los rotores
ρ	Densidad del aire
A	Área de la hélice
Cd,Ct	Coefficientes aerodinámicos
Cx	Controlador de desplazamiento sobre eje X
Cy	Controlador de desplazamiento sobre eje Y
Cz	Controlador de altitud sobre eje Z
r	Radio de las hélices
Ω	Velocidad angular del rotor
Kf	Constantes de fuerza aerodinámica
Km	Constante de momento aerodinámica
FPB	Filtro Pasa bajo
PID	Proporcional integral derivativo
Kp	constante proporcional de controlador PID
Ki	constante integral de controlador PID
Kd	constante derivativa de controlador PID
Ess	Error en estado estable

$e(t)$	Error
I/O	Entradas / Salidas
OS%	Sobrenivel porcentual
SISO	Sistema con una entrada y una salida
Tss	Tiempo de estabilización
X	Parámetro sobre en el eje X
Y	Parámetro sobre el eje Y
Set-point	Valor de referencia para el controlador
Phi	Angulo de rotación para el eje X
Theta	Angulo de rotación para el eje Y
Psi	Angulo de rotación para el eje Z
RMSE	Error cuadrático medio

ÍNDICE DE FIGURAS

Figura 2.1. Cuadricóptero Parrot Mambo®	5
Figura 2.2. Partes de un Cuadricóptero	6
Figura 2.3. Movimiento traslacional	7
Figura 2.4. Movimiento Rotacional: Yaw, Pitch, Roll	7
Figura 2.5. Fuerzas y Momentos en Cuadricóptero: Roll, Pitch, Yaw.	8
Figura 2.6. Ejes de referencia de cuadricóptero.....	10
Figura 2.7 Herramienta Linear Analysis - Simulink®	15
Figura 2.8 Herramienta Sisotool - Matlab®	16
Figura 2.9 Estructura de control en cascada... ..	17
Figura 2.10 Representación de reglas difusas.	19
Figura 2.11 Conjunto recortado (a), y escalado (b)	22
Figura 2.12 Defuzificación con el método del centroide	23
Figura 2.13 Fuzzy Logic Designer - Matlab®	24
Figura 2.14 Fuzzy Logic Designer Evaluación de Reglas - Matlab®.....	24
Figura 3.1 Simulación de Cuadricóptero 3D - Matlab®	25
Figura 3.2 Entradas y Salidas representación del Sistema Cuadricóptero ..	26
Figura 3.3 Modelo no lineal Cuadricóptero - Matlab® - Simulink.....	26
Figura 3.4 Modelo no lineal Dinámica traslacional- Matlab® - Simulink	27

Figura 3.5. Modelo lineal del cuadricóptero - Matlab® - Simulink.....	28
Figura 3.6. Matriz estados vs estados - Matlab® - Simulink.....	29
Figura 3.7. Matriz estados vs entradas - Matlab® - Simulink	29
Figura 3.8. Matriz salidas vs estados - Matlab® - Simulink	29
Figura 3.9. Matriz salidas vs entradas - Matlab® - Simulink.....	30
Figura 3.10. Lugar Geométrico de las raíces Controlador Cz.	31
Figura 3.11 Respuesta de Controlador Cz.	32
Figura 3.12 Lugar geométrico de las raíces C_{ϕ} y C_{θ}	33
Figura 3.13 Respuesta de controlador C_{ϕ} y C_{θ}	33
Figura 3.14 Lugar geométrico de raíces de controlador traslación: X - Y. ...	35
Figura 3.15 Respuesta de controlador de Traslación: X y Y.	36
Figura 3.16 Arreglo de Controladores PD en Cascada en Simulink®.	38
Figura 3.17 Simulación Controlador PID de Cuadricóptero en Simulink.....	40
Figura 3.18 Herramienta Fuzzy Logic Designer - Matlab®.....	43
Figura 3.19 Reglas lógicas Controlador Fuzzy de altitud.	44
Figura 3.20 Entrada error de altitud, función membresía "trimf".....	44
Figura 3.21 Entrada velocidad del error, función membresía "trimf".	45
Figura 3.22 Salida de controlador, función membresía "zmf".....	45
Figura 3.23 Simulación controlador FUZZY - PID Cuadricóptero.....	49

Figura 4.1. Modelo no lineal de cuadricóptero con controlador PID	51
Figura 4.2. Respuesta de controlador de altitud tipo PID	52
Figura 4.3. Señal de error y salida del controlador de altitud PID	52
Figura 4.4. Altitud y desplazamientos con controlador PID	53
Figura 4.5. Movimiento angular con controladores PD.....	54
Figura 4.6. Movimiento rotacional y traslacional del cuadricóptero	55
Figura 4.7. Modelo no lineal de cuadricóptero con controlador Fuzzy-PID ..	56
Figura 4.8. Respuesta de controlador de altitud tipo Fuzzy-PID	57
Figura 4.9. Señal de error y salida del controlador Fuzzy-PID	57
Figura 4.10. Altitud y desplazamientos con controlador Fuzzy-PID	58
Figura 4.11. Movimiento angular con controladores PD.....	59
Figura 4.12. Movimiento rotacional y traslacional del cuadricóptero	60
Figura 4.13. Respuesta de controlador PID vs controlador Fuzzy-PID	61
Figura 4.14. Salida de controlador PID vs controlador Fuzzy-PID	62
Figura 4.15. Índices de rendimiento del controlador PID.....	63
Figura 4.16. Índices de rendimiento del controlador Fuzzy-PID.....	63
Figura 4.17. Perturbación por efecto de la fuerza del viento	64
Figura 4.18. Respuesta controlador de altitud PID ante cambio de viento.....	64

Figura 4.19. Respuesta controlador Fuzzy-PID ante cambio de viento.....	66
Figura 4.20. Perturbación por efecto de impulso externo.....	66
Figura 4.21. Respuesta del controlador de altitud PID ante impulso.....	67
Figura 4.22. Respuesta del controlador de altitud Fuzzy-PID ante impulso.....	68
Figura 4.23 Respuesta de controladores de altitud a perturbación impulso.....	68
Figura 4.24. Señales de controladores frente a la perturbación impulso.....	69

ÍNDICE DE TABLAS

Tabla 1: Movimientos y ángulos relacionados en un cuadricóptero.	10
Tabla 2: Estructura de inferencia Mandani	20
Tabla 3: Estructura de inferencia Takagi-Sugeno	21
Tabla 4: Parámetros de diseño para controlador Cz de altitud.	32
Tabla 5: Parámetros para controlador C_{ϕ} - Roll y C_{θ} - Pitch	34
Tabla 6: Parámetros para controlador de posición en X y Y.	36
Tabla 7: Variables lingüísticas para reglas de lógica difusa.	41
Tabla 8: Reglas de lógica difusa controlador de altitud.	42
Tabla 9: Respuesta de reglas lógica difusa con posición del error en -5 y variación de error de la velocidad.....	46
Tabla 10: Respuesta de reglas lógica difusa con posición del error en 0 y variación de error de la velocidad.....	47
Tabla 11: Respuesta de reglas lógica difusa con posición del error en 5 y variación de error de la velocidad.....	48
Tabla 12: Índices de Desempeño de Controladores PID y Fuzzy-PID	65

INTRODUCCIÓN

En las últimas décadas, los cuadricópteros se han desarrollado en diferentes áreas, especialmente para exploración y aplicaciones militares, por ser un vehículo aéreo no tripulado y que tienen características como habilidad para suspenderse, robustez, simplicidad mecánica y excepcional maniobrabilidad por lo que pueden ser utilizados para diferentes aplicaciones como la realización de la inspección visual remota.

Cuando se realiza un trabajo de inspección visual a diferentes alturas, esta se considera una actividad riesgosa y de difícil acceso para una persona, por lo que la utilización de un cuadricóptero es una opción viable y más segura ya que puede ser operado remotamente.

Sin embargo, es importante controlar la altura de vuelo para evitar colisiones del equipo con el medio circundante y que su sistema de control pueda responder ante las perturbaciones causadas por las condiciones climáticas las cuales frecuentemente son la influencia del viento y la fuerza gravitacional que pueden llegar a desestabilizarlo, este es un ejemplo donde se evidencia que es crítico mantener el cuadricóptero una altura deseada.

Por lo anterior expuesto, es necesario un controlador de altitud para un cuadricóptero que pueda garantizar alcanzar y mantener la altitud deseada puesto a que cada uno de las cuatro hélices produce una fuerza contra la fuerza gravitacional que permiten elevarlo hasta una determinada altura.

Desde los primeros desarrollos, controladores tipo PID convencionales han sido ampliamente usados para regular la altitud en los cuadricópteros. Con respecto a este tema, se pueden encontrar varios resultados en trabajo de investigación anteriores. Los autores Z. Mustapa, S. Saat, S. H. Husin and N. Abas (2014) presentaron el diseño del controlador de altitud para un cuadricóptero, el cual está basado en un controlador PID convencional [1].

En el año 2018, Kamarudin, Muhammad Nizam realizó un controlador PID integrado para controlar la altitud del cuadricóptero y dentro de ello demostró la comparación de controladores PID de lazo simple y PID con doble lazo [5]. Últimamente en el año 2019 los autores Hamdy, Osama & El-Nady, Abdelrady desarrollaron el modelamiento, simulación y control de un cuadricóptero usando un controlador PID [6].

Como complemento a los trabajos descritos anteriormente, queremos desarrollar un controlador Fuzzy-PID y realizar la comparación de su rendimiento con respecto al rendimiento de un controlador PID convencional.

Hemos escogido desarrollar el controlador de altitud Fuzzy-PID debido a las siguientes razones:

1. Los controladores difusos son más robustos que los controladores PID porque pueden cubrir una gama mucho más amplia de condiciones de funcionamiento que los PID y pueden funcionar con ruido y perturbaciones de diferentes naturalezas.
2. Los controladores difusos son personalizables, ya que es más fácil de entender y modificar sus reglas, que no solo utilizan la estrategia de un operador humano, sino que también se expresan en términos lingüísticos naturales.

La realización de este trabajo presenta algunos desafíos porque el cuadricóptero es un sistema no lineal y además tiene seis grados de libertad pero solo cuatro actuadores, lo cual convierte a este en un sistema subactuado.

Para evaluar el rendimiento del sistema se debe determinar qué tan rápido el cuadricóptero alcanza la altitud deseada y que tan estable mantiene su altura en estado estacionario.

CAPÍTULO 1

PLANTEAMIENTO DE LA PROBLEMÁTICA

1.1 Identificación del problema

Durante el vuelo de un cuadricóptero, usualmente se presenta el problema de mantener su posición cuando este equipo se eleva a una determinada altura ya que por su propia inercia y la dinámica producida por el accionar de sus hélices, su posición y altitud tienden a variar [1], [2].

Por lo anteriormente descrito, hemos realizado la implementación de un controlador Fuzzy-PID para enfrentar este problema y estudiar el comportamiento de la estabilidad de altitud de un cuadricóptero utilizando técnicas de control de lógica difusa [3], [4].

1.2 Justificación

En la actualidad, para realizar un proceso de inspección a diferentes alturas, se tiene que recurrir a la habilidad humana. El uso de accesorios y equipos permiten realizar el trabajo en altura pero sigue siendo considerado de alto riesgo.

Existen regulaciones y certificaciones que avalan al personal calificado para la realización de estos trabajos, sin embargo, el riesgo sigue estando presente en cada una de las inspecciones.

En consecuencia, considerar reemplazar el método de inspección haciendo uso de las facilidades que la tecnología nos presenta es una manera no solo de disminuir el riesgo para la persona. El uso de la tecnología también permite eliminarlo, además de poder disminuir los costos y los tiempos de ejecución de esta actividad.

El uso de cuadricópteros durante los últimos años ha permitido innovar en diferentes áreas de la exploración y otras diversas aplicaciones. Para el desarrollo de la presente tesis, la aplicación que se manejó es el posicionamiento

del cuadricóptero en un punto de operación definido para la realización de una inspección remota.

Pero, el uso de cuadricóptero tiene sus desventajas ya que dependiendo del tipo, cuentan con formas y tamaños que pueden llegar a ser perturbados con gran facilidad. Por consiguiente, el controlar sus parámetros como la altura es de suma importancia para poder evitar que el equipo entre en colisión con el medio circundante.

Ante esta condición es necesario mejorar el sistema de control de un cuadricóptero, de tal manera que su respuesta sea mejorada ante las perturbaciones externas.

Estas condiciones pueden ser consideradas climáticas como la influencia del viento, la fuerza gravitacional y posibles colisiones que puedan llegar a desestabilizarlo y perder la estabilidad de la imagen.

El control de altura junto con sus otros parámetros desde los primeros desarrollos de cuadricópteros se han realizado por medio de controladores PID convencionales.

En el presente trabajo, hemos desarrollado un controlador Fuzzy-PID y hemos realizado la comparación de su rendimiento contra un controlador PID convencional.

Al ser los Fuzzy-PID controladores más robustos pueden cubrir una gama más amplia de condiciones de funcionamiento. A su vez se pueden personalizar y entender por medio de las reglas lógicas las condiciones de control al utilizar términos lingüísticos naturales.

Por último esta implementación del controlador Fuzzy-PID nos ha permitido evaluar y validar la rapidez con la que responde ante el punto de operación y las perturbaciones externas.

1.3 Solución Propuesta

Para enfrentar el problema descrito en la sección anterior, hemos implementado el control de altitud de un cuadricóptero utilizando un controlador Fuzzy-PID. El uso de reglas lingüísticas puede determinar de manera lógica una mejor respuesta para el sistema de control de altitud, en comparación de un controlador de altitud tipo PID clásico.

1.4 Objetivos de la Tesis

1.4.1 Objetivos Generales

- Analizar el sistema de control de altitud en un cuadricóptero aplicando técnicas de control inteligente de lógica difusa para estabilizar su posicionamiento en un punto de operación definido

1.4.2 Objetivos Específicos

- Analizar el principio de funcionamiento de un cuadricóptero y sus elementos constitutivos.
- Simular el modelo del cuadricóptero utilizando una librería de Matlab Simulink®.
- Diseñar y sintonizar los controladores tipo PID y tipo Fuzzy-PID para regular su altitud.
- Realizar el análisis comparativo del desempeño de los controladores propuestos.

1.5 Metodología

Para la realización de esta propuesta nos basamos en el modelo simplificado de un cuadricóptero, el cual se fundamenta en las ecuaciones de Newton-Euler. Se ha aplicado técnicas de control regulatorio para desarrollar un controlador tipo Proporcional Integral Derivativo (PID) que fue aplicado en el sistema de control de altitud.

Determinamos sus parámetros de rendimiento y posteriormente utilizamos la metodología aplicada para el desarrollo del controlador Fuzzy-PID. Esta

metodología consiste en seleccionar la estructura jerárquica, escoger y medir las entradas, escalar las entradas, establecimiento de reglas de fusificación, defusificación y el escalamiento de salidas de control.

La implementación del modelo descrito y el diseño de los controladores propuestos fueron desarrollados bajo la plataforma computacional Matlab®-Simulink con el objetivo de validar la respuesta obtenida del modelo y realizar el análisis comparativo del rendimiento de ambos controladores.

Una vez ejecutada la simulación obtuvimos la curva Altitud vs Tiempo y procedimos a comparar el rendimiento de los controladores diseñados, para lo cual vamos a utilizar los siguientes índices de desempeño: Tiempo de Levantamiento (s), Sobrenivel Porcentual (%) , Tiempo de Estabilización (s), Variable de Control (%) y métricas relacionadas al comportamiento del error.

1.6 Alcance

El presente trabajo abarcó el planteamiento y desarrollo de los controladores, PID clásico y Fuzzy-PID por medio de herramientas teóricas y computacionales en Matlab-Simulink®.

Además se analizaron y compararon los resultados obtenidos de los dos controladores, con el fin de evaluar qué controlador permitió mejorar la estabilidad y control de altura de un cuadricóptero.

Igualmente se analizó y comparó el comportamiento de ambos controladores ante perturbaciones externas.

CAPÍTULO 2

ESTADO DEL ARTE

2.1 Fundamentos de un cuadricóptero

Los cuadricópteros también conocidos como drones o UAV son vehículos aéreos no tripulados y controlados remotamente, los cuales tienen un sin número de aplicaciones. Estos funcionan por contener una carcasa en la que se aloja todo el sistema de control, sensores, cámara. Junto a la carcasa se encuentran adaptados los rotores o propulsores más las hélices, que son los que en función de la velocidad de giro, permiten propulsarse hacia arriba y generar los diferentes movimientos.

Para el presente proyecto, nos basamos en el modelo simplificado de un cuadricóptero, el cual se fundamenta en las ecuaciones de Newton-Euler. Lo hemos implementado utilizando la herramienta computacional Matlab®-Simulink donde se podrá realizar la simulación para verificar el funcionamiento de la respuesta de los controladores PID clásico y FUZZY-PID.



Figura 2.1. Cuadricóptero Parrot Mambo®. [18]

Partes de un cuadricóptero

Entre las partes del cuadricóptero que se tiene para permitir la propulsión se tiene:



Figura 2.2. Partes de un Cuadricóptero. [18]

Chasis, fuselaje: es la estructura central en la que van integrados las partes eléctricas y batería, el cual debe poseer características aerodinámicas, bajo peso y rigidez.

Batería: dispositivo que acumula carga eléctrica la cual permite, por medio de una conversión electroquímica, suministrar energía a los circuitos eléctricos.

Placa Controladora de Vuelo: es el equivalente al mainboard en un computador, y es la que permite recoger datos de los diferentes sensores y actuadores, Ultrasonido, Cámara, Sensor de Presión, IMU (Inertial measurement unit) el cual mide e informa de la velocidad, orientación y fuerza gravitacional una combinación de acelerómetro y giroscopio.

Los sensores de ultrasonido y presión permiten determinar la altitud, y la IMU junto con la cámara permiten determinar el movimiento rotacional y traslacional.

Parte propulsora: cuentan con los motores y las hélices que permite realizar los movimientos en dirección traslacional y rotacional, como son:

- Up / down
- Left / right
- Foreward / backward

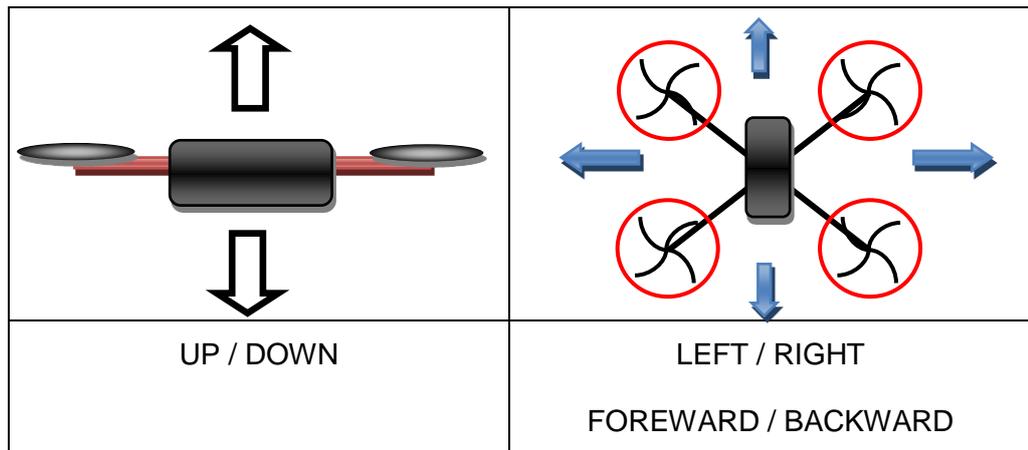


Figura 2.3. Movimiento traslacional

- Roll: rotación respecto del eje X.
- Pitch: rotación respecto del eje Y.
- Yaw: rotación alrededor del eje Z.

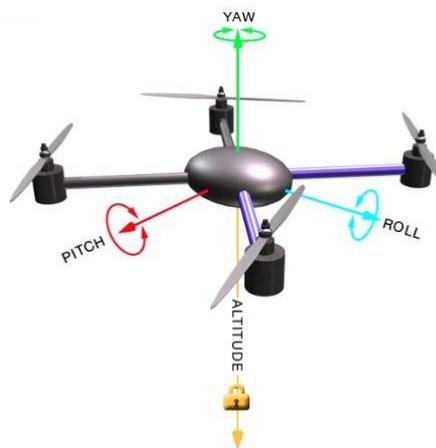


Figura 2.4. Movimiento Rotacional: Yaw, Pitch, Roll. [11]

El modelo matemático tanto cinético y dinámico del cuadricóptero se lo obtiene al derivar las ecuaciones de Newton-Euler, pero antes se deben considerar los siguientes supuestos:

- La estructura del cuadricóptero (considerando carcasa y soporte de partes) es rígida y simétrica para evitar cambios en su centro de masa.
- El centro de gravedad de los 4 rotores coinciden con el centro de masa de la estructura.
- Las hélices son consideradas rígidas para evitar el cambio de velocidad y por ende giros inesperados.
- El empuje y la resistencia son proporcionales al cuadrado de la velocidad de la hélice.
- Cada rotor del cuadricóptero tendrá su controlador de manera independiente, pero para su funcionamiento se trabaja en pares.
- Mientras los rotores de (1,3) giran en un sentido, (2,4) giran en sentido opuesto.
- Cada par (1,3) o (2,4) de rotores girando en la misma dirección, controlan un eje de, Cabeceo - Pitch, o Balanceo - Roll, permitiendo la maniobrabilidad del cuadricóptero.

Con estas consideraciones se puede modelar las ecuaciones matemáticas para determinar el movimiento cinemático distancia \mathbf{r} desde una posición referida de origen. Primero se debe definir los ejes de coordenada X, Y y Z referidos al eje de la Tierra N, E y D que se usarán para orientar de manera absoluta el centro de masa del cuadricóptero. Se tiene que la distancia se la puede representar como un vector \mathbf{r} definido por las coordenadas x, y, z.

$$\mathbf{r} = [x \ y \ z]^T \quad (1)$$

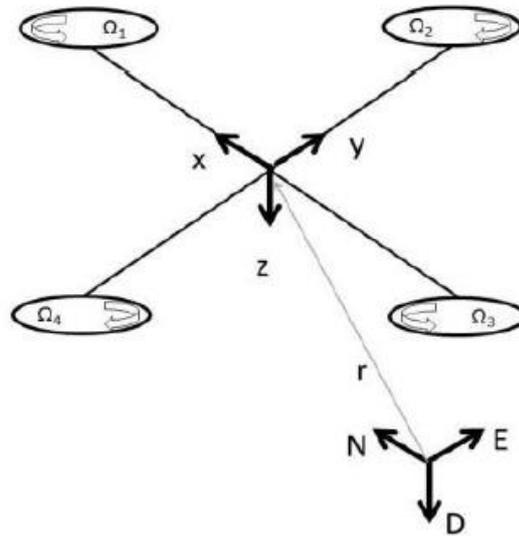


Figura 2.6. Ejes de referencia de cuadricóptero. [16]

Para representar la orientación del cuadricóptero se utilizarán 3 ángulos que describirán su rotación sobre los ejes x , y , y z respectivamente denominados:

Roll = Balance = ϕ	Pitch = Cabeceo = θ	Yaw = Guiñada = ψ
-------------------------	----------------------------	------------------------

Tabla 1: Movimientos y ángulos relacionados en un cuadricóptero.

De esta manera se puede obtener la matriz de R que define de forma matemática el movimiento cinemático;

$$R = \begin{bmatrix} \cos\theta * \cos\psi & \text{sen}\phi * \text{sen}\theta * \cos\psi & \cos\phi * \cos\psi + \text{sen}\phi * \text{sen}\psi \\ \cos\theta * \text{sen}\psi & \text{sen}\phi * \text{sen}\theta * \text{sen}\psi & \cos\phi * \text{sen}\theta * \text{sen}\psi - \text{sen}\theta * \text{sen}\psi \\ -\text{sen}\theta & \text{sen}\phi * \cos\theta & \cos\phi * \cos\theta \end{bmatrix} \quad (2)$$

y para el modelo dinámico del movimiento del cuadricóptero se puede dividir en dos subsistemas; rotacional y traslacional.

La ecuación para la representación del sistema rotacional viene dada por la siguiente ecuación:

$$J\dot{w} + w * Jw + [0 \ 0 \ J_r \Omega_r]^T = M_B \quad (3)$$

Donde:

J matriz de inercia diagonal del cuadricóptero

w velocidad angular del cuerpo

M_B Momento actuante en el cuadricóptero en el centro del cuerpo.

J_r Inercia de los rotores

Ω_r velocidad relativa de los rotores

Y la ecuación de Fuerza y Momento actuantes sobre el cuadricóptero se representan de la siguiente forma. Gráficamente de cómo actúan sobre el cuadricóptero se lo puede visualizar en la representación gráfica de la figura 2.5.

$$F_i = \left(\frac{1}{2} \rho A C_t r^2 \right) \Omega_i^2 = K_f \Omega_i^2 \quad (4)$$

$$M_i = \left(\frac{1}{2} \rho A C_d r^2 \right) \Omega_i^2 = K_m \Omega_i^2 \quad (5)$$

Donde:

ρ Densidad del aire

A Área de la hélice

C_d, C_t Coeficientes aerodinámicos

r Radio de las hélices

Ω_i Velocidad angular del rotor i

K_f Constante de fuerza aerodinámica

K_m Constante de momento aerodinámica

Al realizar la combinación de ecuaciones para calcular los momentos referentes a cada eje se tienen los momentos en, M_x , M_y , M_z :

$$M_x = l K_f (\Omega_4^2 - \Omega_2^2) \quad (6)$$

$$M_y = l K_f (\Omega_1^2 - \Omega_3^2) \quad (7)$$

$$M_z = K_m (\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \quad (8)$$

Con la combinación de las ecuaciones 5, 6, 7 se obtiene la representación de los momentos actuantes en el cuadricóptero representado de forma vectorial.

$$M_B = \begin{bmatrix} l K_f (\Omega_4^2 - \Omega_2^2) \\ l K_f (\Omega_1^2 - \Omega_3^2) \\ K_m (\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \end{bmatrix} \quad (9)$$

El objetivo del presente estudio no es desarrollar las ecuaciones del modelo matemático para el estudio del modelo en espacio de estados, si no comprender las ecuaciones que dominan y permiten el movimiento tanto traslacional y rotacional.

Ahora para poder realizar la simulación es necesario llevar el modelo matemático a una representación de espacios de estados. Se podrá modelar y simular desde la planta, hasta su sistema de control, para lo cual debemos conocer las ecuaciones que rigen este sistema.

Para ello vamos a tomar de referencia ya las ecuaciones de espacio de estado definidas en bibliografía [16] en el cual nos basamos para el desarrollo de este proyecto, donde:

La ecuación que define la posición del cuadricóptero en el espacio y sus velocidades angulares y lineales está dada de la siguiente forma:

Vector de control de entrada, este vector consiste en cuatro entradas definidas de la siguiente manera:

$$U_1 = K_f(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \quad (10)$$

$$U_2 = K_f(-\Omega_2^2 + \Omega_4^2) \quad (11)$$

$$U_3 = K_f(\Omega_1^2 - \Omega_3^2) \quad (12)$$

$$U_4 = K_m(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \quad (13)$$

Las velocidades del rotor, se la puede calcular a partir de las entradas de control U y despejando $\Omega_1, \Omega_2, \Omega_3, \Omega_4$, quedando como sigue:

$$\Omega_1 = \sqrt{\frac{1}{4K_f} U_1 + \frac{1}{2K_f} U_3 + \frac{1}{4K_m} U_4} \quad (14)$$

$$\Omega_2 = \sqrt{\frac{1}{4K_f} U_1 - \frac{1}{2K_f} U_3 - \frac{1}{4K_m} U_4} \quad (15)$$

$$\Omega_3 = \sqrt{\frac{1}{4K_f} U_1 - \frac{1}{2K_f} U_3 + \frac{1}{4K_m} U_4} \quad (16)$$

$$\Omega_4 = \sqrt{\frac{1}{4K_f} U_1 + \frac{1}{2K_f} U_3 - \frac{1}{4K_m} U_4} \quad (17)$$

La forma general de la ecuación de traslación de movimiento está dada por:

$$m\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + RF_B \quad (18)$$

donde;

$r = [x \ y \ z]^T$ distancia del cuadricóptero desde el marco de referencia. Figura 2.6

m masa del cuadricóptero

g aceleración gravitacional $g = 9.81 \text{ m/s}^2$

F_B fuerza no gravitacional que actúa sobre el cuerpo del cuadricóptero

R matriz de rotación (2)

La fuerza no gravitacional actuante en el cuadricóptero puede ser expresada como:

$$F_B = \begin{bmatrix} 0 \\ 0 \\ -U_1 \end{bmatrix} \quad (19)$$

Reescribiendo la ecuación se tiene el modelo matemático del cuadricóptero, con el que se puede simular el sistema y diseñar el controlador para mejorar la respuesta deseada para el control de la altitud.

$$\ddot{x} = \frac{-U_1}{m} (\sin \phi \sin \psi + \cos \phi \cos \psi \sin \theta) \quad (20)$$

$$\ddot{y} = \frac{-U_1}{m} (\cos \phi \sin \psi \sin \theta - \sin \phi \cos \psi) \quad (21)$$

$$\ddot{z} = \frac{U_1}{m} (\cos \phi \cos \theta) \quad (22)$$

2.3 Controlador PID clásico aplicado al control de un cuadricóptero

El controlador PID de bucle simple son de uso común y es el más conocido para procesos industriales, ya que por su fácil configuración se pueden aplicar y obtener mejores resultados. El objetivo del presente estudio es el obtener el diseño de un PID clásico para controlar la posición altura z en un cuadricóptero. También lograr estabilizar su posición ante perturbaciones externas.

Aunque el PID clásico no es adecuado para sistemas no lineales, se puede aplicar para el control de procesos de accionamiento de motor, control de vuelo, trenes de alta velocidad, entre otras, ya que no requiere un conocimiento explícito de la planta.

Teniendo el modelo de la planta del cuadricóptero en espacios de estado y utilizando la herramienta de Análisis Lineal de Simulink, se linealiza el modelo de la planta para obtener un controlador que cumpla con los siguientes parámetros de diseño:

- Set Point: $(x = 1, y = 1, z = 1)$ m
- Sobre nivel porcentual $\leq 20\%$ respecto al set point
- Tiempo de estabilización: ≤ 3 segundos
- Error de estado estable < 0.5

Se utilizan estos parámetros para simular la respuesta a una entrada de tipo escalón unitario. Considerando que para el arranque del cuadricóptero se debe ejercer una fuerza de impulso en contra de la gravedad, por tanto se permite en el arranque un sobrenivel porcentual del 20%. Al posicionar al cuadricóptero a un nivel de referencia, se necesita que su respuesta sea rápida para contrarrestar el efecto de la gravedad. Por tanto se desea que su tiempo de estabilización sea menor o igual a 3 segundos. Se requiere que el error de estado estable sea menor que 0.5, para tener precisión con respecto del punto de operación. Esta precisión es necesaria para que se pueda realizar la inspección sin mayor movimiento.

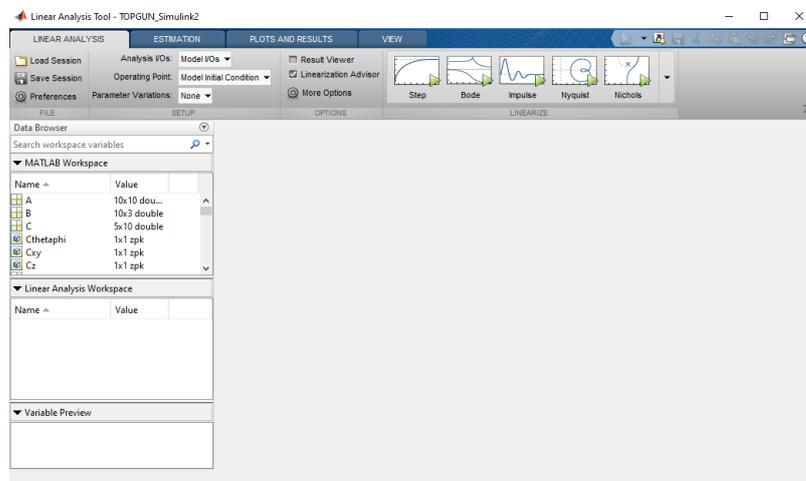


Figura 2.7 Herramienta Linear Analysis - Simulink®

En la teoría de sistemas de control moderno encontramos varios métodos para el diseño de controladores como el método del lugar geométrico de las raíces, el método de la respuesta de frecuencia y el método de análisis en el dominio del tiempo con variables de estado.

Para el desarrollo del controlador clásico PID utilizaremos el método del lugar geométrico de las raíces. Es un método gráfico con el que se dibuja la posición deseada de las raíces en el plano s de acuerdo a los requerimientos de diseño. En este método se utiliza la función de transferencia para convertir las ecuaciones diferenciales que representan el sistema en ecuaciones algebraicas en términos de la variable compleja s .

Una vez al contar con la función de transferencia linealizada sobre el punto de operación se puede proceder utilizando la herramienta de Matlab - Sisotool, para poder diseñar el controlador PID que cumpla con los parámetros establecidos de sobre nivel porcentual, tiempo de estabilización, y error de estado estable.

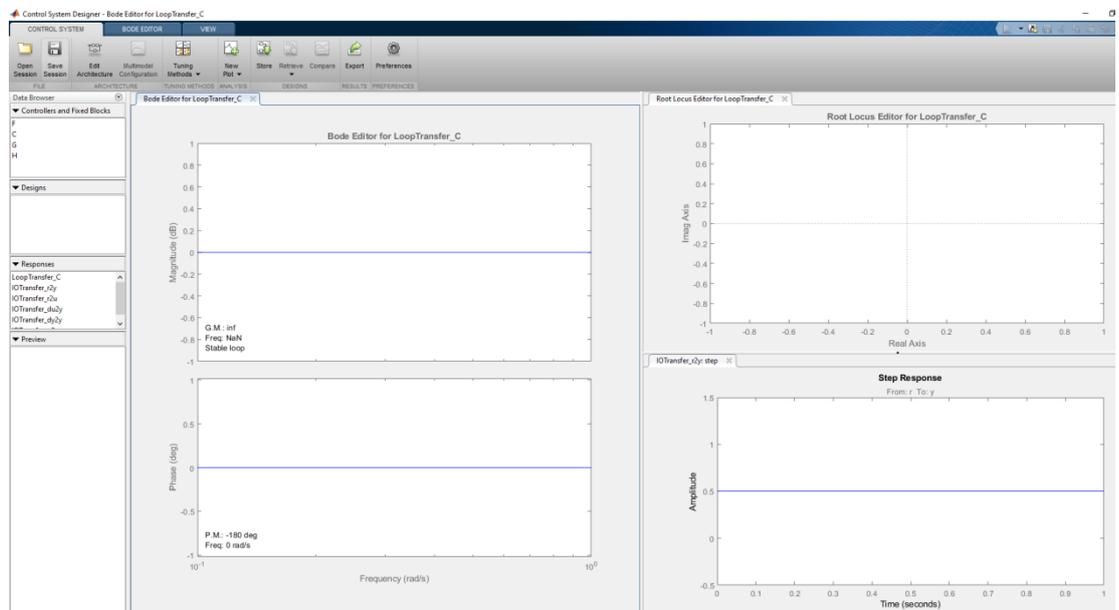


Figura 2.8 Herramienta Sisotool - Matlab®

Al contar con el controlador PID y los valores de K_P , K_I , K_D , se realiza la simulación por medio de Simulink agregando los valores del controlador PID y graficar la respuesta alrededor del punto de operación y verificar que la señal cumple con los parámetros de diseño.

A su vez, por su independencia, se trabajará a los controladores en cascada, de tal manera que la salida de un controlador de realimentación sirva como entrada y ajuste para otro controlador de realimentación, con el fin de obtener una respuesta estable, más rápida y mejorando la dinámica del lazo de control.

La estructura a utilizar tiene dos lazos, un lazo primario con un control primario también llamado maestro C_2 y un lazo secundario también denominado esclavo C_1 , siendo la salida del control primario, el punto de consigna del control secundario y la salida del control secundario, será el que actúe sobre la planta.

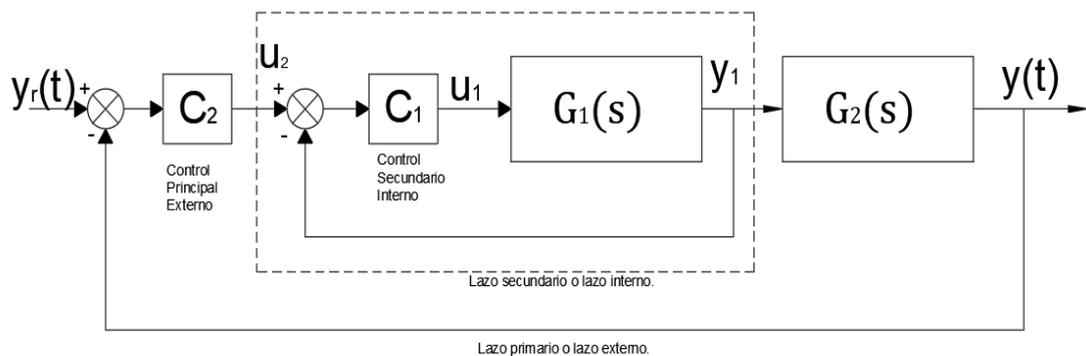


Figura 2.9 Estructura de control en cascada.

2.4 Controlador PID-Fuzzy aplicado al control de un cuadricóptero

Entre las limitaciones del control convencional que se presentan al momento de utilizar un controlador PID clásico, es que presentan dificultades para controlar procesos no lineales y que cambian constantemente produciendo resultados limitados. Por lo cual el controlador Fuzzy-PID cuenta con la característica de comportarse de mejor manera que un controlador PID clásico con sistemas No-lineales. Además del comportamiento temporal para la planta, los controladores clásicos varían en el tiempo, haciéndolos difíciles de modelar.

Los controladores Fuzzy-PID tienen características y beneficios que los hacen más adecuados para el uso en sistemas no lineales, tales como los enuncia en Fuzzy Controllers de Leonid Reznik [16].

- Los controladores difusos son más robustos que los controladores PID porque pueden cubrir una gama mucho más amplia de condiciones de funcionamiento que los PID y pueden funcionar con ruido y perturbaciones de diferentes naturalezas.
- Desarrollar un controlador difuso es más barato que desarrollar un controlador basado en modelos u otro para hacer lo mismo.
- Los controladores difusos son personalizables, ya que es más fácil de entender y modificar sus reglas, que no solo utilizan la estrategia de un operador humano, sino que también se expresan en términos lingüísticos naturales.
- Es fácil aprender cómo funcionan los controladores difusos y cómo diseñarlos y aplicarlos a una aplicación concreta.
- El cuadricóptero es muy sensible a las condiciones externas, como el viento, por lo que un cambio repentino conduce a la inestabilidad durante un vuelo estacionario. Por medio de la lógica difusa se puede lograr la estabilización cambiando las reglas lingüísticas para adaptarse a los cambios ambientales.

Hemos visto que una de las ventajas de los controladores difusos es que permite modelar términos del lenguaje natural por medio de variables lingüísticas.

Variables lingüísticas.

Las variables lingüísticas son aquella a la que se le puede dar un valor de una expresión de lenguaje natural o artificial. De esta manera sirve para representar cualquier elemento que sea complejo, o del cual no tengamos una definición concreta, es decir lo que no podamos describir en términos numéricos. [17]

Una variable lingüística es un término compuesto por una concatenación de términos atómicos, las cuales se pueden dividir en cuatro categorías.

- Términos primarios, que son los nombres de subconjuntos difusos específicos del universo del discurso, por ejemplo, pequeño, grande, bajo, mediano, alto;
- Conectores lógicos permiten realizar operaciones de intersección, unión y complemento y se utilizan "y", "o" y la negación "No";
- Modificadores como MUY, MÁS, MÁS DERECHO, LIGERAMENTE, MÁS O MENOS, permitirán dar mayor valor a una variable lingüística, y mejoran la acción del término.

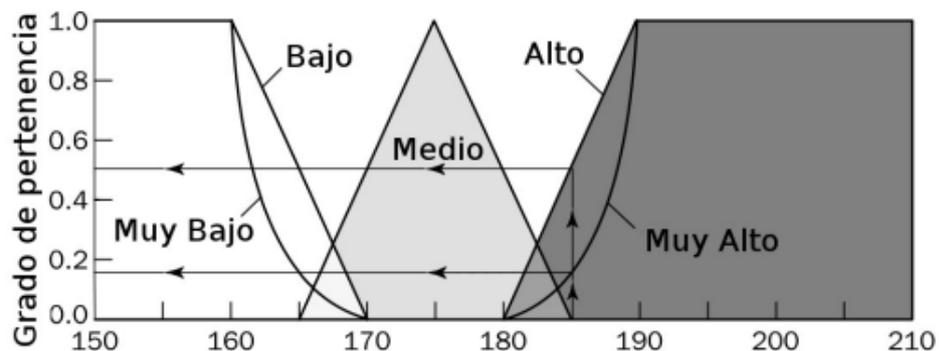


Figura 2.10 Representación de reglas difusas [17].

Reglas Difusas

Las reglas difusas son proposiciones que permiten expresar el conocimiento que se dispone de las entradas llamadas antecedentes o premisas y se lo asocia un conjunto difuso de salida llamada consecuente.

Para expresar este conocimiento de manera completa se sigue una estructura o base de reglas, el cual determinara cual será el comportamiento del controlador difuso y es aquí donde se emula la experiencia del operario.

Una regla difusa son afirmaciones del tipo SI-ENTONCES, y es expresada simbólicamente como:

$$SI < Proposición difusa > ENTONCES < Proposición difusa > \quad (23)$$

En sistemas FUZZY donde el antecedente es difuso, todas las reglas se ejecutan parcialmente, y el consecuente es cierto en cierto grado; si el antecedente es cierto con cierto grado de pertenencia, el consecuente es cierto también en cierto grado.

Existe una gran variedad de tipos de reglas, dos grandes grupos son los que en general se emplean, las reglas difusas de Mamdani y las reglas difusas de Takagi-Sugeno (TS). [17]

Inferencia de Mamdani.

La inferencia Mamdani es utilizada por conformar una estructura netamente de variables lingüísticas tanto en el antecedente como en el consecuente, como se muestra a continuación: [17]

Antecedente	Consecuente
<i>SI x_1 es A y x_2 es B y x_3 es C</i>	<i>ENTONCES u_1 es D, u_2 es E</i>

Tabla 2: Estructura de inferencia Mamdani

Donde:

- x_1, x_2, x_3 , son las variables de entrada (Ej. Error posición, derivada del error de posición)
- A, B y C son funciones de membresía de entrada (Ej. positivo, cero, negativo)
- u_1, u_2 son las acciones de control (Ej. Aumento de voltaje)
- D, E son las funciones de membresía de salida.
- Y es un operador lógico difuso.

Inferencia de Takagi-Sugeno (TS)

A diferencia de la inferencia de Mamdani, el consecuente en este método es posible emplear funciones matemáticas, por lo que la estructura está dada de la siguiente manera: [17]

Antecedente	Consecuente
<i>SI x_1 es A y x_2 es B y x_3 es C</i>	<i>ENTONCES</i> $u_1 = f(x_1, x_2, x_3)$, $u_2 = g(x_1, x_2, x_3)$

Tabla 3: Estructura de inferencia Takagi-Sugeno

Donde:

- x_1, x_2, x_3 , son las variables de entrada (Ej. Error posición, derivada del error de posición)
- A, B y C son funciones de membresía de entrada (Ej. positivo, cero, negativo)
- u_1, u_2 son las acciones de control (Ej. Aumento de voltaje)
- $f(), g()$ son funciones.

Este método presenta mayor eficiencia, pero no presenta un marco tan natural para la representación del conocimiento humano.

Para el desarrollo del presente trabajo, se utilizará la inferencia de Mamdani, por ser intuitivo, tiene una amplia aceptación y está bien adaptada a la incorporación de conocimientos y experiencia, a diferencia de la inferencia de TS, al estar mejor adaptado al análisis matemático, al uso de funciones en el consecuente.

Para la aplicación del método de Inferencia de Mamdani, se deben aplicar los siguientes 4 pasos.

1. Fuzificación de las variables de entrada
2. Evaluación de las reglas
3. Agregación de las salidas de las reglas
4. Defuzificación

Fuzificación

El primer paso consiste en tomar los valores crisp (valores numéricos máximo y mínimo que dan los grados de pertenencia a los valores de entrada, usualmente entre $[0,1]$) de las entradas y determinar el grado de pertenencia de estas entradas a los conjuntos difusos asociados. Para nuestro caso se va a determinar los valores en función de la experiencia para los valores de entrada y en función del comportamiento de la planta no lineal.

Evaluación de las reglas.

Se debe tomar las entradas y se debe aplicar a los antecedentes las reglas difusas, si una regla tiene múltiples antecedentes, se utilizan los operadores "y", "o", para obtener un único número que represente el resultado de la evaluación. Este número el valor de verdad se aplica al consecuente.

Finalmente el resultado de la evaluación del antecedente se aplica al consecuente, aplicando un recorte o escalado según el valor de verdad del antecedente. El método más comúnmente utilizado es el recorte (clipping) que corta el consecuente con el valor de verdad del antecedente. El escalado proporciona un valor más preciso, preservando la forma original del conjunto difuso.

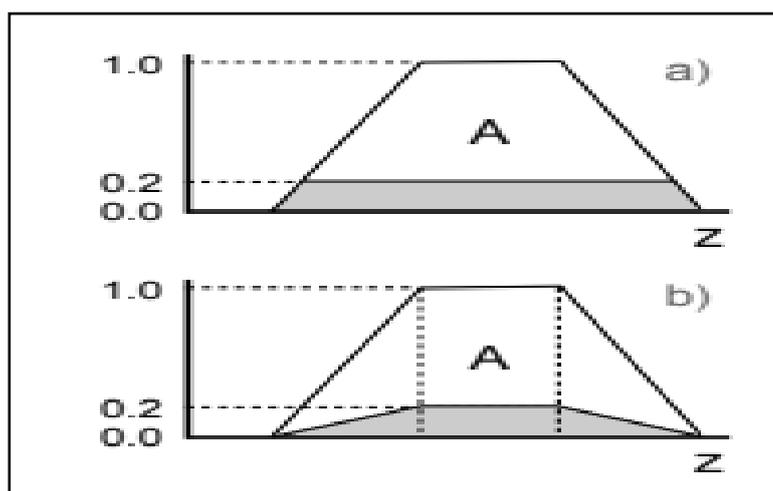


Figura 2.11. Conjunto recortado (a), y escalado (b) [17]

Agregación de las salidas

La agregación de las salidas es el proceso de unificación de todas las reglas, es decir se combinan las funciones de pertenencia de todos los consecuentes previamente recortados o escalados, combinando para obtener un único conjunto difuso por cada variable de salida.

Defuzificación

El resultado final habitualmente es necesario expresarlo mediante un valor crisp. En esta etapa se toma como entrada el conjunto difuso anteriormente obtenido para dar un valor numérico a la salida. El sistema de inferencia obtiene una conclusión a partir de la información de la entrada pero en términos difusos. Para la lectura de los resultados, el método más común utilizado es el de la defuzificación del centroide, que calcula el punto donde una línea vertical divide el conjunto en dos áreas con igual masa, transformando la salida difusa en un número real.

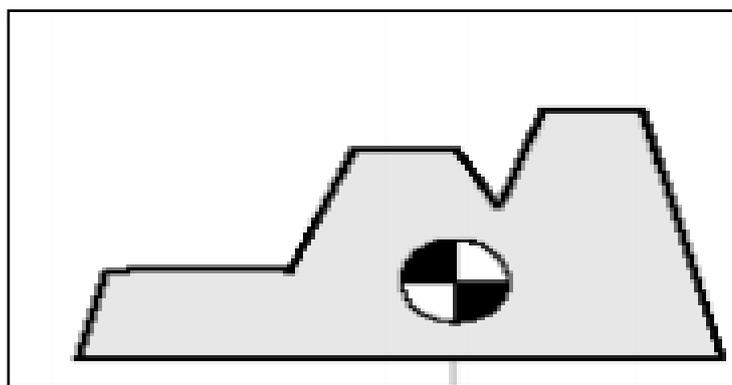


Figura 2.12. Defuzificación con el método del centroide

Para el presente proyecto utilizamos la herramienta Fuzzy Logic Designer, para aplicar el proceso de Mamdani, donde se ingresaron las reglas lógicas en el antecedente y se obtienen los valores numéricos del consecuente por medio del método del fuzificación del centroide, el cual permitió evaluar las reglas lógicas a aplicar.

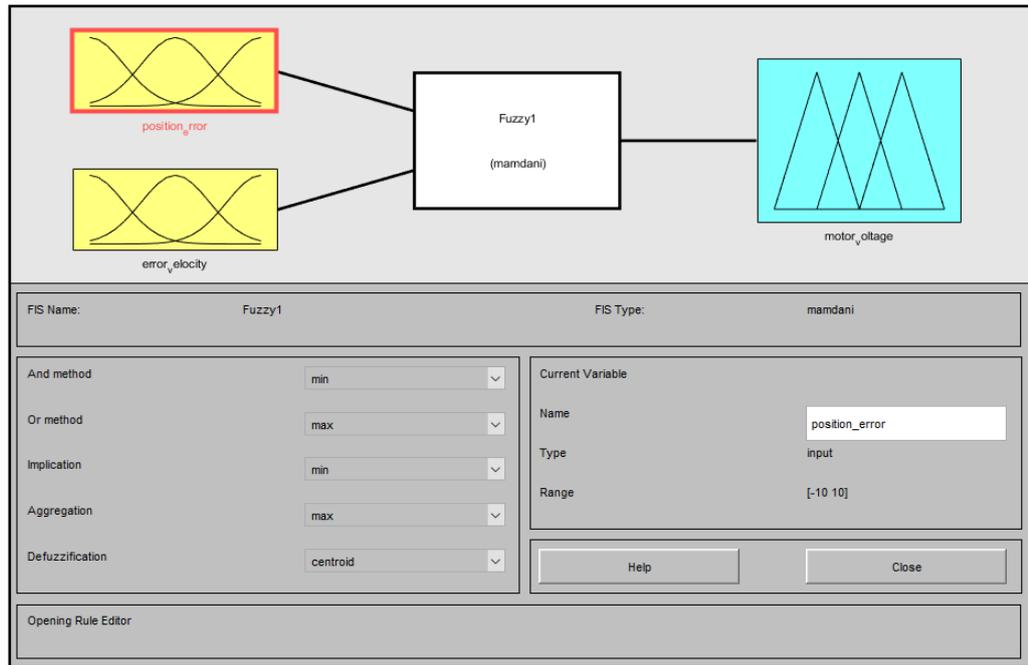


Figura 2.13. Fuzzy Logic Designer - Matlab®

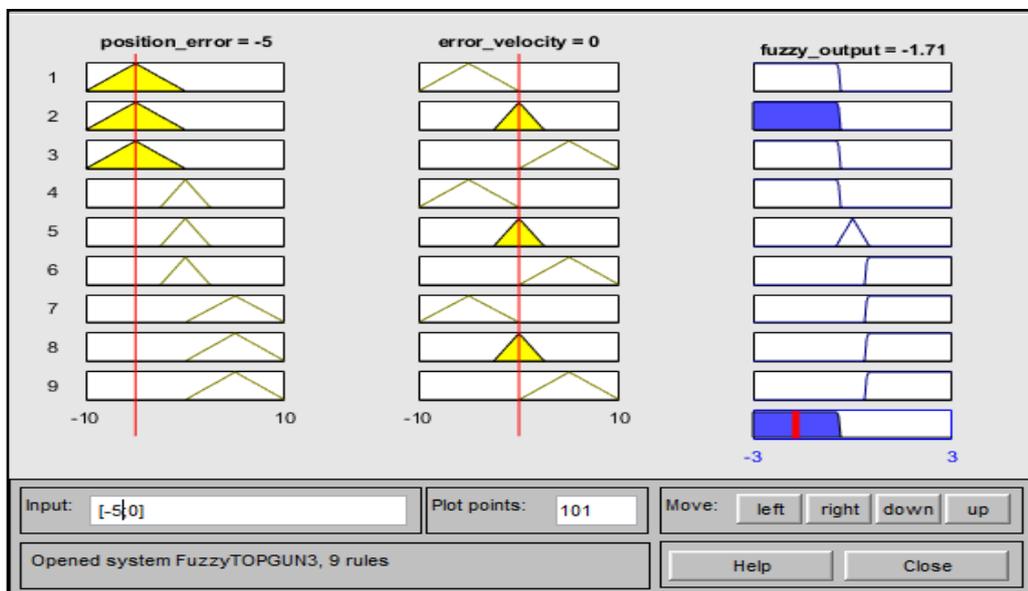


Figura 2.14 Fuzzy Logic Designer Evaluación de Reglas - Matlab®

CAPÍTULO 3

DISEÑO DE CONTROLADORES PID Y PID-FUZZY

3.1. Diseño y sintonización basada en modelo del controlador PID

Para el controlador PID de un cuadricóptero, partimos del modelo de la planta ya establecida en el capítulo anterior en representación de espacios de estados. Se tiene:

$$\ddot{x} = \frac{-U_1}{m} (\sin \phi \sin \psi + \cos \phi \cos \psi \sin \theta) \quad (20)$$

$$\ddot{y} = \frac{-U_1}{m} (\cos \phi \sin \psi \sin \theta - \sin \phi \cos \psi) \quad (21)$$

$$\ddot{z} = \frac{U_1}{m} (\cos \phi \cos \theta) \quad (22)$$

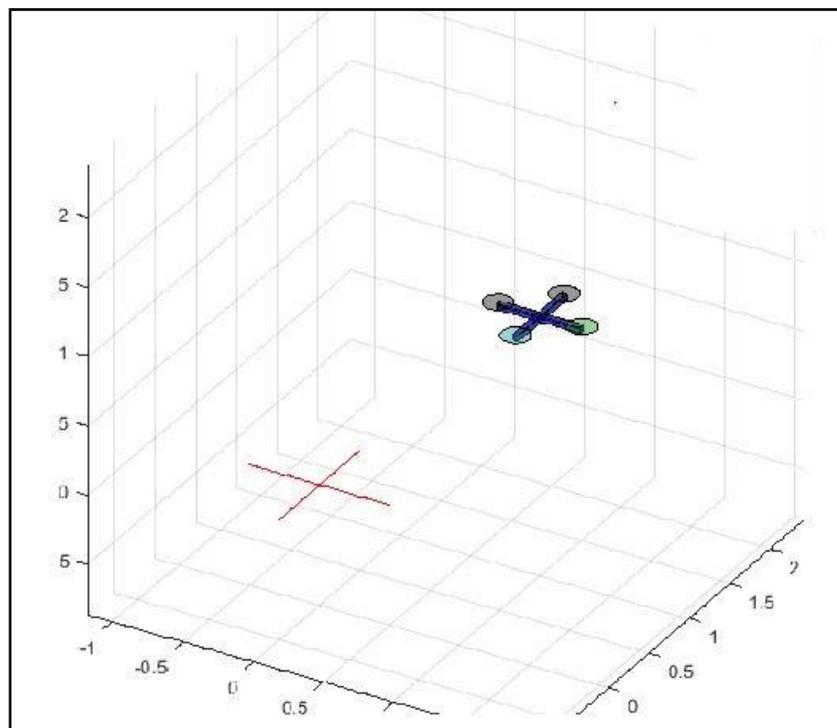


Figura 3.1 Simulación de Cuadricóptero 3D - Matlab®

Representación de Modelo no Lineal

El modelo matemático de un cuadricóptero es de naturaleza no lineal, como se lo puede ver en las ecuaciones 20, 21 y 22 que son las que se utilizan para realizar la simulación de la planta en la herramienta de Matlab® - Simulink. En las figura 3.2 se muestra la representación del sistema y en la figura 3.3. se representa las entradas y salidas necesarias para realizar la simulación del control de altitud de un cuadricóptero.



Figura 3.2. Entradas y Salidas representación del Sistema Cuadricóptero

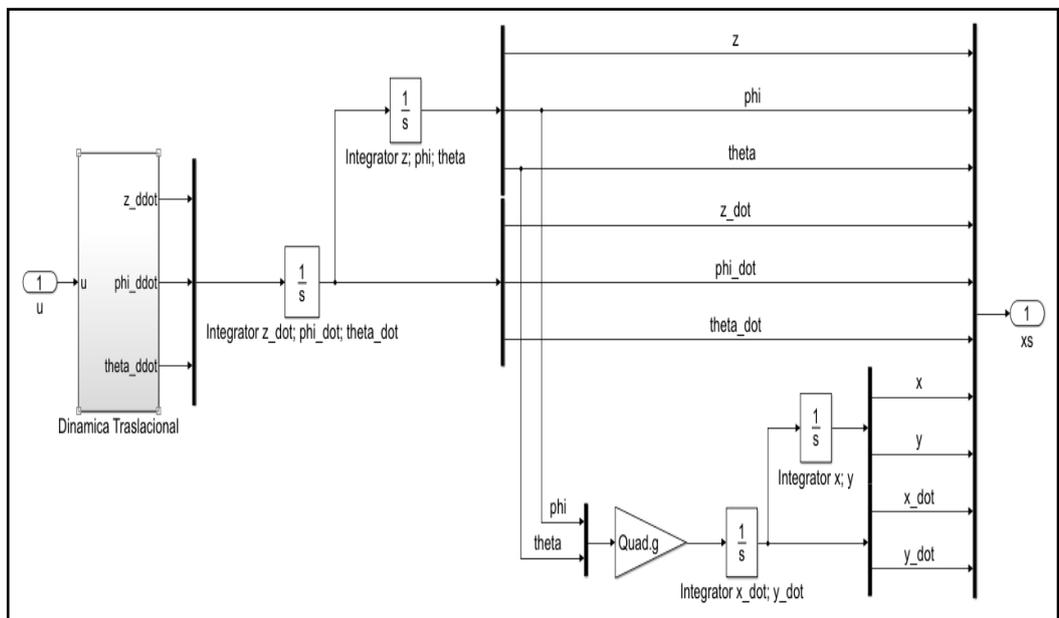


Figura 3.3. Modelo no lineal Cuadricóptero - Matlab® - Simulink

En este se representa las ecuaciones no lineales del modelo traslacional que permitirá accionar y realizar los movimientos en el cuadricóptero.

Dinámica traslacional

En la dinámica traslacional se representa la entrada "U", que son los vectores de control de entrada, ecuaciones U_1, U_2, U_3 , y se visualizan la salida de altura \ddot{z} , y posición angular $\ddot{\phi}$ y $\ddot{\theta}$ que controlan el movimiento respecto al eje X y Y respectivamente. Para el modelo Simulink se adoptan las siguientes representaciones; $\ddot{z} = z_ddot$, $\ddot{\phi} = phi_ddot$ y $\ddot{\theta} = theta_ddot$.

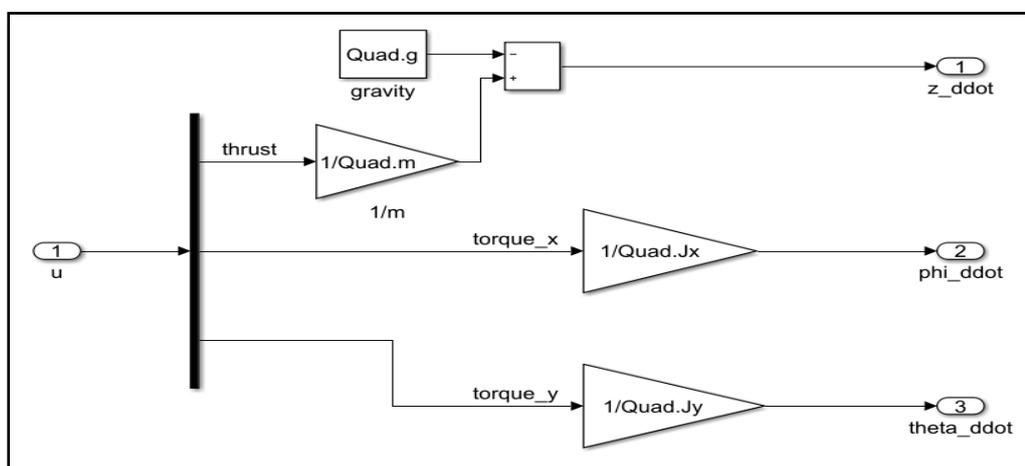


Figura 3.4. Modelo no lineal Dinámica traslacional- Matlab® - Simulink

Linealización del modelo

Para la aplicación de las técnicas de control que se van a emplear para este problema, se necesita linealizar este modelo. Para la linealización, se debe definir el punto de operación alrededor del cual vamos a realizar el control.

Primero, definimos los estados y las entradas iniciales en Matlab® - Simulink, los cuales son representados por los vectores x_0 y u_0 :

Se considera un punto de operación en el que no hay variaciones angulares en su posición, por tanto los valores de $\phi, \theta, z', \phi', \theta', x', y'$ se aproximan a 0.

Donde el estado Inicial " x_0 "

$$x_0 = \begin{bmatrix} 1.0000 \\ 0.0125 \\ 0.0125 \\ 1.0000 \\ 1.0000 \\ 0.2000 \\ 0.0025 \\ 0.0025 \\ 0.4000 \\ 0.4000 \end{bmatrix} = \begin{bmatrix} z \\ \phi \\ \theta \\ x \\ y \\ z' \\ \phi' \\ \theta' \\ x' \\ y' \end{bmatrix} \quad (24)$$

y de igual manera se considera que el cuadricóptero no rota en su posición, por tanto los valores de torque para τ_x y $\tau_y \cong 0$, por tanto la entrada inicial " u_0 "

$$u_0 = \begin{bmatrix} 13.762000 \\ 0.0000125 \\ 0.0000125 \end{bmatrix} = \begin{bmatrix} (m * g) + f_e \\ \tau_x \\ \tau_y \end{bmatrix} \quad (25)$$

Luego, utilizando el módulo Linear Analysis, obtenemos el punto de operación "TrimPointQuad01.mat" y finalmente las matrices A, B, C y D que nos permiten implementar el modelo sobre la plataforma de Simulink®

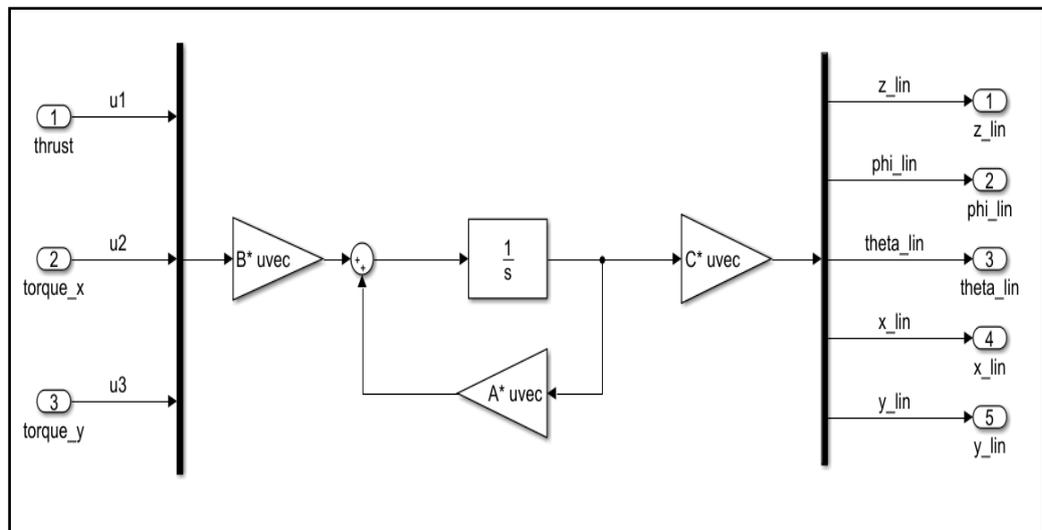


Figura 3.5. Modelo lineal del cuadricóptero - Matlab® - Simulink

Matrices de Espacio de Estados.

A =

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
x1	0	0	1	0	0	0	0	0	0	0
x2	0	0	0	1	0	0	0	0	0	0
x3	0	0	0	0	0	9.81	0	0	0	0
x4	0	0	0	0	0	0	9.81	0	0	0
x5	0	0	0	0	0	0	0	1	0	0
x6	0	0	0	0	0	0	0	0	1	0
x7	0	0	0	0	0	0	0	0	0	1
x8	0	0	0	0	0	0	0	0	0	0
x9	0	0	0	0	0	0	0	0	0	0
x10	0	0	0	0	0	0	0	0	0	0

Figura 3.6. Matriz estados vs estados - Matlab® - Simulink

B =

	u1	u2	u3
x1	0	0	0
x2	0	0	0
x3	0	0	0
x4	0	0	0
x5	0	0	0
x6	0	0	0
x7	0	0	0
x8	0.7143	0	0
x9	0	20	0
x10	0	0	20

Figura 3.7. Matriz estados vs entradas - Matlab® - Simulink

C =

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
y1	0	0	0	0	1	0	0	0	0	0
y2	0	0	0	0	0	1	0	0	0	0
y3	0	0	0	0	0	0	1	0	0	0
y4	1	0	0	0	0	0	0	0	0	0
y5	0	1	0	0	0	0	0	0	0	0

Figura 3.8. Matriz salidas vs estados - Matlab® - Simulink

D =			
	u1	u2	u3
y1	0	0	0
y2	0	0	0
y3	0	0	0
y4	0	0	0
y5	0	0	0

Figura 3.9. Matriz salidas vs entradas - Matlab® - Simulink

Tras la linealización se obtiene la función de transferencia en tiempo continuo $G(s)$, alrededor del punto de operación (1, 1,1)

$$G(s) = \begin{pmatrix} \frac{0.7143}{s^2} & 0 & 0 \\ 0 & \frac{20}{s^2} & 0 \\ 0 & 0 & \frac{20}{s^2} \\ 0 & \frac{196.2}{s^4} & 0 \\ 0 & 0 & \frac{196.2}{s^4} \end{pmatrix} \quad (26)$$

Donde se obtiene que;

La función de transferencia para el control del Empuje vs Z es:

$$G_z = \frac{0.71429}{s^2} \quad (27)$$

La función de transferencia para el control del torque que permite el movimiento del ángulo Phi (Roll) a un determinado valor.

$$G_{phi} = \frac{20}{s^2} \quad (28)$$

La función de transferencia para el control del torque que permite el movimiento del ángulo Theta (Pitch) a un determinado valor:

$$G_{theta} = \frac{20}{s^2} \quad (29)$$

La función de transferencia que permite el control el desplazamiento en función del torque para cada eje respectivamente (X o Y).

$$G_x = G_y = \frac{196.2}{s^4} \quad (30)$$

Diseño de controlador de altitud "Cz" utilizando herramienta Sisotool.

Utilizando el método del lugar geométrico de las raíces y por medio de la herramienta Sisotool, determinaremos el controlador PID adecuado para cada sistema, y que cumplan con los parámetros, sobre nivel porcentual $\leq 20\%$, tiempo de estabilización: ≤ 3 segundos y Error de estado estable $< 0.5\%$.

Con la ubicación de tres polos, un polo real y un par de polos complejos conjugados y, un par de zeros, se puede diseñar el controlador para el control del movimiento Roll, alrededor del eje X, y alrededor del eje Y. Aplicando el método del lugar geométrico de las raíces con la herramienta de Matlab - Sisotool, determinamos el controlador que mejor cumpla con los parámetros de diseño.

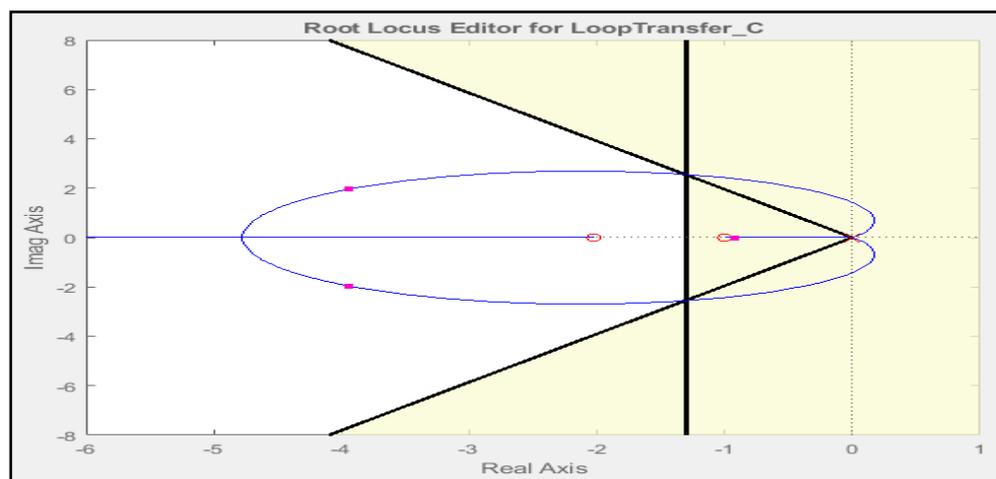


Figura 3.10. Lugar Geométrico de las raíces Controlador Cz.

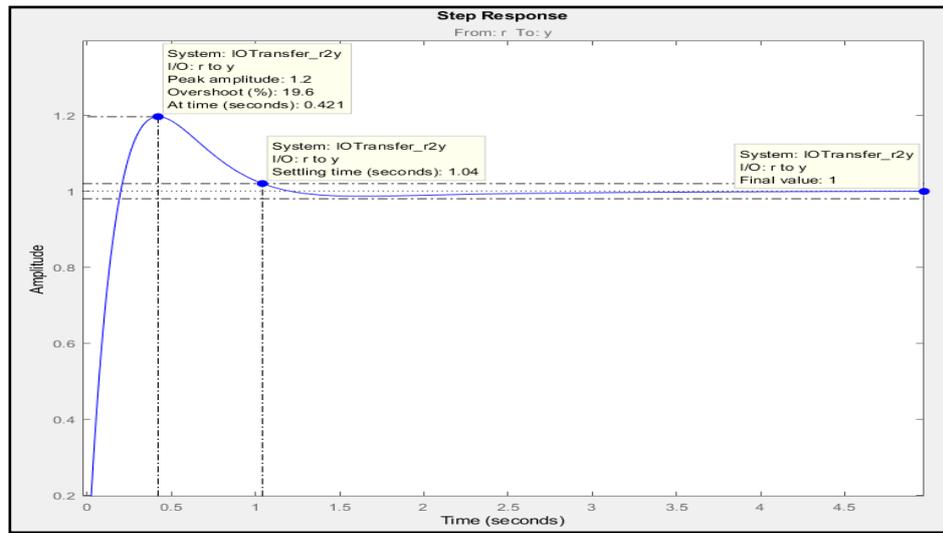


Figura 3.11 Respuesta de Controlador Cz.

Con dos zeros y un polo se puede observar que se cumplen con los parámetros de diseño requeridos para el controlador como se puede ver en la tabla 4.

Parametros	Requerido	Obtenido
OS%	<=20	19.6
Ts (s)	<=3	1.04
Ess	<0.5	0

Tabla 4: Parámetros de diseño para controlador Cz de altitud.

Función de transferencia del controlador de Altitud.

$$C_Z = \frac{12.312 (S + 1)(S + 2.025)}{S} \quad (31)$$

Donde los valores de K_p , K_i , K_d son:

$$k_p = k_{pz} = 37.2440 \quad (32)$$

$$k_i = k_{iz} = 24.9324 \quad (33)$$

$$k_d = k_{dz} = 12.3116 \quad (34)$$

Diseño de controlador de movimiento Roll (movimiento lateral) "Cphi" y "Ctheta" (movimiento de cabeceo - Pitch).

Con la ubicación de dos polos y un zero se puede diseñar el controlador para el control del movimiento Roll, alrededor del eje X, y alrededor del eje Y.

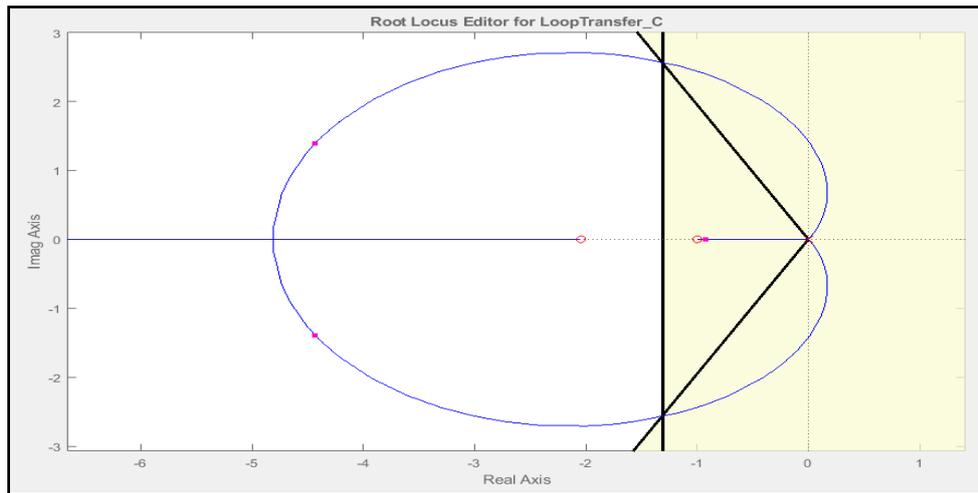


Figura 3.12 Lugar geométrico de las raíces C_{phi} y C_{theta} .

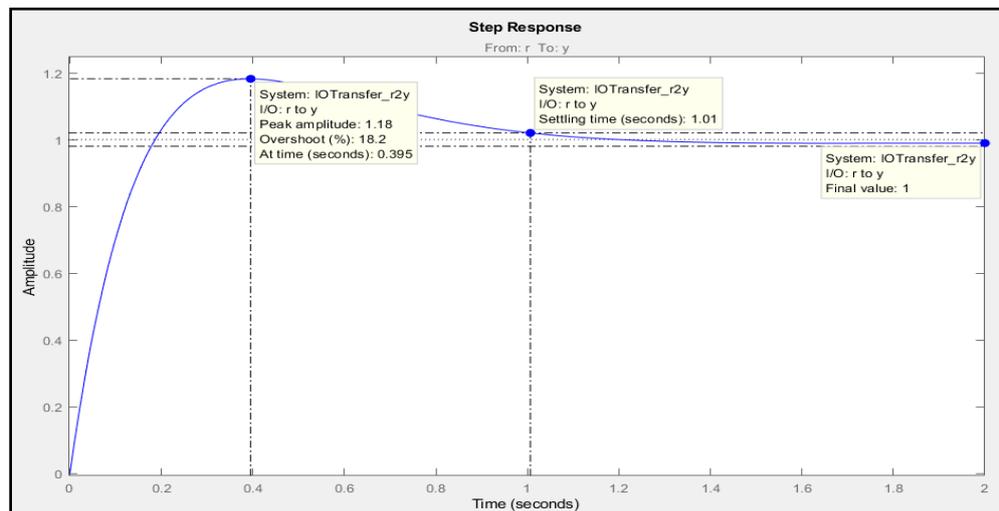


Figura 3.13 Respuesta de controlador C_{phi} y C_{theta} .

Con dos zeros y un polo se puede observar que se cumplen con los parámetros de diseño requeridos para el controlador como se puede ver en la tabla 5.

Parametros	Requerido	Obtenido
OS%	≤ 20	18.2
Ts (s)	≤ 3	1.01
Ess	< 0.5	0

Tabla 5: Parámetros para controlador C_{phi} - Roll y C_{theta} - Pitch

Función de transferencia del controlador de Roll y Pitch.

$$C_{theta} = C_{phi} = \frac{0.48935 (s + 1) (s + 2.044)}{s} \quad (35)$$

Donde los valores de K_p , K_i , K_d para G_{phi} y G_{theta} son:

$$k_p = k_{pphi} = k_{ptheta} = 1.4894 \quad (36)$$

$$k_i = k_{iphi} = k_{itheta} = 1.000 \quad (37)$$

$$k_d = k_{dphi} = k_{dtheta} = 0.4894 \quad (38)$$

Diseño de controlador de movimiento de Traslación en X y Y, que al ser un sistema simétrico para el movimiento efectuado, sus valores de K_p , K_i , K_d son iguales.

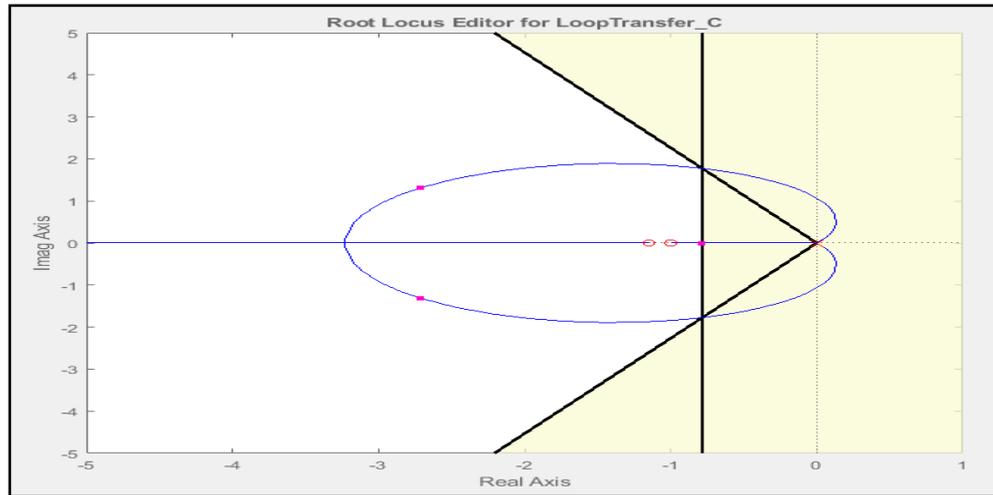


Figura 3.14 Lugar geométrico de las raíces de controlador de traslación: X - Y.

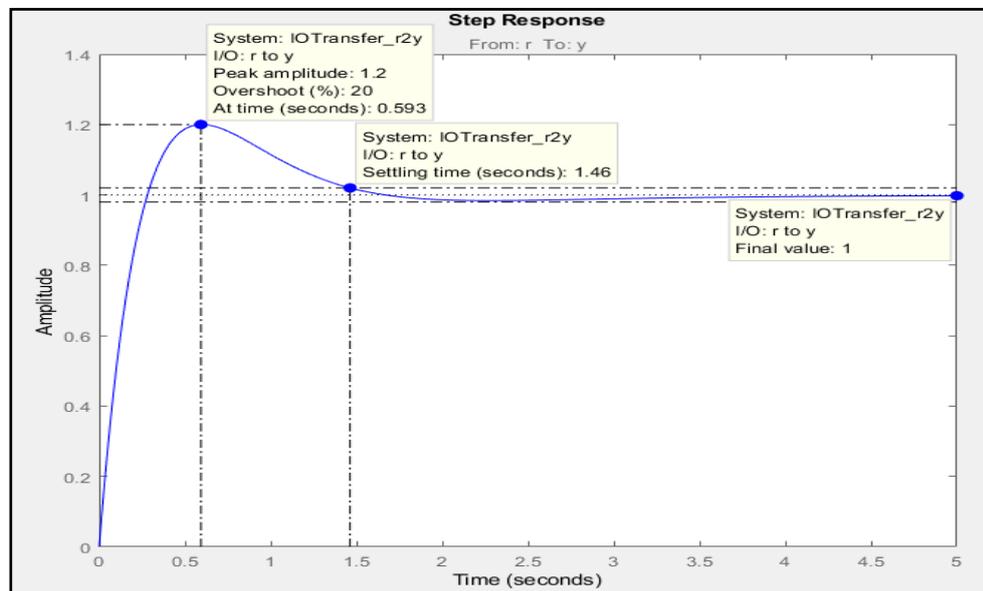


Figura 3.15 Respuesta de controlador de Traslación: X y Y.

Con dos zeros y un polo se puede observar que se cumplen con los parámetros de diseño requeridos para el controlador como se puede ver en la tabla 6.

Parámetros	Requerido	Obtenido
OS%	≤ 20	20
Ts (s)	≤ 3	1.46
Ess	< 0.5	0

Tabla 6: Parámetros para controlador de posición en X y Y.

Función de transferencia del controlador de posición X - Y

$$C_x = C_y = \frac{0.63389 (s + 1) (s + 1.149)}{s} \quad (39)$$

Donde los valores de K_p , K_i , K_d para el control de X y Y son:

$$k_p = k_{px} = k_{py} = 1.3621 \quad (40)$$

$$k_i = k_{ix} = k_{iy} = 0.7282 \quad (41)$$

$$k_d = k_{dx} = k_{dy} = 0.6339 \quad (42)$$

Simulación del modelo lineal utilizando los controladores encontrados.

La simulación del modelo se lo realiza en Simulink®, donde se colocan a los controladores diseñados en una estructura de cascada para ver su funcionamiento, de tal manera que la salida del controlador de traslación "X", sirva como entrada para el controlador de realimentación del ángulo de inclinación "Gphi". Este permite inclinar el cuadricóptero para su traslación y movimiento en X. De igual manera la salida del controlador de realimentación de traslación "Y", servirá como entrada para el controlador de realimentación que controla el ángulo de inclinación "Gtheta", que permite inclinar el cuadricóptero para su traslación y movimiento en Y, como se lo puede ver en la figura 3.16.

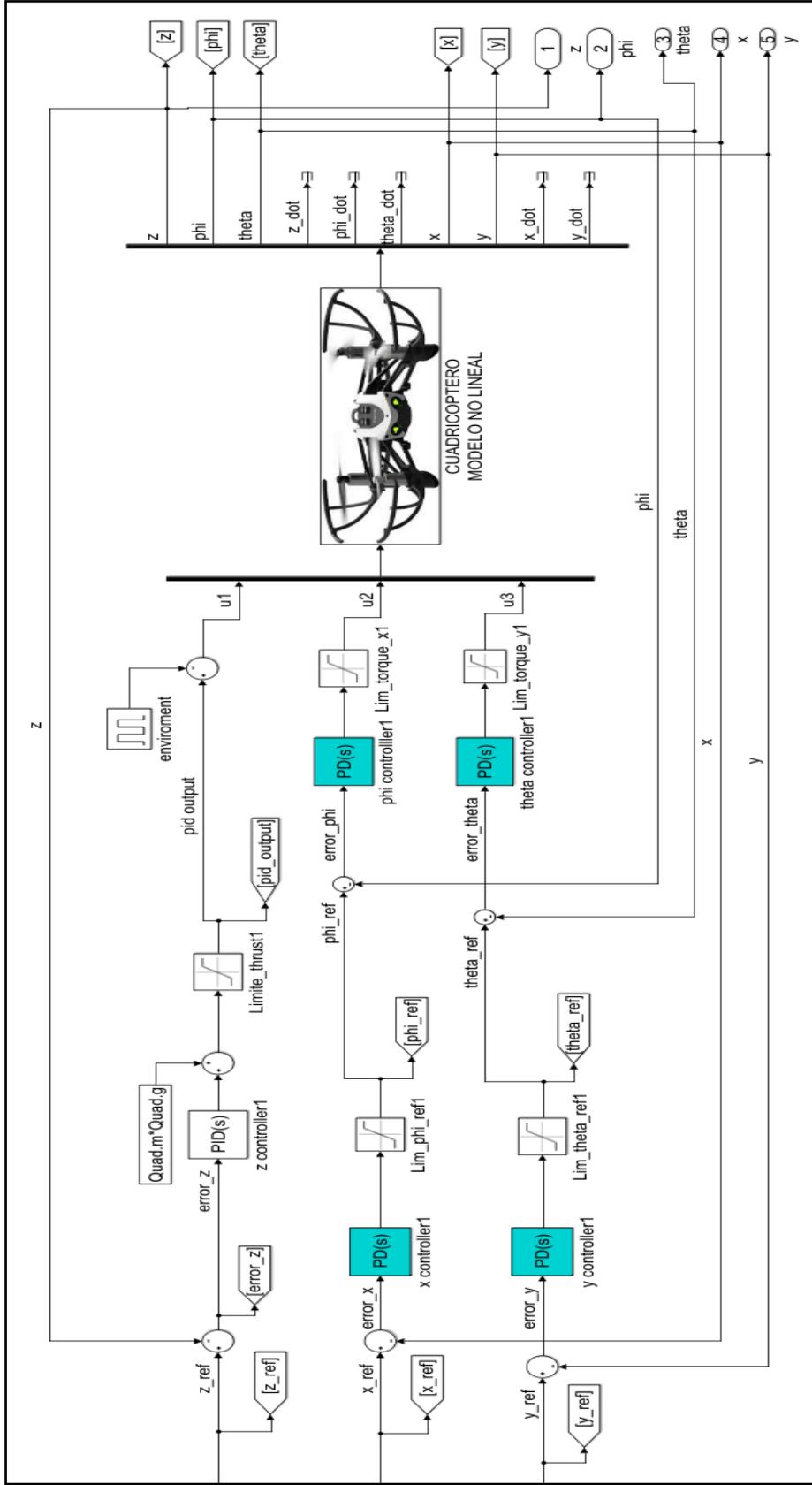


Figura 3.16 Arreglo de Controladores PD en Cascada en Simulink®.

Para el controlador del eje Z, no se realizará en estructura de cascada, al no depender de más que de la posición en altura, por lo que el control es de retroalimentación simple. Se implementaron todos los controladores en la plataforma Simulink®, como se muestra en la siguiente figura 3.17.

Posteriormente, se procedió a correr la simulación para obtener las gráficas de salidas y respuestas del sistema.

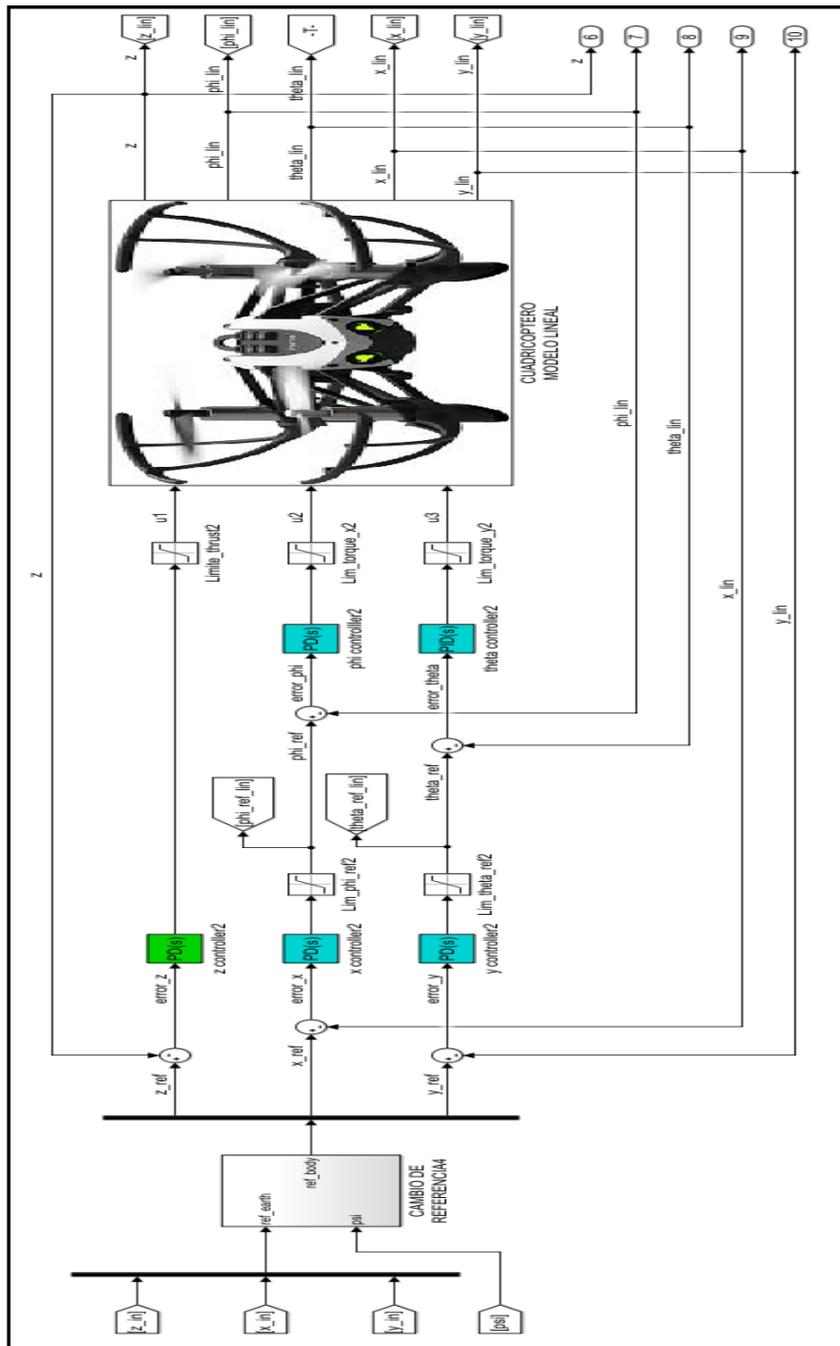


Figura 3.17 Simulación de Controlador PID de un Cuadróptero en Simulink.

3.2. Diseño de reglas de lógica difusa para el controlador FUZZY- PID

Al ser el modelo de la planta de un cuadricóptero un sistema no lineal, el controlador difuso es apropiado como aplicación de control. En este tipo de sistemas la aplicación de un Fuzzy-PID puede mejorar la respuesta para el control de altitud.

Para el diseño de las reglas del controlador difuso, del control de altura, nos basaremos la lógica difusa de Mamdani. Basados en la experiencia sobre el comportamiento lógico del vuelo de un cuadricóptero, para determinar las reglas lógicas del control de altura siguientes:

- Se empezará tomando el punto de referencia como cero.
- Se toman dos variables de entrada; error y el voltaje de los motores y una salida.
- Y se asignan los valores lingüísticos a las siguientes siglas.

NE: Error de posición negativo.	NV: Error de velocidad negativo	NO: Salida del controlador negativa
ZE: Error de posición cero	ZV: Error de velocidad cero	ZO: Salida del controlador cero
PE: Error de posición positiva	PV: Error de velocidad positiva	PO: Salida del controlador positiva

Tabla 7: Variables lingüísticas para reglas de lógica difusa.

De esta manera para las reglas lógicas se pueden ver en el siguiente arreglo.

V \ E	NE	ZE	PE
NV	NO	NO	PO
ZV	NO	ZO	PO
PV	NO	PO	PO

Tabla 8: Reglas de lógica difusa controlador de altitud.

Esto quiere decir que; si:

1. El error de altitud es negativo y el error de la velocidad es negativa, entonces la salida del controlador es negativa.
2. El error de altitud es negativo y el error de la velocidad es cero, entonces la salida del controlador es negativa.
3. El error de altitud es negativo y el error de la velocidad es positiva, entonces la salida del controlador es negativa.
4. El error de altitud es cero y el error de la velocidad es negativa, entonces la salida del controlador es negativa.
5. El error de altitud es cero y el error de la velocidad es cero, entonces la salida del controlador es cero.
6. El error de altitud es cero y el error de la velocidad es positiva, entonces la salida del controlador es positiva.
7. El error de altitud es positivo y el error de la velocidad es negativa, entonces la salida del controlador es positiva.
8. El error de altitud es positivo y el error de la velocidad es cero, entonces la salida del controlador es positiva.

9. El error de altitud es positivo y el error de la velocidad es positiva, entonces la salida del controlador es positiva.

Con estas reglas lingüísticas, se procede a ingresar en el Toolbox Fuzzy - Matlab®.

3.3. Ajuste y sintonización del controlador FUZZY - PID

Con las reglas lógicas descritas con anterioridad para el controlador de altura de un cuadricóptero, procederemos a utilizar la herramienta de Matlab® - Fuzzy Logic Designer, para ingresar las variables lingüísticas y poder probar el controlador Fuzzy - PID.

Se definen dos variables de entrada necesarias para el controlador Fuzzy - PID; el error de la posición, el error de la velocidad además se define la variable de salida del controlador.

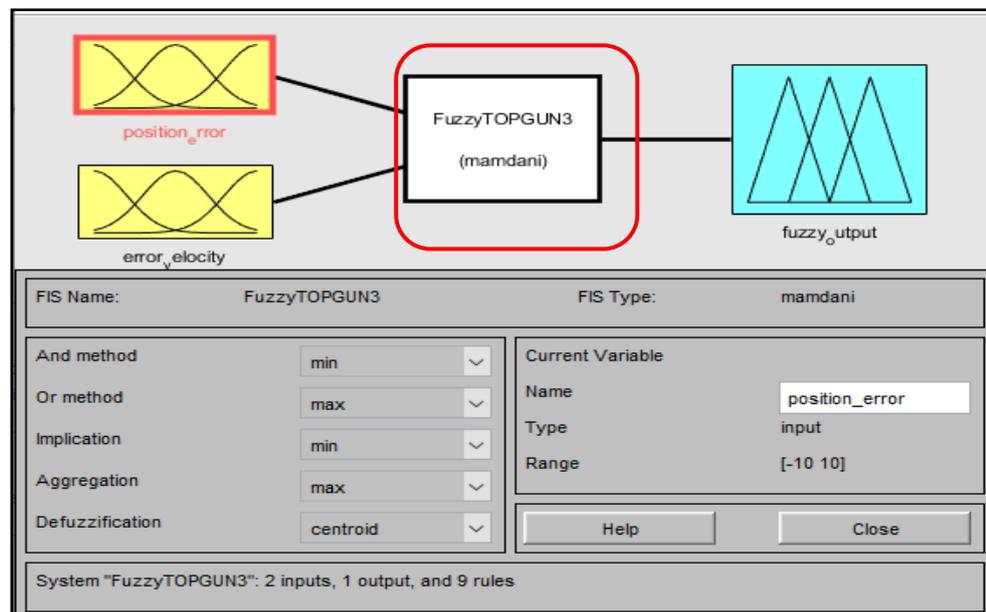


Figura 3.18. Herramienta Fuzzy Logic Designer - Matlab®.

Donde utilizando el método de Inferencia de Mamdani ingresamos las 9 reglas lógicas de las variables lingüísticas.

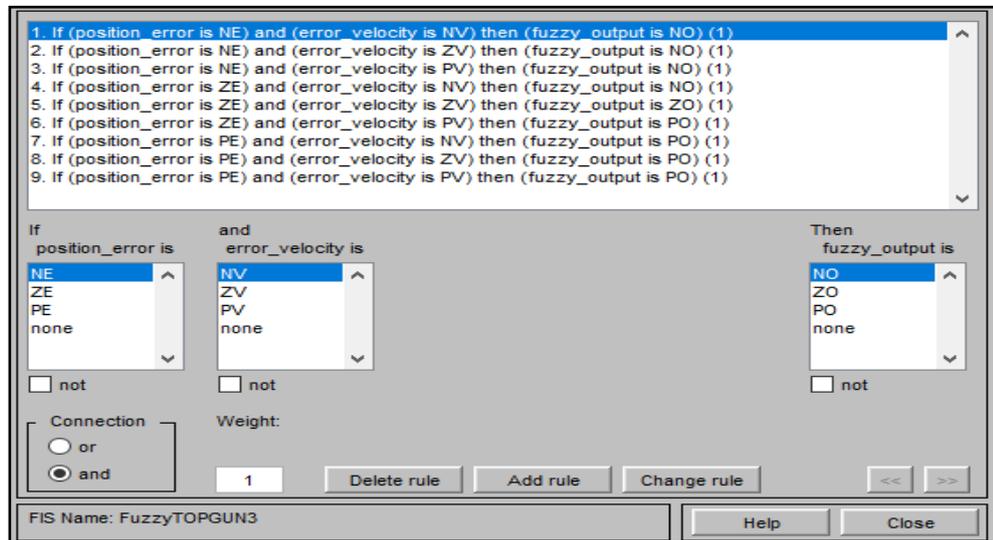


Figura 3.19 Reglas lógicas Controlador Fuzzy de altura para un cuadricóptero.

Para la entrada del error de altitud y error de la velocidad los valores de pertenencia lo realizaremos por medio de una función de membresía triangular "trimf" por permitir simplicidad y facilidad de los cálculos.

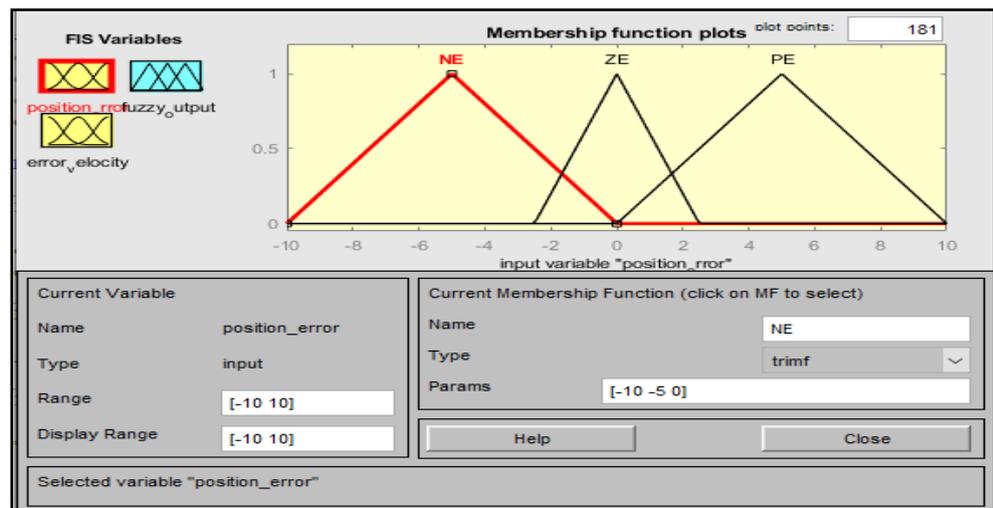


Figura 3.20 Entrada error de altitud, función membresía "trimf".

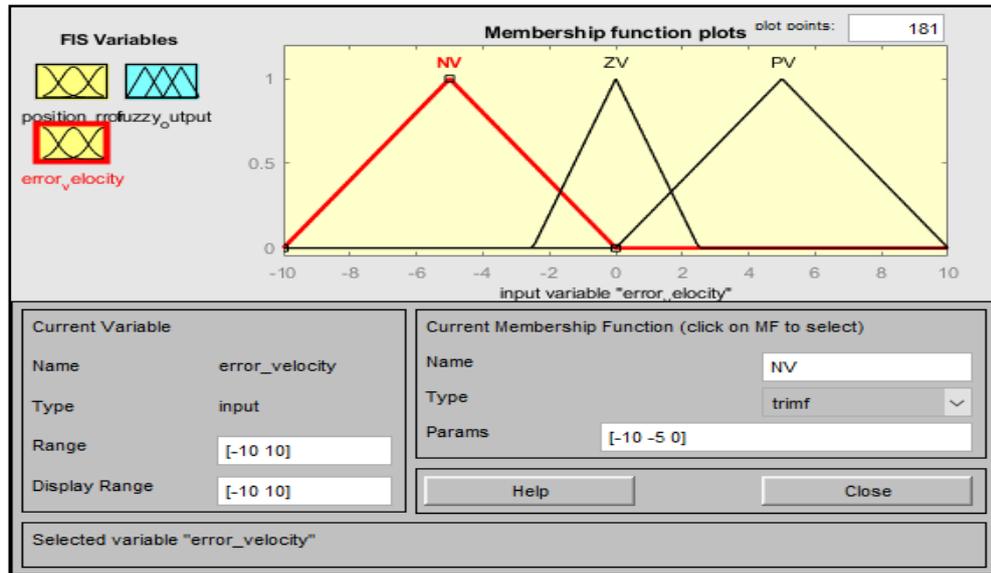


Figura 3.21 Entrada velocidad del error, función membresía "trimf".

Para la salida elegiremos la membresía zmf, la cual nos permitirá evaluar la salida por medio de los valores de las reglas lógicas. En la tabla 9 se puede validar la respuesta rápida del controlador, en función de sus entradas.

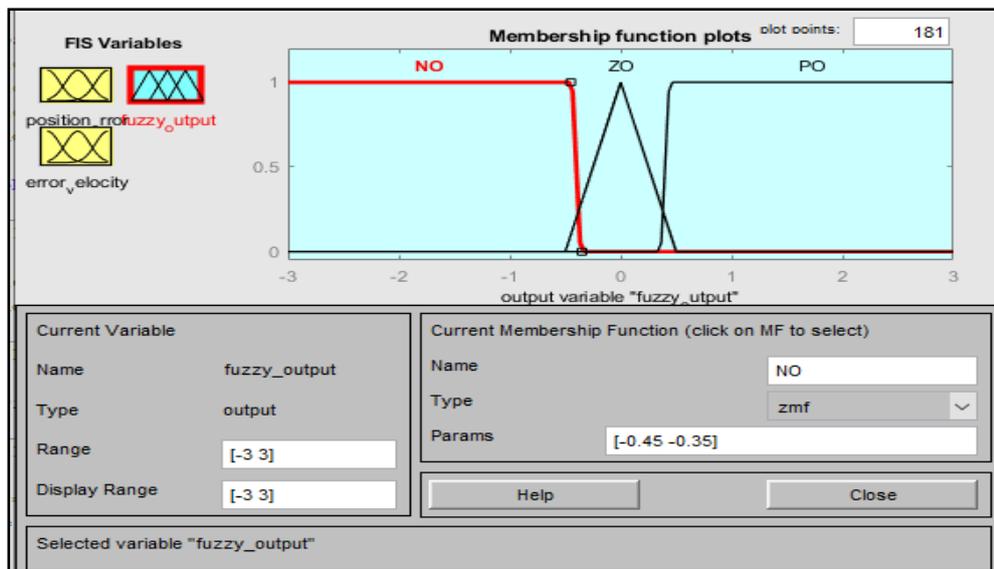


Figura 3.22 Salida de controlador, función membresía "zmf".

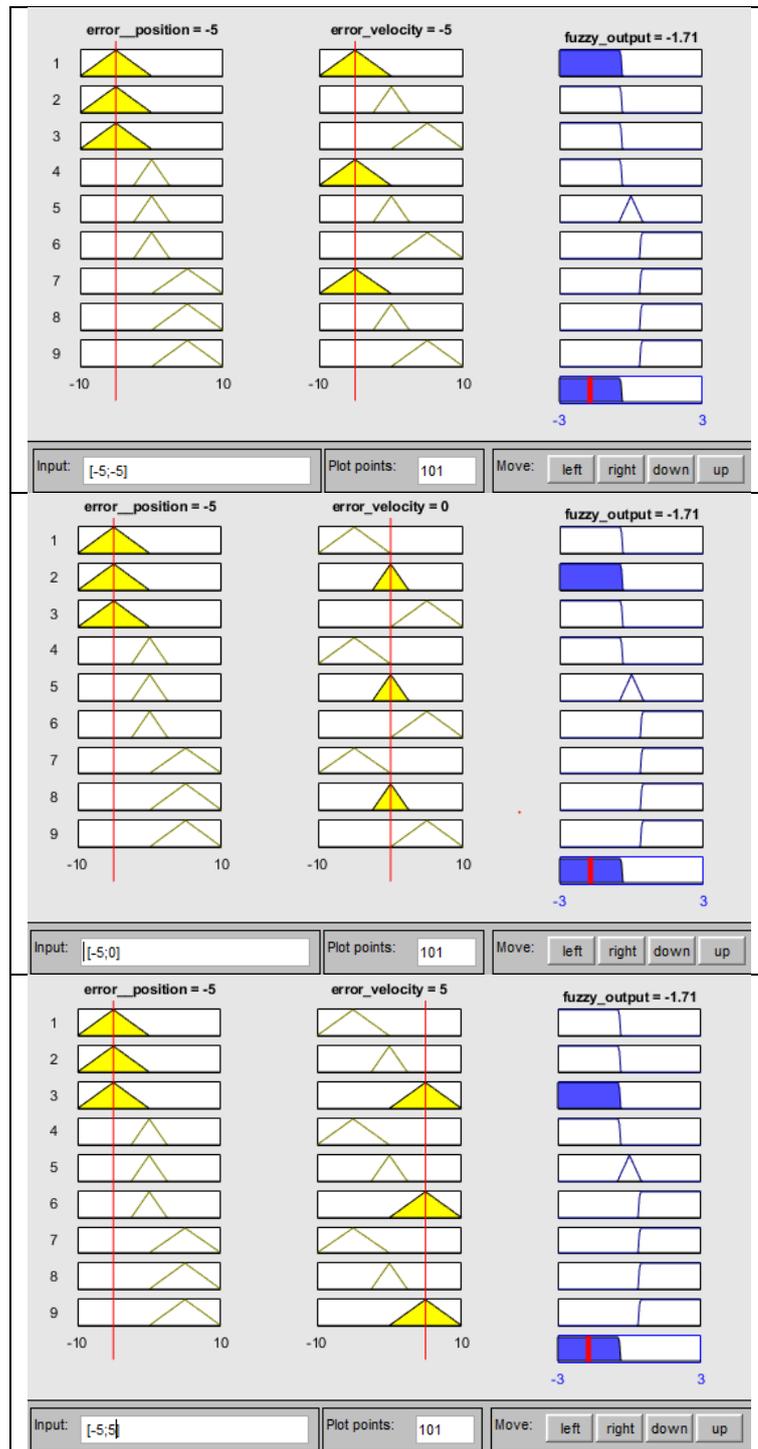


Tabla 9: Respuesta de reglas lógicas difusas con posición del error en -5 y variación de error de la velocidad.

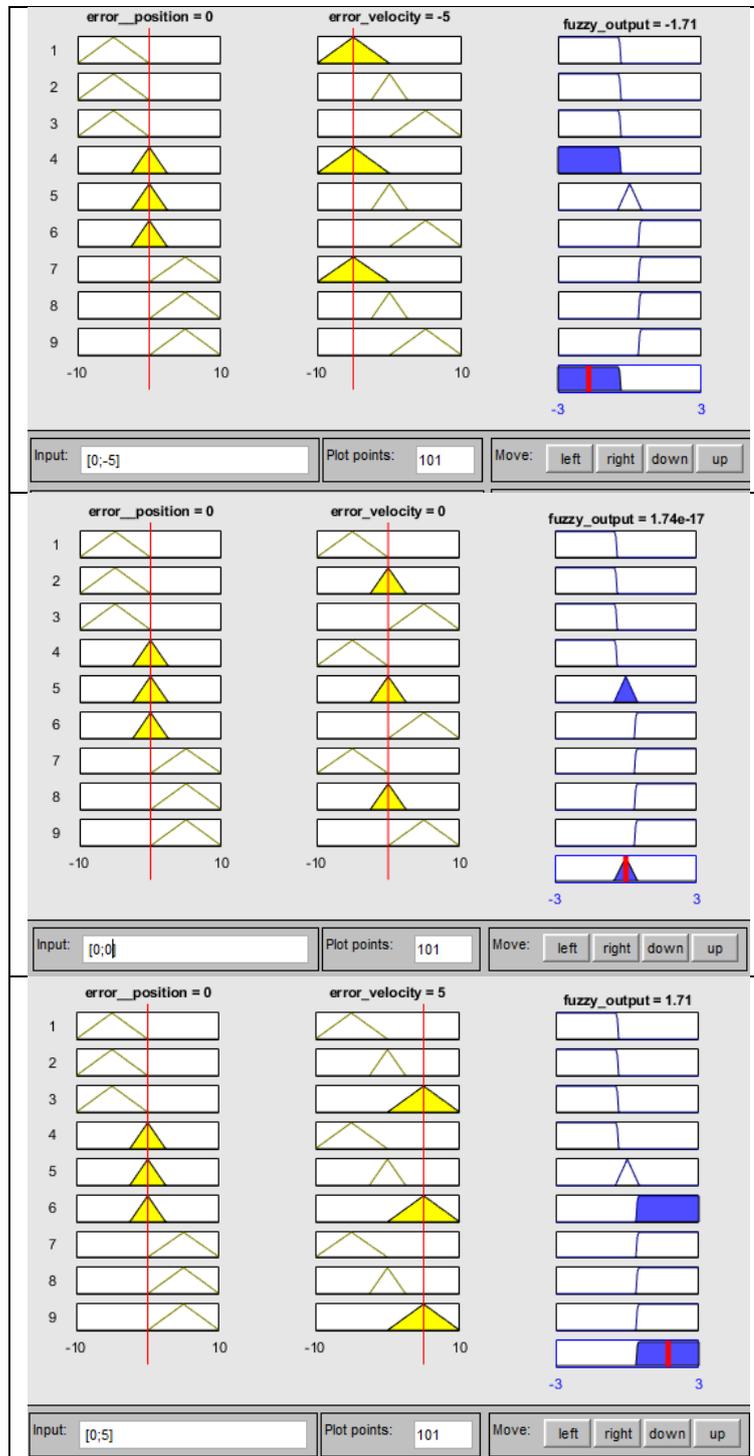


Tabla 10: Respuesta de reglas lógica difusa con posición del error en 0 y variación de error de la velocidad.

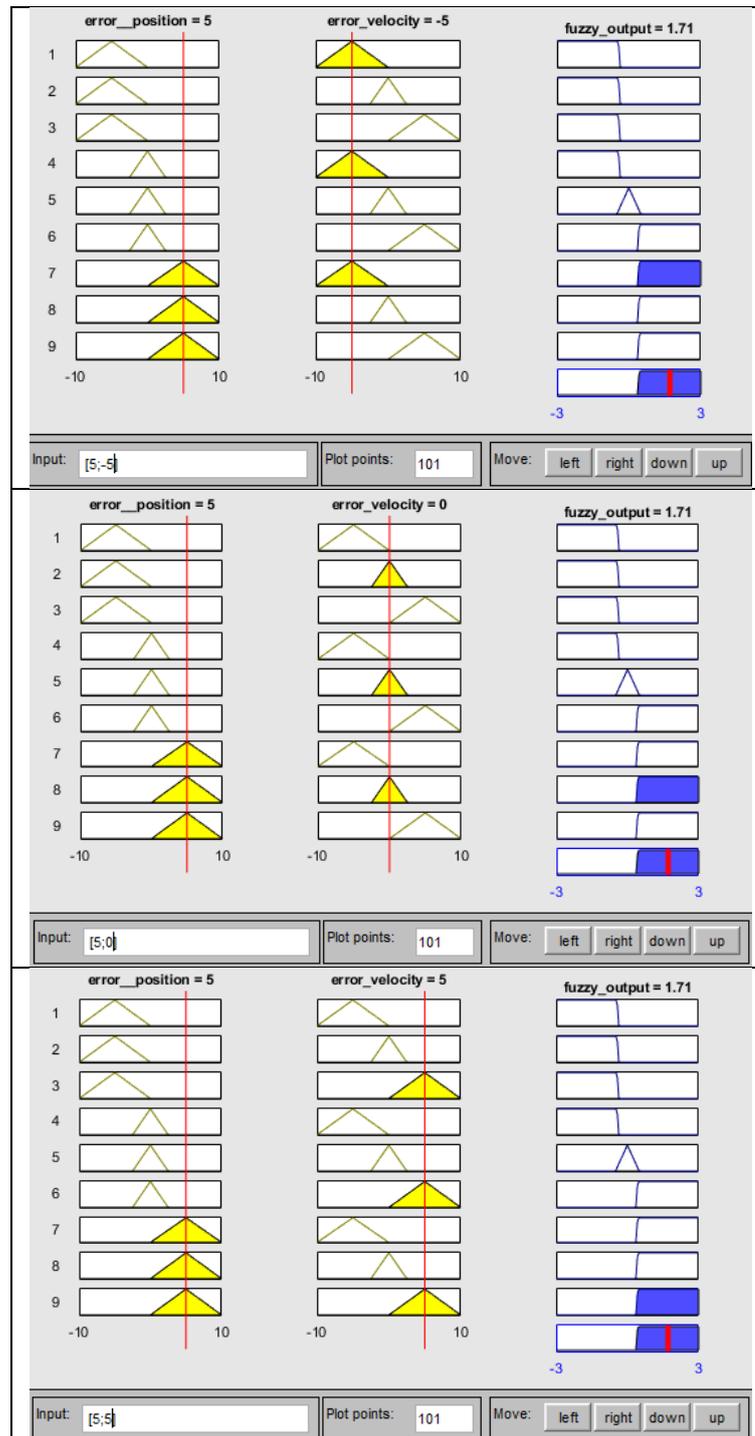


Tabla 11: Respuesta de reglas lógica difusa con posición del error en 5 y variación de error de la velocidad.

Con el controlador Fuzzy obtenido, lo integramos el proceso de control del cuadricóptero para analizar la respuesta de la salida. en la plataforma Simulink®, como se muestra en la siguiente figura 3.17.

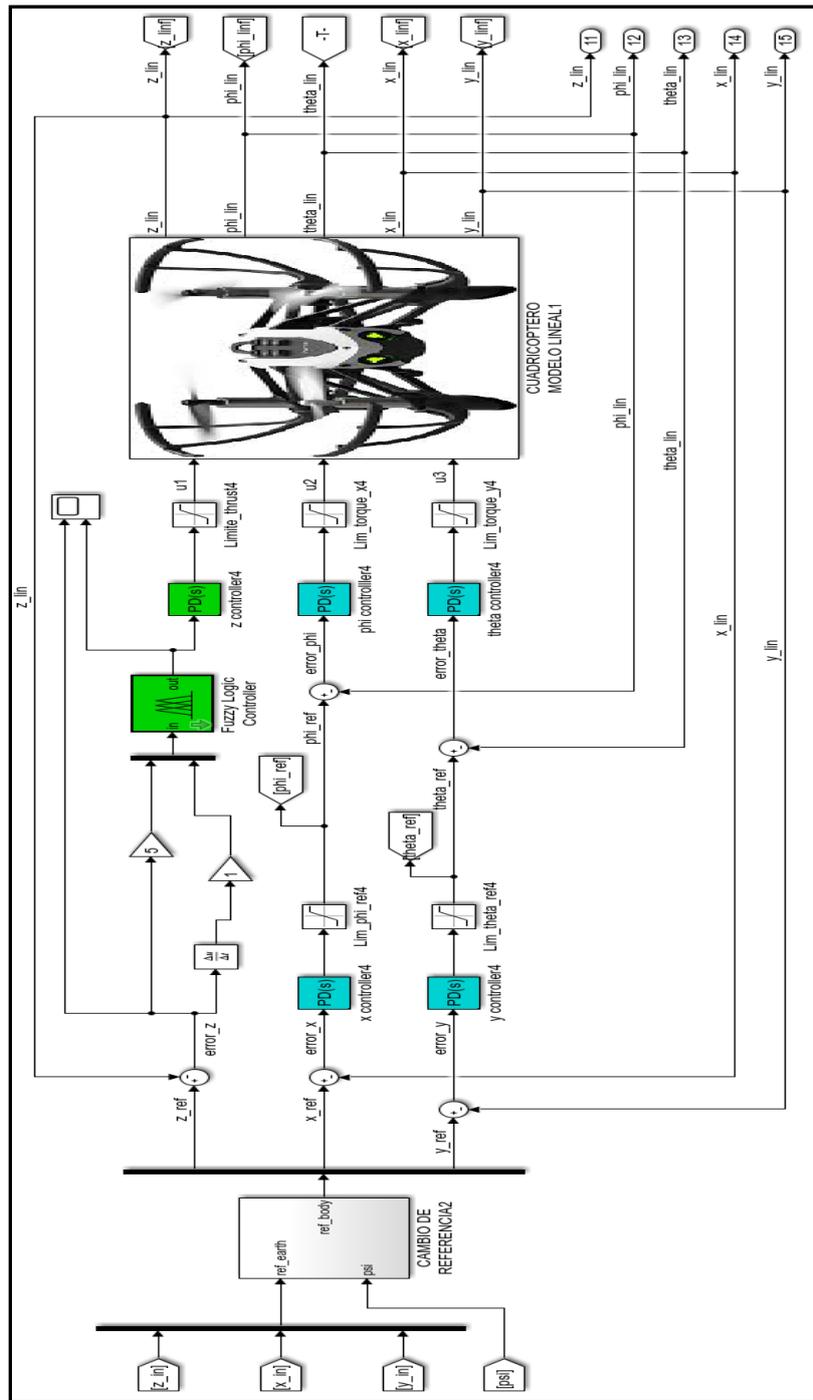


Figura 3.23. Simulación de Controlador FUZZY - PID de un Cuadricóptero en Simulink.

CAPÍTULO 4

ANÁLISIS COMPARATIVO DEL DESEMPEÑO DE LOS CONTROLADORES

4.1 Descripción de los escenarios de pruebas.

Una vez que fueron diseñados los dos tipos de controladores, hicimos un análisis comparativo en base a los indicadores de desempeño. Para obtener el valor de los indicadores de desempeño, se implementaron dos escenarios de pruebas utilizando la herramienta de simulación Matlab-Simulink®. Con estas simulaciones pudimos identificar cuál de ellos presentaba un mejor rendimiento para el control de altitud en el cuadricóptero.

En el primer escenario, utilizamos el modelo no lineal aproximado del cuadricóptero para realizar la implementación del controlador PID de altitud sobre el eje Z y los controladores para los ángulos de posición (ϕ y θ) que determinan el desplazamiento del cuadricóptero sobre los ejes X-Y.

En el segundo escenario, utilizamos el modelo no lineal aproximado del cuadricóptero para realizar la implementación del controlador Fuzzy-PID para el control de altitud, manteniendo los controladores determinados para los ángulos de posición.

Es importante anotar que definimos usar controladores tipo PD para los ángulos de posición (ϕ y θ), porque fueron los que nos brindaron mejor tiempo de respuesta ante los cambios de referencia.

4.2 Resultados de la simulación.

En esta sección, se muestran todos los resultados obtenidos bajo el primer escenario de pruebas, con respecto a un cambio de referencia tipo escalón unitario.

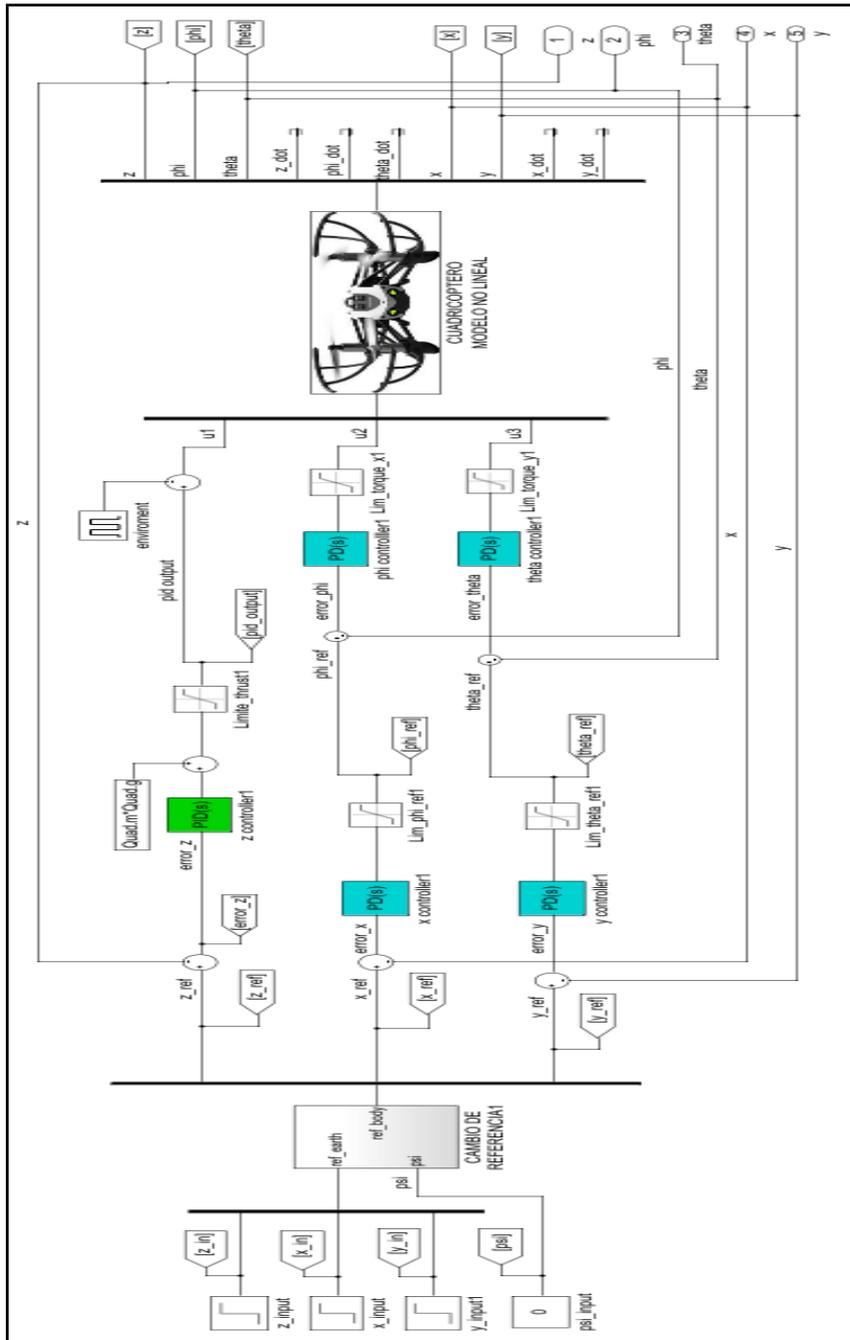


Figura 4.1. Modelo no lineal de cuadricóptero con controlador PID

Como resultado de esta simulación obtuvimos la gráfica donde se muestra la curva de referencia y la curva de respuesta del controlador PID clásico para regular la altitud del cuadricóptero.

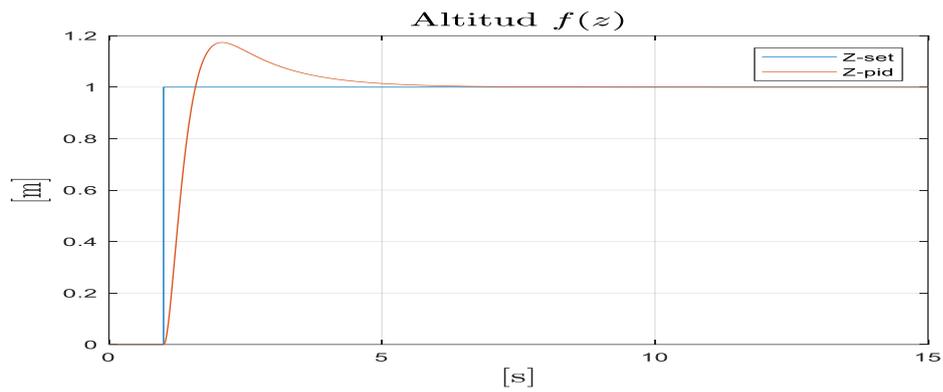


Figura 4.2. Respuesta de controlador de altitud tipo PID.

Observamos que el cambio de referencia ocurre en $t=1s$, produciendo una respuesta subamortiguada, con un sobrenivel porcentual menor al 20% y un tiempo de estabilización menor a 4 segundos.

Además se pudo ver como resultado la señal del error y salida del controlador PID.

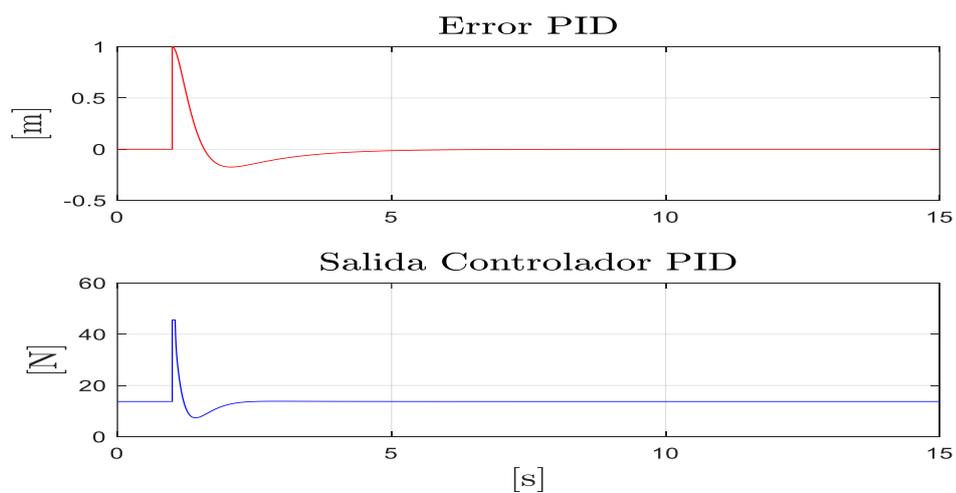


Figura 4.3. Señal de error y salida del controlador de altitud PID.

La salida del controlador alcanza su pico máximo en el momento que se produce el cambio de referencia y empieza a estabilizarse, hasta volver al nivel necesario de empuje para mantener en equilibrio al cuadricóptero en su posición de suspensión en el aire.

Por otra parte, la rápida respuesta de la salida del controlador provoca la disminución del error que alcanza el valor de cero en un corto periodo de tiempo.

Como resultado final de este primer escenario, observamos los movimientos del cuadricóptero representado por las curvas de desplazamiento.

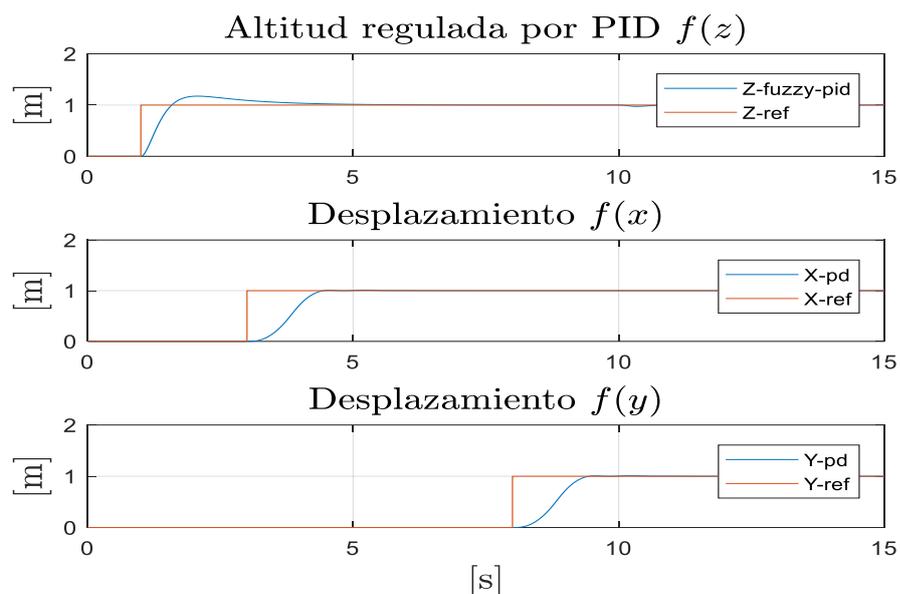


Figura 4.4. Altitud y desplazamientos con controlador PID.

La referencia de altitud sobre el eje Z cambia en $t=1$ s. y se observa la respuesta del controlador de altitud (C_z) ante este cambio. La referencia de desplazamiento sobre el eje X cambia en $t=3$ s. y se observa la respuesta de su controlador (C_x). Finalmente,

la referencia de desplazamiento sobre el eje Y cambia en $t=8s$. y se observa la respuesta de su controlador (C_y).

Obtuvimos el posicionamiento del cuadricóptero resultado del movimiento angular.

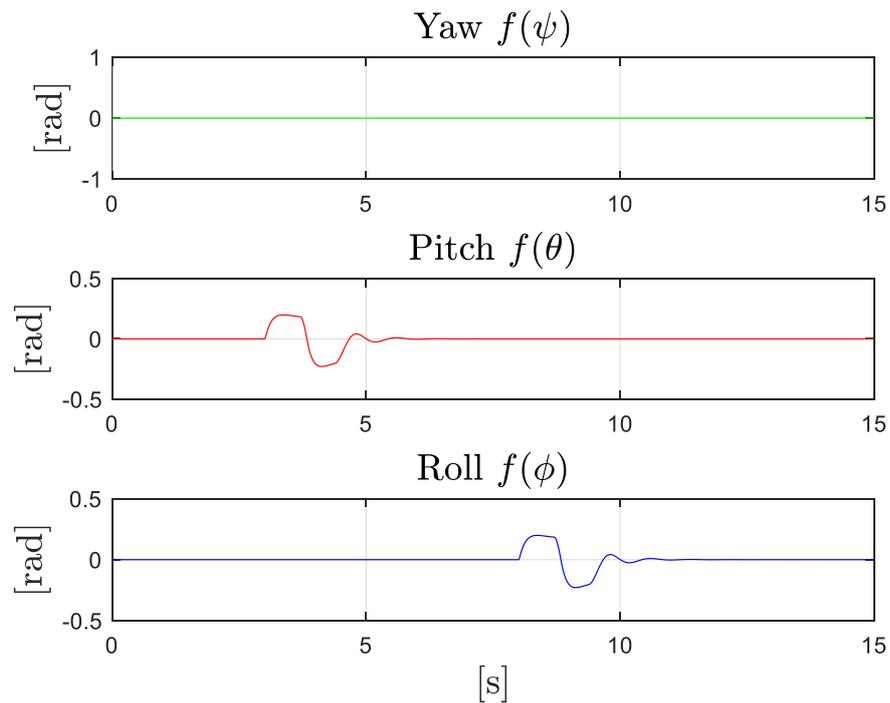


Figura 4.5. Movimiento angular con controladores PD.

En la figura 4.5 se muestra que no existe movimiento rotacional alrededor del eje Z, por lo que no se crea el movimiento de Yaw. El movimiento Pitch empieza en el $t=3s$. cuando cambia la referencia sobre el eje X y termina antes de $t=6s$. Mientras que el movimiento de Roll inicia en $t=8s$. cuando cambia la referencia sobre el eje Y y finaliza antes de $t=11s$.

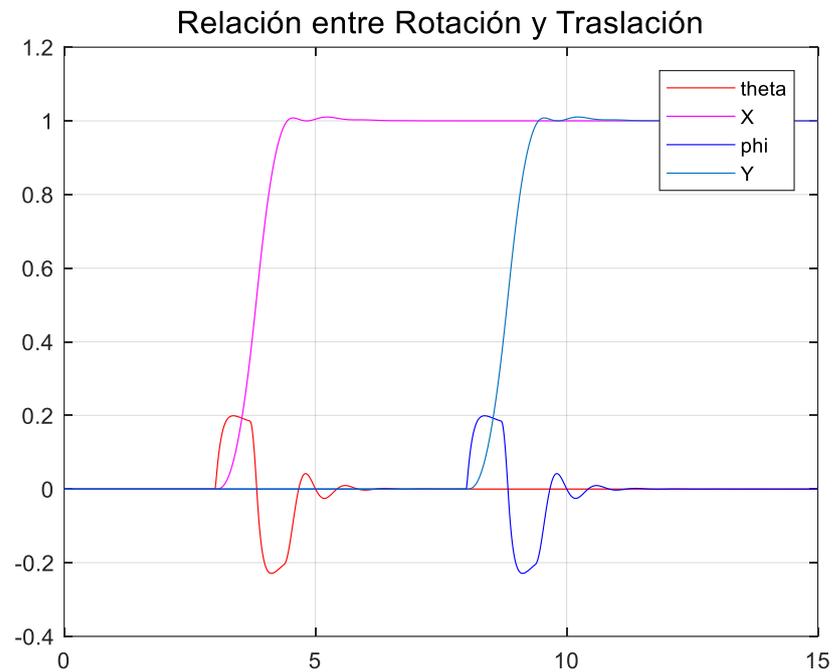


Figura 4.6. Movimiento rotacional y traslacional del cuadricóptero.

La figura 4.6 nos permite mostrar la relación que existe entre el movimiento Pitch que empieza en $t=3s$. el cual provoca el desplazamiento sobre el eje X. También se muestra que el movimiento Roll que empieza en $t=8s$. provoca el desplazamiento sobre el eje Y.

Como resultado de esta simulación obtenemos la gráfica donde se muestra la curva de referencia y la curva de respuesta del controlador Fuzzy-PID.

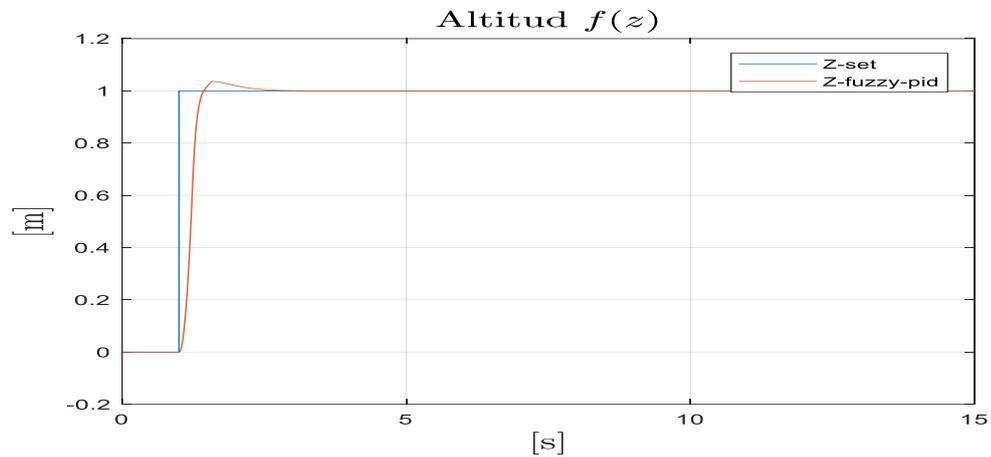


Figura 4.8. Respuesta de controlador de altitud tipo Fuzzy-PID.

Observamos que el cambio de referencia ocurre en $t=1s$, produciendo una respuesta subamortiguada, con un sobrenivel porcentual menor al 4% y un tiempo de estabilización menor a 2 segundos.

Hemos obtenido la gráfica de la señal del error y la salida del controlador PID.

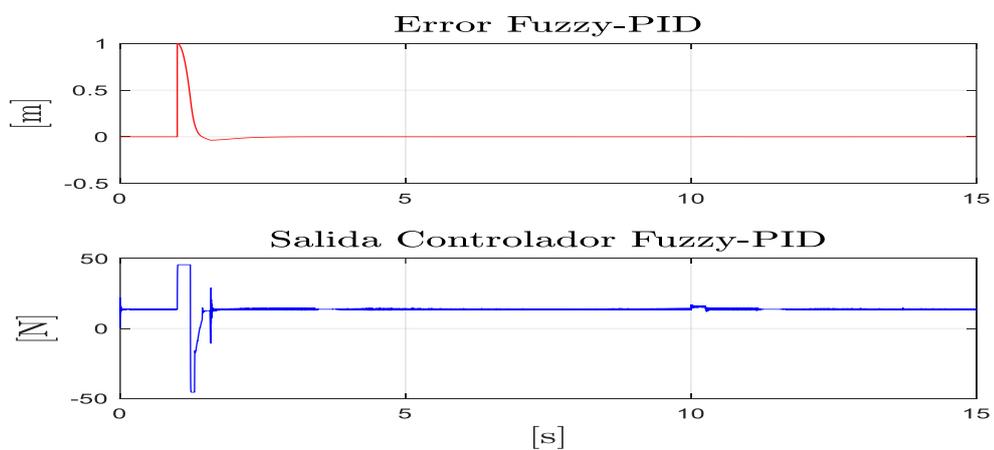


Figura 4.9. Señal de error y salida del controlador Fuzzy-PID.

La salida del controlador alcanza su pico máximo en el momento que se produce el cambio de referencia y empieza a estabilizarse en corto tiempo, hasta volver al nivel necesario de empuje para mantener en equilibrio al cuadricóptero en su posición de suspensión en el aire.

Por otra parte, la rápida respuesta de la salida del controlador provoca la disminución casi inmediata de la señal del error que alcanza el valor de cero en un breve periodo de tiempo.

Como resultado final de este escenario, podemos observar los movimientos del cuadricóptero representado por las curvas de desplazamiento,

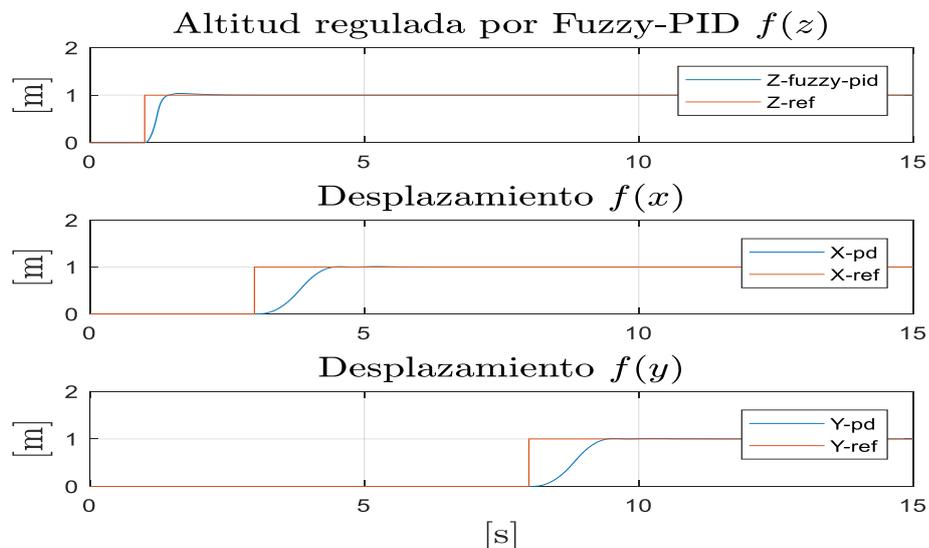


Figura 4.10. Altitud y desplazamientos con controlador Fuzzy-PID.

La referencia de altitud sobre el eje Z cambia en $t=1s$. y se observa la respuesta del controlador de altitud (C_z) ante este cambio. La referencia de desplazamiento sobre el eje X cambia en $t=3s$. y se observa la respuesta de su controlador (C_x). Finalmente,

la referencia de desplazamiento sobre el eje Y cambia en $t=8s$. y se observa la respuesta de su controlador (C_y).

Vemos el posicionamiento del cuadricóptero representado por el movimiento angular.

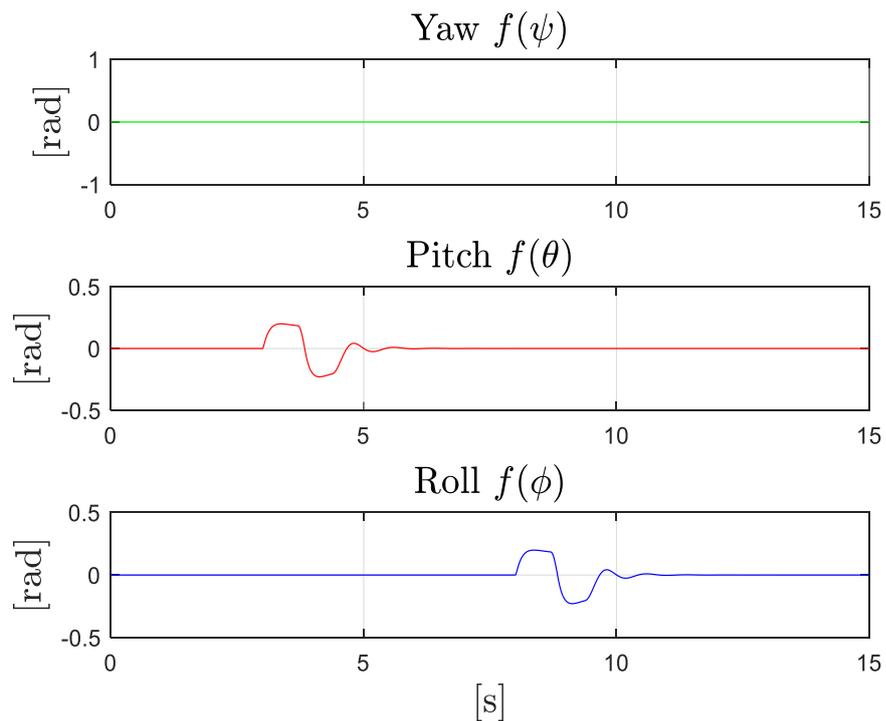


Figura 4.11. Movimiento angular con controladores PD.

En la figura 4.11 se muestra que no existe movimiento rotacional alrededor del eje Z, por lo que no se crea el movimiento de Yaw. El movimiento Pitch empieza en el $t=3s$. cuando cambia la referencia sobre el eje X y termina antes de $t=6s$. Mientras que el movimiento de Roll inicia en $t=8s$. cuando cambia la referencia sobre el eje Y y finaliza antes de $t=11s$.

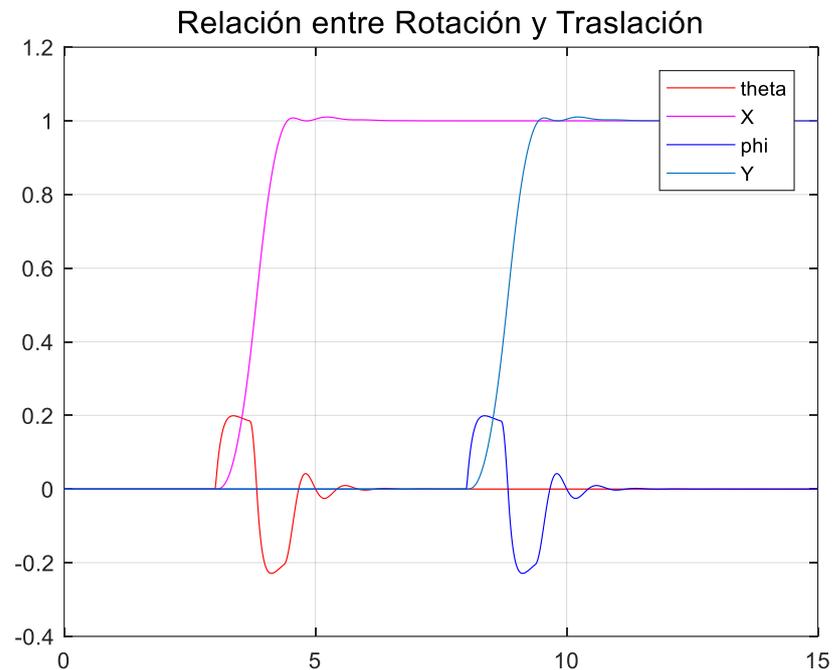


Figura 4.12. Movimiento rotacional y traslacional del cuadricóptero.

La figura 4.12 nos permite mostrar la relación que existe entre el movimiento Pitch que empieza en $t=3s$. el cual provoca el desplazamiento sobre el eje X. También se muestra que el movimiento Roll que empieza en $t=8s$. provoca el desplazamiento sobre el eje Y.

4.3 Análisis de resultados

En esta sección, analizaremos las curvas de respuesta y procederemos a comparar el rendimiento de los controladores diseñados, para lo cual vamos a utilizar los siguientes índices de desempeño: Tiempo de Levantamiento (s), Sobrenivel Porcentual (%), Tiempo de Estabilización (s), Respuesta del controlador (%) y métricas relacionadas al comportamiento del error.

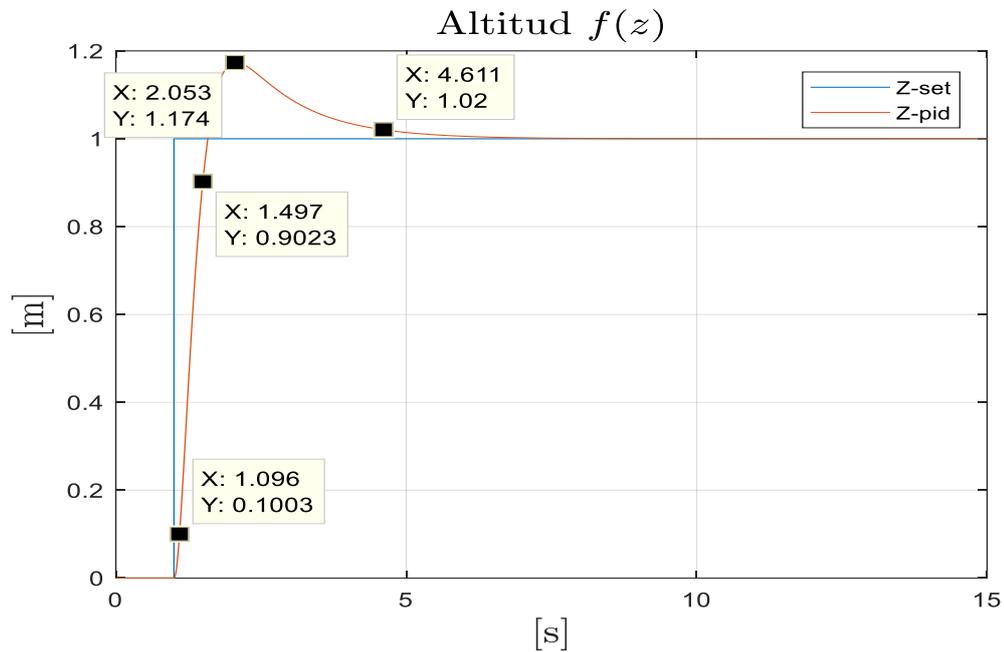


Figura 4.13. Respuesta de altitud de cuadricóptero con controlador PID.

Como se puede observar en la gráfica 4.13 con el controlador PID pudimos obtener los siguientes índices de desempeño: el tiempo de levantamiento es de 0.40 segundos, el sobrenivel porcentual es de 17.4 % y el tiempo de estabilización es de 3.61 segundos.

Al observar la salida del controlador mostrada en la figura 4.14, vemos que mientras el cuadricóptero se encuentra en estado estable la salida del controlador se mantiene en 13.73 N y cuando se presenta el cambio de referencia esta salida se eleva hasta 45.62 N que es el valor máximo de empuje permitido definido por las características de los rotores del cuadricóptero.

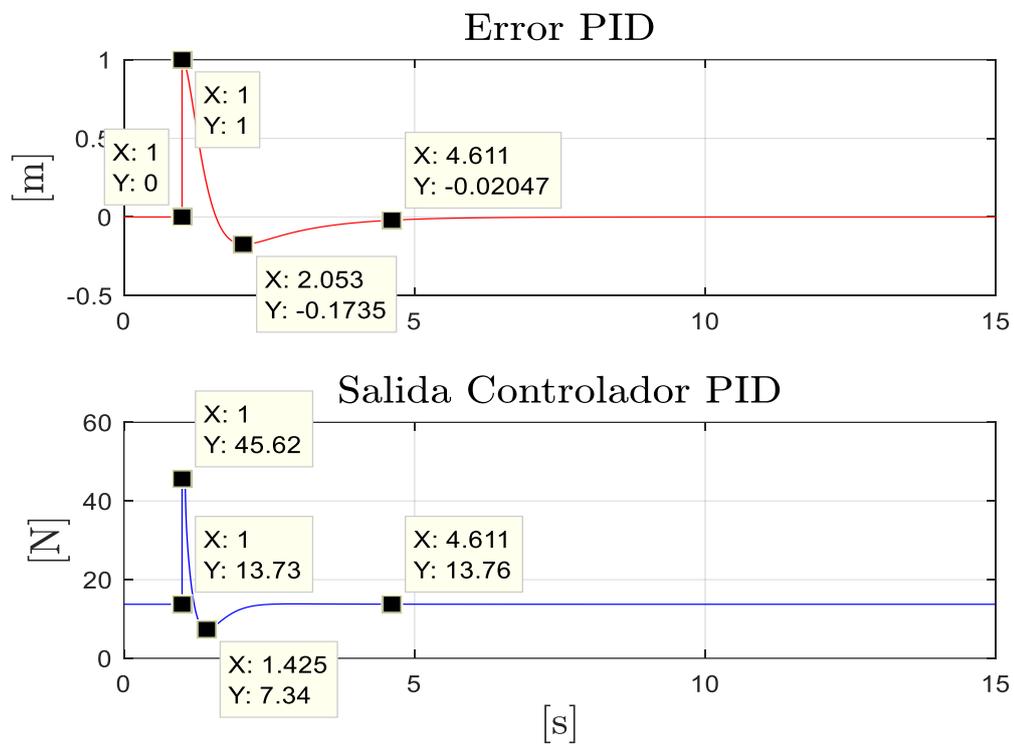


Figura 4.14. Señales de error y salida del controlador PID.

Finalmente, la salida del controlador PID se establece en el valor de 13.76 N cuando el cuadricóptero alcanza nuevamente su estado estable. Esto significa que se mantiene flotando en el aire con esa fuerza resultante por el accionamiento de sus hélices.

En la figura 4.15 mostramos los índices de desempeño del controlador Fuzzy-PID los cuales dieron como resultado: el tiempo de levantamiento es de 0.23 segundos, el sobrenivel porcentual es de 3.60 % y el tiempo de estabilización es de aproximadamente 1 segundo.

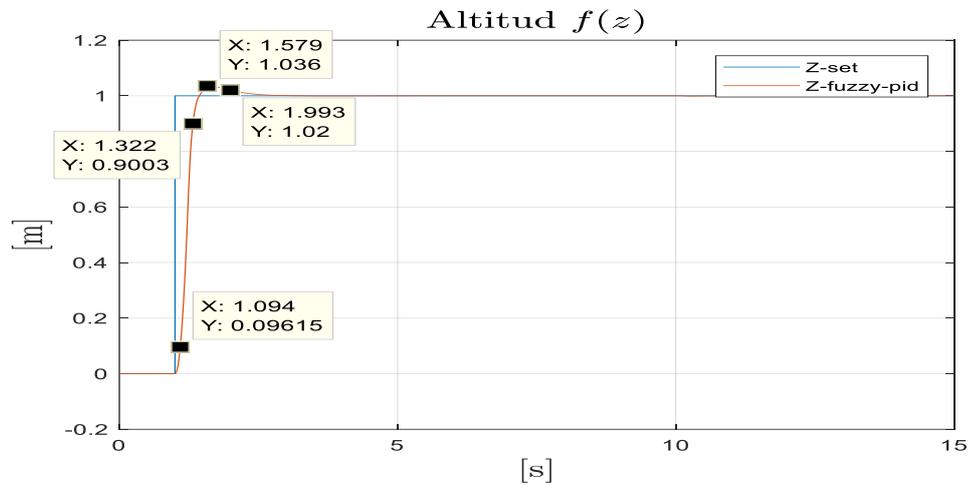


Figura 4.15. Respuesta de altitud de cuadricóptero con controlador Fuzzy-PID.

En estado estable la salida del controlador se mantiene en 13.70 N y cuando cambia la referencia esta salida se eleva hasta 45.62 N que es el valor máximo de empuje permitido, para evitar colapsar los rotores de las hélices del cuadricóptero.

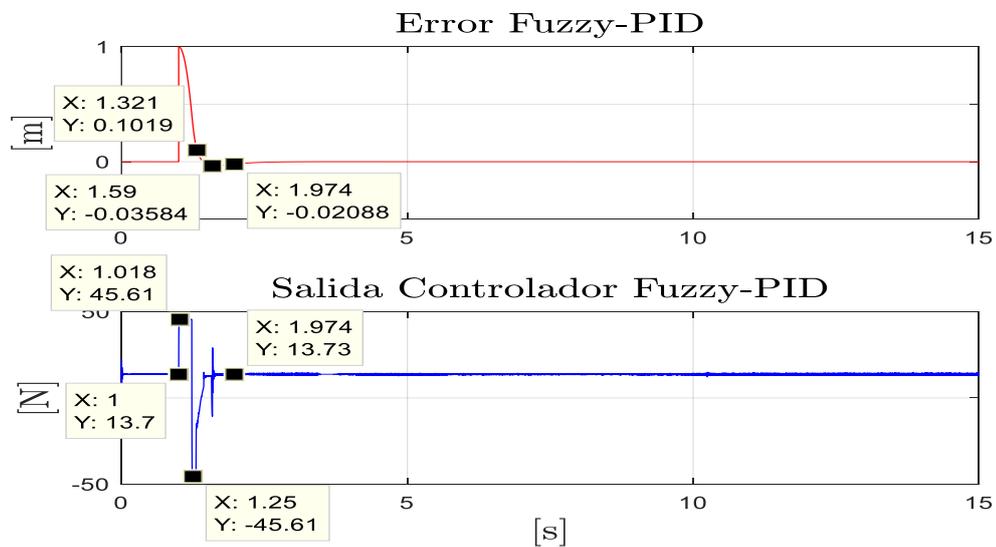


Figura 4.16. Señales de error y salida del controlador Fuzzy-PID.

Es importante anotar que la salida en estado estable del controlador Fuzzy-PID es de tipo oscilatoria debido a la corrección instantánea del error y a su característica no lineal.

Finalmente obtuvimos las gráficas que plantean el resultado más importante de nuestro trabajo. Se muestran las respuestas ante una entrada escalón unitario para el sistema no lineal aproximado de un cuadricóptero con controlador PID clásico y la respuesta con un controlador Fuzzy-PID.

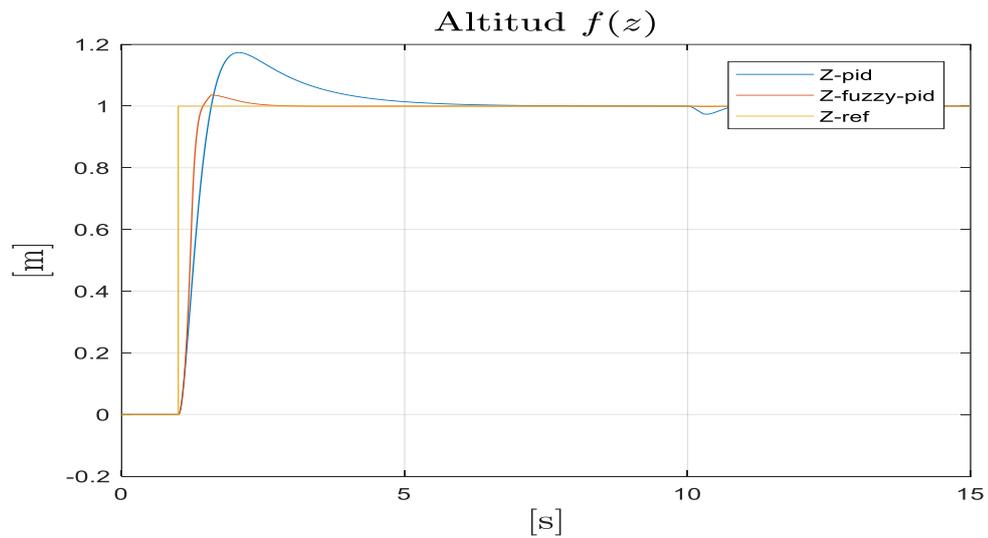


Figura 4.17. Respuesta de controlador PID vs controlador Fuzzy-PID.

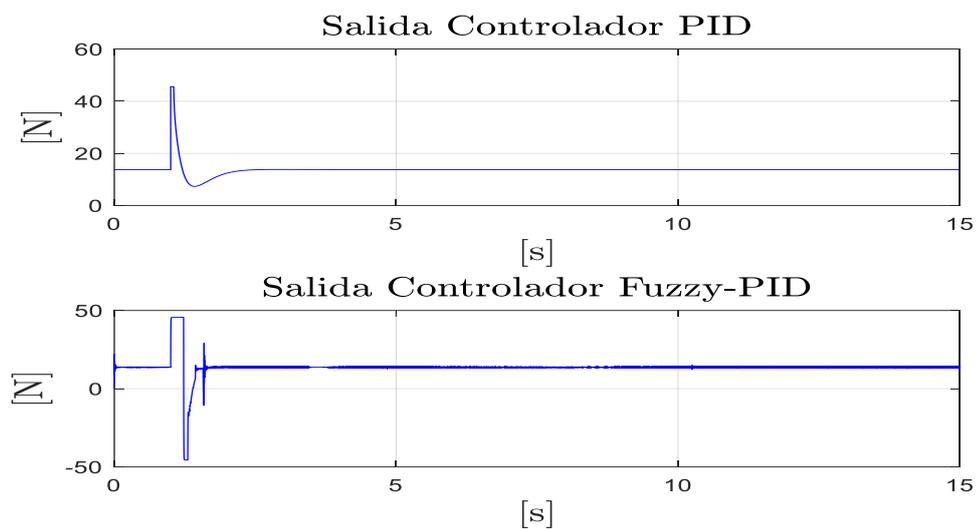


Figura 4.18. Salida de controlador PID vs controlador Fuzzy-PID.

Adicionalmente, hemos utilizado la aplicación Microsoft Excel® para determinar índices relacionados con los errores, estos resultados se muestran en la tabla 10.

Índice de Desempeño	Controlador PID	Controlador Fuzzy-PID
Tiempo de Levantamiento	0.4030 [s]	0.2280 [s]
Sobrenivel Porcentual	17.4 [%]	3.6 [%]
Tiempo de Estabilización	3.6160 [s]	0.9930 [s]
Desviación Estándar	0.0991	0.0716
RMSE	0.9715	0.9558

Tabla 12: Índices de Desempeño de Controladores PID y Fuzzy-PID.

Los datos presentados en la tabla 10 nos han permitido determinar que el controlador de altitud tipo Fuzzy-PID muestra mejor desempeño, ya que tiene una mayor velocidad de respuesta, menor sobrenivel porcentual y menor tiempo de estabilización con respecto al controlador de altitud tipo PID clásico.

4.4 Respuesta ante perturbaciones.

En este primer escenario se muestra un análisis de respuesta de los controladores PID y Fuzzy-PID ante una perturbación aleatoria por lo que se usa el bloque “band-limited white noise” de Simulink® el cual representa la fuerza aleatoria del viento.

En la figura 4.19 se muestra la perturbación utilizada para nuestra simulación la cual consiste en un ruido blanco con una potencia de 0.1, el cual representa la fuerza aleatoria que ejerce el viento sobre el cuadricóptero.

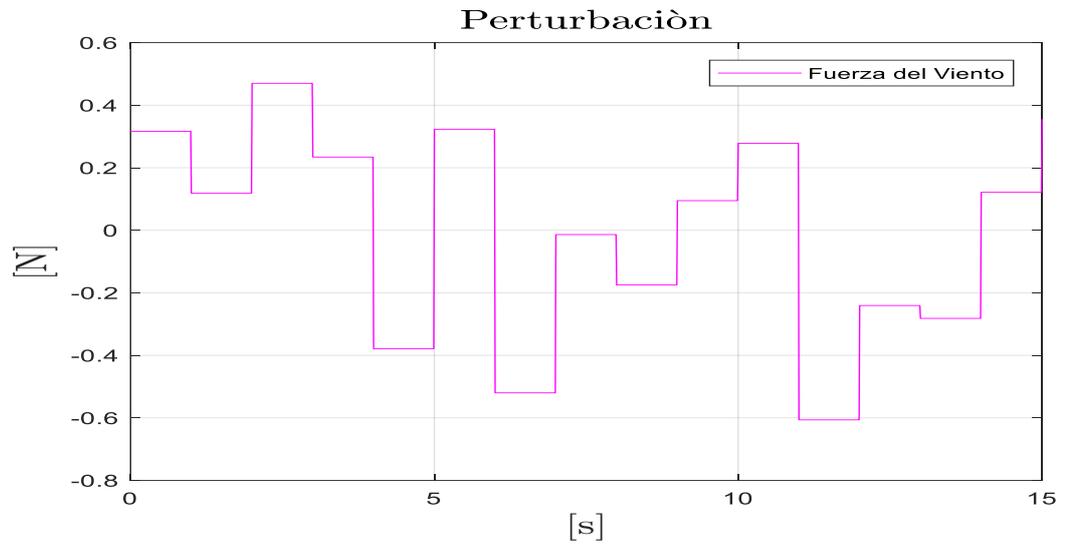


Figura 4.19. Perturbación por efecto de la fuerza del viento.

En la figura 4.20 se observan las respuestas de los controladores de altitud frente a la fuerza aleatoria que ejerce el viento sobre el cuadricóptero.

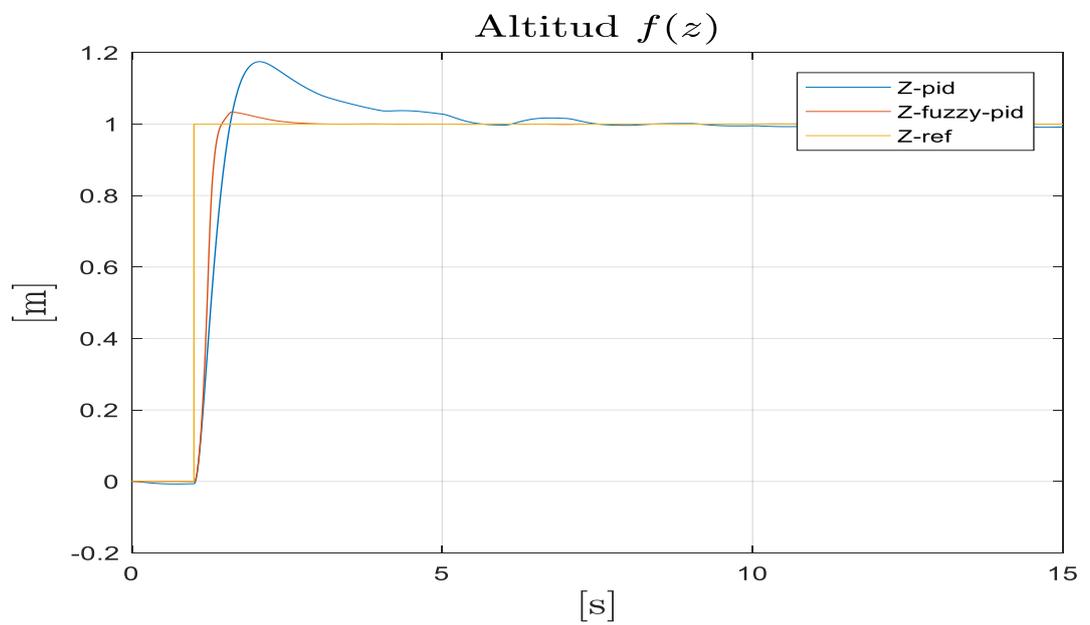


Figura 4.20. Respuesta de controladores de altitud a perturbaciones del viento.

A pesar de la perturbación que afecta al cuadricóptero, ambos controladores responden de manera adecuada. En ambos casos la altitud es regulada hasta alcanzar el valor deseado, mostrando un mejor desempeño el controlador de tipo Fuzzy-PID ya que se ajusta más rápidamente al valor de la referencia.

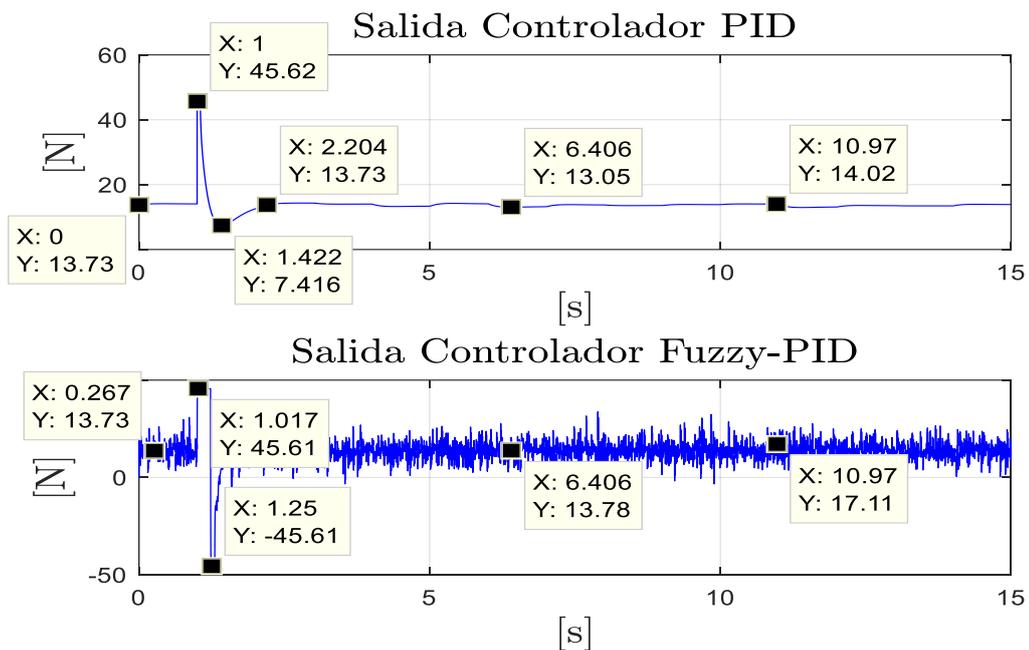


Figura 4.21. Señales de controladores frente a la perturbación del viento.

Cuando analizamos la respuesta de ambos controladores frente al efecto de la perturbación, vemos que el controlador PID clásico mantiene un comportamiento casi regular, con ligeros cambios en su curva de respuesta.

Por otra parte, el controlador Fuzzy-PID presenta una respuesta que se va ajustando de manera instantánea ante la variación de la señal de perturbación lo que provoca que finalmente tenga un mejor desempeño frente a la respuesta del controlador PID.

En el segundo escenario se propuso como perturbación un impulso ejercido por una fuerza externa al cuadricóptero. La curva representativa de este impulso se muestra en la figura 4.22 cuya amplitud es de 2 N. y ocurre a los 10 segundos.

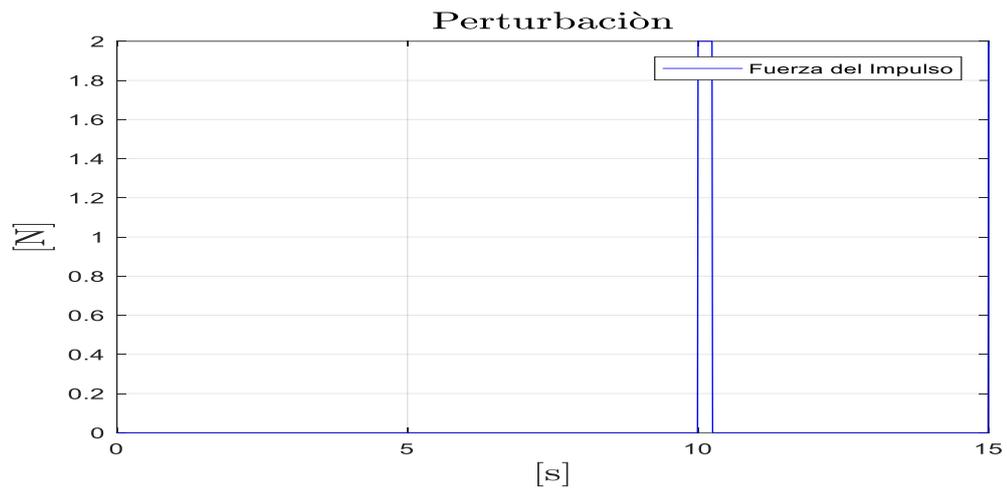


Figura 4.22. Perturbación por efecto de impulso externo.

En la figura 4.23 se observan las respuestas de los controladores de altitud frente a la fuerza de tipo impulso ejercido sobre el cuadricóptero.

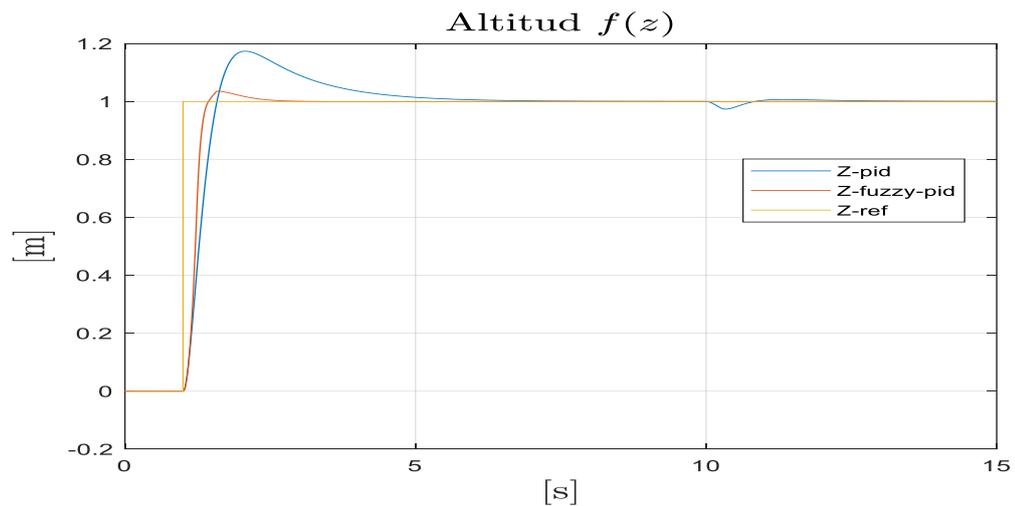


Figura 4.23 Respuesta de controladores de altitud a perturbación impulso.

A pesar de la perturbación que afecta al cuadricóptero, ambos controladores responden de manera adecuada. En ambos casos la altitud es regulada hasta alcanzar el valor deseado, mostrando un mejor desempeño el controlador de tipo Fuzzy-PID ya que se ajusta más rápidamente al valor de la referencia.

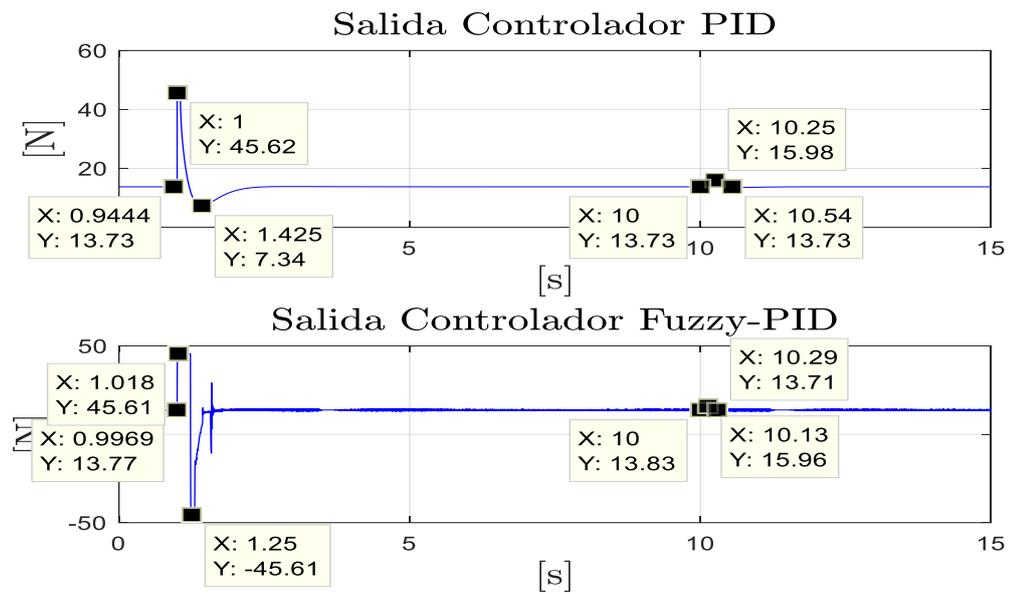


Figura 4.24. Señales de controladores frente a la perturbación impulso.

CONCLUSIONES

- 1.- Es necesario linealizar el sistema para aplicar un controlador tipo lineal, como el PID clásico, en el sistema no lineal del cuadricóptero. Con este objetivo, debemos definir los estados y las entradas iniciales para determinar correctamente el punto de operación del sistema alrededor del cual va a trabajar el controlador.
- 2.- La función de transferencia para el control de altitud sobre el eje Z demuestra que este es un sistema tipo SISO (single input, single output) por lo cual se aplica la herramienta sisotool para el diseño del controlador PID clásico.
- 3.- El controlador de altitud PID diseñado ha logrado cumplir satisfactoriamente los requerimientos, lo cual se evidencia en su correcto desempeño durante las simulaciones realizadas en la etapa de pruebas del sistema de control.
- 4.- Hemos seleccionado para este proyecto el controlador Fuzzy-PID tipo Mamdani debido a que su operación se fundamenta en el uso de las variables lingüísticas, lo cual permite una mayor flexibilidad y mejor comprensión al momento de realizar su configuración y ajuste.
- 5.- De acuerdo a lo que muestran los índices de desempeño obtenidos se demuestra que el controlador Fuzzy-PID tiene un mejor rendimiento ante los cambios de referencia y una mejor respuesta frente a las perturbaciones, en comparación a un controlador PID clásico.

RECOMENDACIONES Y TRABAJOS FUTUROS

- 1.- Se recomienda conocer el principio de funcionamiento del cuadricóptero previo a desarrollar cualquier tipo de sistema de control. Para lograr determinar cuáles serían las respuestas esperadas ante el accionamiento del controlador a ser diseñado.
- 2.- Podemos recomendar como trabajo futuro la realización de un controlador predictivo basado en modelo para regular la altitud y posicionamiento de un cuadricóptero.
- 3.- Implementar los controladores propuestos en el presente trabajo sobre un cuadricóptero real sería un nuevo desafío para contrastar la respuesta real del sistema con respecto a la respuesta simulada que hemos desarrollado.

BIBLIOGRAFÍA

- [1] S. Mustapa, S. Saat, S. H. Husin y N. Abas, «Altitude controller design for multicopter UAV,» de *International Conference on Computer, Communications, and Control Technology (I4CT)*, pp.382-387, Langkawi, 2014.
- [2] A. Astudillo, P. Muñoz, F. Álvarez y E. Rosero, «Altitude and attitude cascade controller for a smartphone-based quadcopter, pp.1447-1454,» de *International Conference on Unmanned Aircraft Systems (ICUAS)*, Miami, FL, USA, 2017.
- [3] D. Domingos, G. Camargo y F. Gomide, «Autonomous Fuzzy Control and Navigation of Quadcopters,» *IFAC-PapersOnLine*,, vol. 49, nº 5, pp. 73-78, 2016.
- [4] L. Alsafadi, N. Chulin y I. Mironova, «Synthesis of Fuzzy Controller Based on Simple PID Controller,» *Procedia Computer Science*, , vol. 150, nº ISSN1877-0509, pp. 28-38, 2019.
- [5] Hernández Millán, Germán, Ríos Gonzales, Luís Hernando, & Bueno López, Maximiliano. (2016). Implementación de un controlador de posición y movimiento de un robot móvil diferencial. *Tecnura*, 20(48), 123-136.
- [6] J. A. Sanchez, *Instrumentación y control avanzado de procesos*, Madrid-España: Diaz de Santos, 2006.

- [7] K. Ogata, *Ingeniería de Control Moderno*, Madrid: Pearson, 2010.
- [8] V. M. Alfaro Ruíz, «Métodos de Sintonización de Controladores PID que Operarán como Reguladores,» *Ingeniería*, vol. 2, nº 12, pp. 21-36., 2002.
- [9] M. & P. A. Kushwah, «Tunning PID Controller for Speed Control of DC Motor Using Soft Computing Techniques - A Review.,» *Advance in electronic an Electric Engineering*, vol. 4, nº 2, pp. 141-148, 2014.
- [10] T. Osuna Altamirano, «Navegación y Control de Robot Móvil.,» de *Centro de Investigación y Desarrollo de Tecnología Digital.*, México, 2010.
- [11] M. A. C. Dueñas, «Localización probabilística en un robot con visión local,» Madrid, 2003.
- [12] R. E. Kalman, «A new approach to linear filtering and prediction problems,» *J. Basic Eng*, vol. 82, nº 1, pp. 33-45, 1960..
- [13] B. D. O. a. M. J. B. Anderson, «Optimal Filtering,» Dover, New York, 2005.
- [14] J. H. Bibby, «Prediction and Improved Estimation in Linear Models,» Wiley, New York:, 1977.
- [15] G. B. G. Welch, «University of North Carolina, An Introduction to the Kalman Filter,» Chapel Hill, USA, 2006.

- [16] Hamdy, Osama & El-Nady, Abdelrady. (2019). Modeling, Simulation and Control of Quadcopter using PID Controller.
- [17] Gonzáles Morcillo, Carlos. Lógica Difusa Una introducción práctica. Técnicas de Softcomputing.
- [18] I. Rubio Scola, G. A. Guijarro Reyes, L. R. Garcia Carrillo, J. Hespanha and J. Xie, «Translational Model Identification and Robust Control for the Parrot Mambo UAS Multicopter,» 2019 IEEE Globecom Workshops (GC Wkshps), Waikoloa, HI, USA, 2019, pp. 1-6, doi:

ANEXOS

PROGRAMA DE MATLAB

```

%% ENCABEZADO
%*****
%           ESCUELA SIPERIOR POLITECNICA DEL LITORAL
%   MAESTRIA EN AUTOMATIZACION Y CONTROL INDUSTRIAL
%                   VI COHORTE

%   TRABAJO DE TITULACIÓN PARA LA OBTENCIÓN DEL TÍTULO
%   DE MAGÍSTER EN AUTOMATIZACIÓN Y CONTROL INDUSTRIAL

% TEMA:
% DISEÑO Y ANALISIS DEL DESEMPEÑO DE CONTROLADORES PID Y
% PID-FUZZY PARA CONTROL DE ALTITUD EN UN CUADRICOPTERO.

% AUTORES:
% ING. CESAR RONALD FLOR ROLDAN
% ING. LUIS ANTONIO REYES RIZO

% DIRECTOR:
% Ph.D. DOUGLAS ANTONIO PLAZA GUINGLA
%*****
%% FUNCIONES Y SUBRUTINAS

addpath subrutinas % Se anade carpeta de funciones y
subrutinas

%% INICIALIZACION WORKSPACE

close all,clear all, clc; % Cierre de Graficos, Limpieza de
Workspace y Command Window
load 'LinsysQuad01.mat' % Carga de variables en Workspace
load 'TrimPointQuad01.mat'
load 'TrimPointQuad01Settings.mat'
load 'QuadWorkspace01.mat'

global Quad; % Definicion de variable global

%% INICIALIZACION GRAFICOS

inicio_plot; % Grafica el espacio tridimensional con el punto
de referencia
quad_3D; % Grafica del cuadricoptero en el espacio
tridimensional

```

```

%% PARAMETROS Y VARIABLES DEL SISTEMA

variables; % Sección para almacenar variables del
cuadricoptero

%% INICIALIZACION DEL MODELO

%x0=[z,phi,theta,x,y,z_dot,phi_dot,theta_dot,x_dot,y_dot]
x0=[1; % z
    1.25e-2; % phi
    1.25e-2; % theta
    1; % x
    1; % y
    0.2; % z_dot
    2.5e-3; % phi_dot
    2.5e-3; % theta_dot
    0.4; % x_dot
    0.4]; % y_dot

u0=[(Quad.m*Quad.g)+0.028; %trust
    0.0000125; % torque_x
    0.0000125]; % torque_y

%% REPRESENTACION DEL SISTEMA EN ESPACIO DE ESTADOS

A=linsys1.A; % Matriz Amxm; m = # de estados del sistema
B=linsys1.B; % Matriz Bmxn; n = # de entradas del sistema
C=linsys1.C; % Matriz Cnxm; n = # de salidas del sistema
D=linsys1.D; % Matriz Dmxn; m = # salidas y n = # entradas del
sistema
sys=ss(A,B,C,D); % Representacion en Espacio de Estados
Gs=tf(sys); % Funcion de Transferencia del Sistema

Gz=zpk(Gs(1,1)); % Funcion de Transferencia de Thrust vs Z
Gphi=zpk(Gs(2,2)); % Funcion de Transferencia de Torque_X vs
Phi
Gtheta=zpk(Gs(3,3)); % Funcion de Transferencia de Torque_Y vs
Theta
Gx=tf([Quad.g],[1 0 0]); % Funcion de Transferencia de
Torque_X vs X
Gy=tf([Quad.g],[1 0 0]); % Funcion de Transferencia de
Torque_Y vs Y

```

```

%% DISEÑO DE LOS CONTROLADORES

% Requisitos del Sistem
% Mp=30% Sobrenivel Porcentual
% Tiempo de Estabilizacion = 3 seg

% sisotool Gz
Cz; % Funcion de Transferencia del controlador de Altitud
Kpz=pid(Cz).Kp; % Constante Kp del controlador Cz
Kiz=pid(Cz).Ki; % Constante Ki del controlador Cz
Kdz=pid(Cz).Kd; % Constante Kd del controlador Cz

% sisotool Gphi
Cthetaphi; % Funcion de Transferencia del controlador angular
en X
Kpphi=pid(Cthetaphi).Kp; % Constante Kp del controlador
Cthetaphi
Kiphi=pid(Cthetaphi).Ki; % Constante Ki del controlador
Cthetaphi
Kdphi=pid(Cthetaphi).Kd; % Constante Kd del controlador
Cthetaphi

% sisotool Gx
Cxy; % Funcion de Transferencia del controlador de posición en
X
Kpx=pid(Cxy).Kp; % Constante Kp del controlador Cxy
Kix=pid(Cxy).Ki; % Constante Ki del controlador Cxy
Kdx=pid(Cxy).Kd; % Constante Kd del controlador Cxy

% sisotool Gtheta = Gphi
Cthetaphi; % Funcion de Transferencia del controlador angular
en Y
Kptheta=pid(Cthetaphi).Kp; % Constante Kp del controlador
Cthetaphi
Kitheta=pid(Cthetaphi).Ki; % Constante Ki del controlador
Cthetaphi
Kdtheta=pid(Cthetaphi).Kd; % Constante Kd del controlador
Cthetaphi

% sisotool Gy = Gx
Cxy; % Funcion de Transferencia del controlador de posición en
Y
Kpy=pid(Cxy).Kp; % Constante Kp del controlador Cxy
Kiy=pid(Cxy).Ki; % Constante Ki del controlador Cxy
Kdy=pid(Cxy).Kd; % Constante Kd del controlador Cxy

```

```

%% SIMULACION

TopGun = sim('TOPGUN_Simulink2');

for S = 1:1:size(TopGun,1)

    Quad.X = X_out.signals.values(S);
    Quad.Y = Y_out.signals.values(S);
    Quad.Z = -Z_fuz.signals.values(S);
    Quad.phi = Phi_out.signals.values(S);
    Quad.theta = Theta_out.signals.values(S);
    Quad.psi = Psi_out.signals.values(S);

    plot_quad
    drawnow;
end

%% DESPLAZAMIENTOS DEL CUADRICOPTERO CON CONTROLADOR PID

% DESPLAZAMIENTO EN Z
figure()
subplot(3,1,1)
plot(TopGun,Z_pid.signals.values)
hold on;
plot(TopGun,Z_set.signals.values);
title('Altitud regulada por PID
 $f(z)$ ', 'fontweight', 'bold', 'Interpreter', ...
    'Latex', 'fontsize', 16)
legend('Z-fuzzy-pid', 'Z-ref')
ylabel(' [m] ', 'fontweight', 'bold', 'Interpreter', ...
    'Latex', 'fontsize', 16);
grid on;

% DESPLAZAMIENTO EN X
subplot(3,1,2)
plot(TopGun,X_out.signals.values);
hold on;
plot(TopGun,X_set.signals.values);
title('Desplazamiento
 $f(x)$ ', 'fontweight', 'bold', 'Interpreter', ...
    'Latex', 'fontsize', 16)
legend('X-pd', 'X-ref')
ylabel(' [m] ', 'fontweight', 'bold', 'Interpreter', ...
    'Latex', 'fontsize', 16);
grid on;

```

```

% DESPLAZAMIENTO EN Y
subplot(3,1,3)
plot(TopGun,Y_out.signals.values);
hold on;
plot(TopGun,Y_set.signals.values);
title('Desplazamiento
 $f(y)$ ', 'fontweight', 'bold', 'Interpreter', ...
      'Latex', 'fontsize', 16)
legend('Y-pd', 'Y-ref')
xlabel(' [s]', 'fontweight', 'bold', 'Interpreter', ...
      'Latex', 'fontsize', 16), ylabel(' [m]', 'fontweight', 'bold', 'Inter
preter', ...
      'Latex', 'fontsize', 16);
grid on;

%% DESPLAZAMIENTOS DEL CUADRICOPTERO CON CONTROLADOR FUZZY-PID

% DESPLAZAMIENTO EN Z
figure()
subplot(3,1,1)
plot(TopGun,Z_fuz.signals.values)
hold on;
plot(TopGun,Z_set.signals.values);
title('Altitud regulada por Fuzzy-PID
 $f(z)$ ', 'fontweight', 'bold', 'Interpreter', ...
      'Latex', 'fontsize', 16)
legend('Z-fuzzy-pid', 'Z-ref')
ylabel(' [m]', 'fontweight', 'bold', 'Interpreter', ...
      'Latex', 'fontsize', 16);
grid on;

% DESPLAZAMIENTO EN X
subplot(3,1,2)
plot(TopGun,X_out.signals.values);
hold on;
plot(TopGun,X_set.signals.values);
title('Desplazamiento
 $f(x)$ ', 'fontweight', 'bold', 'Interpreter', ...
      'Latex', 'fontsize', 16)
legend('X-pd', 'X-ref')
ylabel(' [m]', 'fontweight', 'bold', 'Interpreter', ...
      'Latex', 'fontsize', 16);
grid on;

```

```

% DESPLAZAMIENTO EN Y
subplot(3,1,3)
plot(TopGun,Y_out.signals.values);
hold on;
plot(TopGun,Y_set.signals.values);
title('Desplazamiento
 $f(y)$ ', 'fontweight', 'bold', 'Interpreter', ...
      'Latex', 'fontsize', 16)
legend('Y-pd', 'Y-ref')
xlabel(' [s]', 'fontweight', 'bold', 'Interpreter', ...
      'Latex', 'fontsize', 16), ylabel(' [m]', 'fontweight', 'bold', 'Inter
preter', ...
      'Latex', 'fontsize', 16);
grid on;

%% MOVIMIENTOS DE YAW, PITCH Y ROLL

% MOVIMIENTO YAW
figure()
subplot(3,1,1)
plot(TopGun, Psi_out.signals.values, 'g');
title('Yaw  $f(\psi)$ ', 'fontweight', 'bold', 'Interpreter', ...
      'Latex', 'fontsize', 16)
ylabel(' [rad]', 'fontweight', 'bold', 'Interpreter', ...
      'Latex', 'fontsize', 16);
grid on;

% MOVIMIENTO PITCH
subplot(3,1,2)
plot(TopGun, Phi_out.signals.values, 'r');
title('Pitch
 $f(\theta)$ ', 'fontweight', 'bold', 'Interpreter', ...
      'Latex', 'fontsize', 16)
ylabel(' [rad]', 'fontweight', 'bold', 'Interpreter', ...
      'Latex', 'fontsize', 16);
grid on;

% MOVIMIENTO ROLL
subplot(3,1,3)
plot(TopGun, Theta_out.signals.values, 'b');
title('Roll  $f(\phi)$ ', 'fontweight', 'bold', 'Interpreter', ...
      'Latex', 'fontsize', 16)
xlabel(' [s]', 'fontweight', 'bold', 'Interpreter', ...
      'Latex', 'fontsize', 16), ylabel(' [rad]', 'fontweight', 'bold', 'Int
erpreter', ...
      'Latex', 'fontsize', 16);
grid on;

```

```

%% RELACION ENTRE ROTACION Y TRASLACION

figure()
plot(TopGun,Phi_out.signals.values,'r')
hold on;
plot(TopGun,X_out.signals.values,'m');
hold on;
plot(TopGun,Theta_out.signals.values,'b')
hold on;
plot(TopGun,Y_out.signals.values);
title('Relaci\`on entre Rotaci\`on y
Traslaci\`on','fontweight','bold','Interpreter',...
'Latex','fontsize',16)
legend('theta','X','phi','Y');
grid on;

%% RESPUESTAS DE CONTROLADOR PID PARA ALTITUD

figure()
plot(TopGun,Z_set.signals.values);
hold on;
plot(TopGun,Z_pid.signals.values);
title('Altitud $f(z)$','fontweight','bold','Interpreter',...
'Latex','fontsize',16)
legend('Z-set','Z-pid');
xlabel('[s]','fontweight','bold','Interpreter',...
'Latex','fontsize',16),ylabel('[m]','fontweight','bold','Inter
preter',...
'Latex','fontsize',16);
grid on;

%% RESPUESTAS DE CONTROLADOR PID-FUZZY PARA ALTITUD

figure()
plot(TopGun,Z_set.signals.values);
hold on;
plot(TopGun,Z_fuz.signals.values);
title('Altitud $f(z)$','fontweight','bold','Interpreter',...
'Latex','fontsize',16)
legend('Z-set','Z-fuzzy-pid');
xlabel('[s]','fontweight','bold','Interpreter',...
'Latex','fontsize',16),ylabel('[m]','fontweight','bold','Inter
preter',...
'Latex','fontsize',16);
grid on;

```

```

%% RESPUESTAS DE CONTROLADOR PID Y CONTROLADOR FUZZY-PID PARA
ALTIMUD

figure()
plot(TopGun,Z_pid.signals.values);
hold on;
plot(TopGun,Z_fuz.signals.values);
hold on;
plot(TopGun,Z_set.signals.values);
title('Altitud  $f(z)$ ', 'fontweight', 'bold', 'Interpreter', ...
      'Latex', 'fontsize', 16)
legend('Z-pid', 'Z-fuzzy-pid', 'Z-ref');
xlabel(' [s]', 'fontweight', 'bold', 'Interpreter', ...
      'Latex', 'fontsize', 16), ylabel(' [m]', 'fontweight', 'bold', 'Inter
preter', ...
      'Latex', 'fontsize', 16);
grid on;

%% ERROR Y SALIDA DEL CONTROLADOR PID

% ERROR
figure()
subplot(2,1,1)
plot(TopGun,Error_z.signals.values, 'r')
title('Error PID', 'fontweight', 'bold', 'Interpreter', ...
      'Latex', 'fontsize', 16)
ylabel(' [m]', 'fontweight', 'bold', 'Interpreter', ...
      'Latex', 'fontsize', 16);
grid on;

% SALIDA DE CONTROLADOR PID
subplot(2,1,2)
plot(TopGun,PID_out.signals.values, 'b');
title('Salida Controlador
PID', 'fontweight', 'bold', 'Interpreter', ...
      'Latex', 'fontsize', 16)
xlabel(' [s]', 'fontweight', 'bold', 'Interpreter', ...
      'Latex', 'fontsize', 16)
ylabel(' [N]', 'fontweight', 'bold', 'Interpreter', ...
      'Latex', 'fontsize', 16);
grid on;

```

```
%% ERROR Y SALIDA DEL CONTROLADOR FUZZY-PID

% ERROR
figure()
subplot(2,1,1)
plot(TopGun,Error_zf.signals.values,'r')
title('Error Fuzzy-PID','fontweight','bold','Interpreter',...
      'Latex','fontsize',16)
ylabel('[m]','fontweight','bold','Interpreter',...
      'Latex','fontsize',16);
grid on;

% SALIDA DE CONTROLADOR FUZZY-PID
subplot(2,1,2)
plot(TopGun,Fuzzy_PID_out.signals.values,'b');
title('Salida Controlador Fuzzy-
PID','fontweight','bold','Interpreter',...
      'Latex','fontsize',16)
xlabel('[s]','fontweight','bold','Interpreter',...
      'Latex','fontsize',16)
ylabel('[N]','fontweight','bold','Interpreter',...
      'Latex','fontsize',16);
grid on;
```