



UNIVERSIDAD SANTO TOMÁS
FACULTAD DE INGENIERÍA ELECTRÓNICA

**Diseño e Implementación de estrategias de control con UAV con
fines de seguimiento de objetos**

TRABAJO DE GRADO
PARA OPTAR POR EL TÍTULO DE INGENIERO ELECTRÓNICO

DANIEL CAMILO SANABRIA BURGOS

Director: Ing. Edgar Andrés Gutiérrez Cáceres Esp.Msc.PhD.(st)

Co-director: Ing. Wiliam Fabian Chaparro Esp.Msc.PhD.(st)

Tunja, 2021



Universidad Santo Tomás



Universidad Santo Tomás
Facultad de Ingeniería Electrónica

Diseño e Implementación de estrategias de control con UAV con fines de seguimiento de objetos

TRABAJO DE GRADO
PARA OPTAR POR EL TÍTULO DE INGENIERO ELECTRÓNICO

DANIEL CAMILO SANABRIA BURGOS

Director: Ing. Edgar Andrés Gutiérrez Caceres Esp.Msc.PhD.(st)

Co-Director: Ing. Wiliam Fabian Chaparro Esp.Msc.PhD.(st)

Approved by

Date:.

(Firma)

(Firma)

.....
Edgar Andrés Gutiérrez Caceres
Docente Tutor

.....
William Fabian Chaparro
Docente Cocutor

.....
Nombre del Primer Lector
Docente

.....
Nombre del Segundo Lector
Docente

Tunja, 2021



Universidad Santo Tomás
Facultad de Ingeniería Electrónica

Copyright © – All rights free. Daniel Camilo Sanabria Burgos, 2021.

Las ideas presentadas en este trabajo de investigación son responsabilidad exclusiva del autor y no representan la opinión de la Universidad Santo Tomás ni la Facultad

(Signature)

.....
Daniel Camilo Sanabria
Burgos

Resumen

Se propone el diseño e implementación de un control PID que permita estimar los diferentes errores que pueden existir en cada uno de los movimientos que describe el drone. El control en cascada es una configuración de control en donde la salida de un controlador de retroalimentación es el punto de ajuste para otro controlador de retroalimentación, por lo menos. El control de cascada involucra sistemas de control de retroalimentación. Se plantea usar esta técnica de control y adaptarla para la estimación de error dentro de los movimientos y proporcionar así una referencia para el control PID para eliminar dicho error. Es por eso que se propone el diseño, implementación y comparación de estrategias de control servovisual para la detección y seguimiento de objetos dentro de un ambiente controlado haciendo uso de programas como ROS, Python y/o Matlab. De manera inicial el control se aplica para hacer un face Tracking mediante la implementación de un código en Python, esto se hace gracias al uso del software PyCharm, implementando en este un controlador PID, la finalidad de este proyecto es la del seguimiento de objetos y/o personas, y en dicho seguimiento que conserve un vuelo estacionario es decir que conserve una misma altura y una misma perspectiva del objeto que se está siguiendo

Palabras Claves

Seguimiento de objetos, UAV, agricultura de precisión, Control, ambientes controlados, PID.

Esta tesis esta dedicada para mis hermanos Sanabria Burgos y Orozco Burgos.

Índice general

Abstract	1
1. Introducción	9
1.0.1. Definición del problema	9
1.0.2. Delimitación del problema	10
1.1. Justificación	10
1.2. Formulación de preguntas	10
1.3. Objetivos	11
1.3.1. Objetivo general	11
1.3.2. Objetivos específicos	11
2. Marco Teorico	12
2.1. Modelo Cinemático del UAV	12
2.1.1. Rotación Angular - Roll ϕ	12
2.1.2. Rotación Angular -Pitch Θ	12
2.1.3. Rotación Angular - Yaw Ψ	13
2.2. Control PID	13
2.2.1. Proporcional	14
2.2.2. Integral	14
2.2.3. Derivativo	14
2.3. Control por Lógica Difusa	14
2.3.1. Funciones de Membresia	15
2.3.2. Fuzzificacion	17
2.3.3. Reglas	17
2.4. Inferencia	18
2.5. Defusificacion	18
2.6. Estado del Arte	20
2.7. Marco Legal	21
3. Desarrollo	22
3.1. Arquitectura de Control	22
3.2. Detección de objetos	23
3.2.1. Reconocimiento Facial	23
3.2.2. Detección de Cuerpo	24
3.3. Diseño y simulación de controlador PID	24
3.3.1. Área	24
3.3.2. Coordenadas de CX Y CY	25
3.3.3. Identificación del Sistema	26
3.4. Diseño y simulación de controlador Lógica Difusa	32
3.4.1. Funciones de Membresia para Yaw	33
3.4.2. Funciones de Membresia para el eje Y	33
3.4.3. Funciones de Membresia para el Área	35

3.4.4. Método de inferencia	35
3.4.5. Defuzzificacion	36
4. Análisis de Resultados	38
4.0.1. Gráfica de error y acción de control en Yaw	38
4.0.2. Gráfica de error y acción de control para el eje Y	39
4.0.3. Gráfica de error y acción de control para el eje Z	40
4.0.4. Conclusiones	42
APPENDICES	43
A. Participación en eventos	44
A.1. V Encuentro Zonal de Semilleros de Investigación	44
A.2. VI Congreso Internacional de Instrumentación, Control y Telecomunicaciones . . .	44
Bibliography	45

Índice de figuras

2.1.	Esquema de Cuadricoptero <i>Fuente: [1]</i>	12
2.2.	Estructura del PID <i>Fuente: https://commons.wikimedia.org</i>	13
2.3.	Estructura de la Logica Difusa <i>Fuente: [2]</i>	14
2.4.	Función de Membresia Triangular - Fuente: [3]	15
2.5.	Funcion de Membresia Trapezoidal - Fuente: [3]	15
2.6.	Funcion de Membresia Gaussiana - Fuente: [3]	16
2.7.	Función de Membresia Campana - Fuente: [3]	16
2.8.	Función de Membresia Sigmoidal - Fuente: [3]	16
2.9.	Ejemplo de Fuzzificacion de una variable Fuente: [3]	17
2.10.	Ejemplo de Desfusicacion de una variable - Fuente: Autor	19
2.11.	Curvas que describen cada tipo de defusicacion - Fuente: Autor	20
3.1.	Arquitectura de control - Fuente: Autor	22
3.2.	Reconocimiento Facial mediante el uso de Media Pipe - Fuente: Autor	23
3.3.	Reconocimiento de Pose - Fuente : Autor	24
3.4.	Interpretación del área del recuadro delimitador con respecto al Frame - Fuente: Autor	25
3.5.	Movimientos que describe el UAV con respecto al área detectada - Fuente: Autor	25
3.6.	Movimientos que describe el UAV con respecto a la posición del Recuadro delimitador para el ángulo en Yaw - Fuente: Autor	26
3.7.	Respuesta al paso en Yaw - Fuente: Autor	26
3.8.	Identificación del Sistema con respecto al movimiento en Yaw - Fuente: Autor	27
3.9.	Respuesta del Sistema con las constantes - Fuente: Autor	27
3.10.	Respuesta al paso con respecto al movimiento de subir y bajar el UAV - Fuente: Autor	28
3.11.	Identificación del Sistema con respecto al movimiento de subir y bajar con el UAV - Fuente: Autor	28
3.12.	Respuesta del Sistema con las constantes - Fuente: Autor	29
3.13.	Respuesta al paso con respecto al movimiento de alejar y acercar con respecto al área del recuadro delimitador el UAV - Fuente: Autor	29
3.14.	Identificación del Sistema con respecto al movimiento de acercarse y alejarse con el UAV	30
3.15.	Respuesta del Sistema con las constantes	30
3.16.	Respuesta al paso con respecto al movimiento de alejar y acercar con respecto al área del recuadro delimitador de la detección de Pose	31
3.17.	Identificación del Sistema con respecto al movimiento de acercarse y alejarse con el UAV	31
3.18.	Respuesta del Sistema con las constantes	32
3.19.	Funciones de membresia tanto para el error como para la velocidad en Yaw - Fuente: Autor	33
3.20.	Funciones de membresia tanto para el error como para la velocidad en Y - Fuente: Autor	34

3.21.	Funciones de membresia tanto para el error como para la velocidad en Z - Fuente: Autor	35
3.22.	Funciones de membresia para el error con respecto al área del recuadro delimitador de Body Following - Fuente: Autor	35
3.23.	Lógica que se maneja en la inferencia difusa Fuente: [4]	36
3.24.	Gráfica de Valor de pertenencia y método de inferencia - Fuente: Autor	36
3.25.	Gráfica de defuzzificacion - Fuente: Autor	37
4.1.	Gráfica de Respuesta del error y la acción de control por medio de Control Clásico - Fuente: Autor	39
4.2.	Gráfica de Respuesta del error y la acción de control por medio de Control no convencional - Fuente: Autor	39
4.3.	Gráfica de Respuesta del error y la acción de control por medio de Control Clásico - Fuente: Autor	40
4.4.	Gráfica de Respuesta del error y la acción de control por medio de Control Clásico - Fuente: Autor	40
4.5.	Gráfica de Respuesta del sistema mediante dos estrategias de control - Fuente: Autor	42
4.6.	Gráfica de Respuesta del sistema mediante dos estrategias de control - Fuente: Autor	42
A.1.	Certificado de participación	44

Índice de cuadros

2.1.	Definición de los cuatro métodos de inferencia mas populares [5]	18
2.2.	Marco de antecedentes <i>Fuente: Autor</i>	20

Capítulo 1

Introducción

Actualmente, se puede esperar que las personas participen en la ejecución de una determinada tarea y, en algunos casos, conllevan riesgos considerables en el instante de desarrollarlo. Esto requiere el uso de mecanismos automatizados para reducir los riesgos que enfrenta el personal. Esto ha llevado al desarrollo de sistemas autónomos en diferentes países, entre los cuales la diferencia entre países desarrollados y países en proceso radica en el grado de automatización industrial y El sistema de control que puede reflejarse en el proceso de producción o el producto que contiene el sistema de control integrado.

La implementación de un sistema de control implica una visión tanto tecnológica como económica y de seguridad en la operación, que se une para así obtener una serie de acciones continuas, precisas y de manera eficiente, las cuales tienen que ser controladas y a su vez estar monitoreadas con el objetivo de conseguir el mayor beneficio posible, esto se hace con la finalidad de conseguir reducir los riesgos laborales, riesgos de operación y a su vez operaciones que sean imposible ejecutarlas; para que un sistema sea autónomo se tiene que partir de la premisa que dentro del sistema las tareas tienen que ser ejecutadas sin la intervención de un humano, sin embargo en la mayoría de los casos las plataformas de control son moderados mediante una base de control.

Al pasar de los años se ha podido observar el aumento en la utilidad de los UAV (Unmanned Aerial Vehicle), que son conocidos comúnmente como Drones, esto gracias a los diferentes progresos que se han hecho frente a la tecnología de control, diseño y comunicaciones, gracias a estos avances podemos acceder a los drones de manera menos costosa, lo cual da la posibilidad para que las empresas en el sector de la industria o las personas del común puedan trabajar en proyectos con los drones. Recordemos que los drones cuentan con una amplia gama de aplicación o funcionalidad, desde la agricultura de precisión hasta el rescate de personas. [6] En cualquiera de los casos estas todas estas implementaciones se basan en la interacción de nuestro dron con el ambiente es decir un ambiente no controlado, donde nuestro dron hace las respectivas mediciones del entorno y estas se envían remotamente al usuario, o dichas mediciones las obtiene para así poder determinar la ruta o movimientos que se deseen hacer.

Para este tipo de Robots o sistemas es imprescindible contar con un controlador apropiado, de manera generalizada se tiene que tener en cuenta el modelo cinemático y dinámico para diseñar un controlador que funcione para la planta en este caso el dron, posteriormente se elige el tipo de controlador, en el presente trabajo de investigación se implementará control clásico con PID y control no convencional mediante lógica difusa para seguimiento de objetos.

1.0.1. Definición del problema

Al pasar de los años hemos podido observar el aumento en la utilidad de los UAV (Unmanned Aerial Vehicle), que son conocidos comúnmente como Drones, esto gracias a los diferentes progresos que se han hecho frente a la tecnología de control, diseño y comunicaciones, gracias a estos avances podemos acceder a los drones de manera menos costosa, lo cual da la posibilidad para que las empresas en el sector de la industria o las personas del común puedan trabajar en proyectos con los drones. Recordemos que los drones cuentan con una amplia gama de aplicación o funcionalidad,

desde la agricultura de precisión hasta el rescate de personas. [6] En cualquiera de los casos estas implementaciones se basan en la interacción de nuestro dron con el ambiente es decir un ambiente no controlado, donde nuestro dron hace las respectivas mediciones del entorno y estas se envían remotamente al usuario, o dichas mediciones las obtiene para así poder determinar la ruta o movimientos que se deseen hacer. El seguimiento de objetos es una herramienta que es útil para diferentes implementaciones. Por ejemplo, en el caso de las vías ferreas, en este caso es de mucha mas comodidad enviar el dron con la finalidad de monitorear la vía, con el objetivo de que el dron se maneje de forma automática, de tal manera que los operarios cuenten con la única tarea de identificar anomalías en la vía y no tener la tediosa tarea de controlar el dron, de la misma manera en la industria cinematográfica actualmente se usan drones para obtener tomas de diferentes ángulos que tiempo atrás se creía no eran posibles de captar

1.0.2. Delimitación del problema

La elaboración de sistemas de control para UAV es un proceso conformado por diferentes etapas, dado que cada UAV cuenta con diferentes especificaciones técnicas y por ende una diferente función de transferencia entre ellos, el elaborar un sistema de control generalizado para cualquier quadcopter resulta casi imposible dado que los rotores no cuentan con las mismas especificaciones técnicas y en si la dinámica de cada sistema UAV es diferente. Es por esto que en el desarrollo de este proyecto la estrategia de control se plantea para implementarla en el Dji Tello, por medio de la elaboración de un algoritmo de control en Python que junto con las librerías de OpenCV contara con la posibilidad de reconocer y hacer seguimiento de objetos.

1.1. Justificación

Los robots aéreos son dispositivos que los podemos encontrar con mucha mas frecuencia en nuestro entorno común, esto se logra gracias a la miniaturización de los componentes electrónicos que contienen este, haciendo posible utilizarlos en ocasiones muy variadas dentro de las que podemos destacar la localización de personas, filmación de eventos y/o seguimiento de personas dentro de la industria del deporte, en este caso el ciclismo; las tomas de la carrera podría mejorarse si el dron enfocara automáticamente el objeto de interés que en este caso sería el pelotón de la carrera para seguirlos y que a su vez envíe estos datos al camarógrafo que esta ya sea en moto o en carro. [7]. Dentro de la carrera de ingeniería electrónica las áreas de robótica e imágenes muestran una gran oportunidad de investigación. Por tal motivo, se propone el diseño, implementación y comparación de estrategias de control servovisual en una plataforma de UAV para la identificación y seguimiento de objetos dentro de un ambiente controlado. Buscando aportar e impactar el conocimiento en la búsqueda a través de sistemas robóticos visuales para usos como detección de personas y/o objetos. A partir de los ejemplos previamente mencionados, se puede observar la necesidad que tiene la industria y el mercado en general con respecto al manejo autónomo de vehículos aéreos no tripulados. El enfoque de este proyecto esta enfocado hacia el reconocimiento de imágenes mediante la implementación de visión artificial y el control, dado que en la mayoría de los casos la vía de comunicación entre el usuario y el quadcopter es una cámara.

1.2. Formulación de preguntas

- ¿Que estrategias de control existen actualmente para el seguimiento de objetos con UAV?
- ¿Cuales son las ventajas o desventajas que pueden tener las diferentes estrategias de control frente al seguimiento de objetos?
- ¿Mediante qué tecnología o algoritmo se puede realizar una estrategia de control para el seguimiento de objetos, que sea efectivo en reducir el error?

1.3. Objetivos

1.3.1. Objetivo general

Analizar y comparar las estrategias de control PID y Lógica Difusa para el seguimiento de objetos mediante UAV.

1.3.2. Objetivos específicos

- Analizar el funcionamiento de los sistemas de control para el seguimiento de objetos mediante UAV
- Definir la arquitectura de control para determinar la variable a controlar y las diferentes variables que pueden afectar a nuestro UAV
- Diseñar un controlador PID y por lógica difusa, para pitch, yaw y roll, mediante python y MatLab
- Implementar y Analizar el comportamiento del UAV para una trayectoria definida(Path Planning)
- Divulgar el proyecto de investigación en diferentes medios académicos y de Investigación científica

Capítulo 2

Marco Teorico

2.1. Modelo Cinemático del UAV

En una primera instancia tenemos que tener en cuenta que el UAV con el que se trabajó a lo largo del desarrollo de esta tesis fue un cuadricoptero es decir una nave con cuatro rotores, un rotor frontal, uno lateral derecho, un posterior y finalmente uno lateral izquierdo como se puede observar en la siguiente figura. El movimiento de la aeronave de cuatro ejes es rotación y traslación,

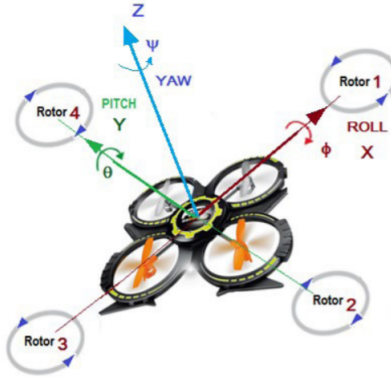


Figura 2.1: Esquema de Cuadricoptero Fuente: [1]

que se determinan en tres ejes fijos (X, Y, Z) y oscilan a través del ángulo de navegación: guiñada (eje Z); cabeceo (eje Y), y finalmente balanceo (Eje X), consideremos también que el quadcopter tiene seis grados de libertad. A continuación veremos las tres rotaciones angulares con las que cuenta el dron.

2.1.1. Rotación Angular - Roll ϕ

Esta rotación recae principalmente sobre el eje x. esta rotación se obtiene mediante la variación de las velocidades de los rotores lateral derecho y lateral izquierdo, manteniendo en velocidad constante los rotores restantes, este movimiento también se conoce como Alabeo, hacia la derecha y la izquierda.

Matriz de Rotación

$$R(x, \phi) = \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} = \begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} \quad (2.1)$$

2.1.2. Rotación Angular -Pitch Θ

Esta rotación se hace sobre el eje y, esta rotación se obtiene mediante la variación de las velocidades del rotor frontal y el rotor posterior, manteniendo en velocidad constante los rotores

restantes, este movimiento también se conoce como Cabeceo, hacia la adelante y atrás
Matriz de Rotacion

$$R(y, \theta) = \begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} = \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} \quad (2.2)$$

2.1.3. Rotación Angular - Yaw Ψ

Esta rotación cae en el eje z. El movimiento de guiñada o guiñada se obtiene aumentando la velocidad de los rotores delantero y trasero y reduciendo adicionalmente la velocidad de los rotores derecho e izquierdo, lo que tiene en cuenta el movimiento del dron hacia la derecha. Esto crea un desequilibrio aerodinámico de torque, lo que permite que el quadcopter gire alrededor de su propio eje. Si todos los rotores mantienen la misma velocidad, el par total es cero; además, si el empuje total de la aeronave es igual a su peso, permanecerá en vuelo estacionario.

Matriz de Rotación

$$R(z, \Psi) = \begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} = \begin{bmatrix} \cos\Psi & -\sin\Psi & 0 \\ \sin\Psi & \cos\Psi & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} \quad (2.3)$$

2.2. Control PID

Un controlador PID (controlador proporcional, integral y derivativo) es una estrategia de control que cuenta con un lazo de retroalimentación el cual nos permite regular una variable de interés a controlar que en dicho caso puede ser tanto la velocidad, temperatura, presión o flujo entre otras variables que pueden existir. [8] En general el controlador se calcula hallando la diferencia entre variable real contra la variable que desea controlar. Su formulación, en el espacio de Laplace viene dado por la siguiente ecuación:

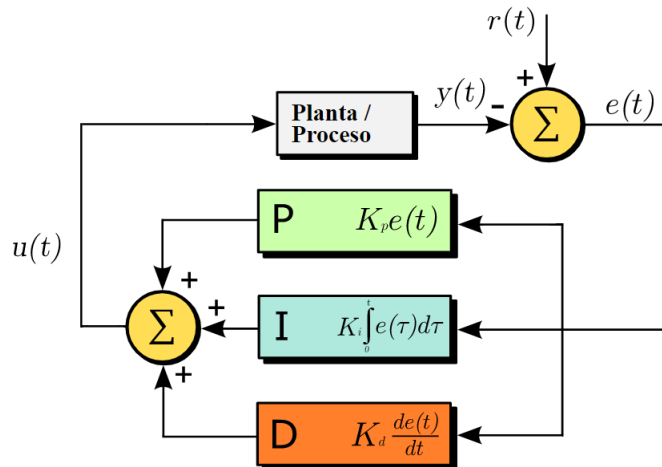


Figura 2.2: Estructura del PID Fuente: <https://commons.wikimedia.org>

$$D(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (2.4)$$

2.2.1. Proporcional

$$P = K_p e(t) \quad (2.5)$$

La parte proporcional del controlador se basa en el producto entre la señal de error que en la ecuación anterior viene expresada por $e(t)$ y la constante proporcional K_p esto se hace con la finalidad de lograr que el error de estado estacionario se acerque al cero. Tengamos en cuenta que el error es directamente proporcional a la acción proporcional es decir que entre mas grande sea el error mas incidente es la acción de control, aumentando la constante proporcional podemos aumentar la velocidad de respuesta de nuestro sistema pero esto puede llevar a la inestabilidad.

2.2.2. Integral

$$I = \frac{K_p}{T_i} \int e(t) \cdot dt \quad (2.6)$$

La parte integral tiene como función el disminuir y eliminar el error de estado estacionario, los cuales son provocados por perturbaciones exteriores, los cuales la parte proporcional no es capaz de eliminar y/o corregir. Cabe resaltar que esta parte del controlador, puede inestabilizar el sistema de control dado que añade un polo en cero.

2.2.3. Derivativo

$$D = K_p T_d \frac{de(t)}{dt} \quad (2.7)$$

La parte de acción derivativa tiene como principal función el de mantener el error al mínimo es decir que tiene que corregir proporcionalmente con la misma velocidad que produce, con la finalidad de evitar que el margen de error inicie a incrementarse. En otras palabras esta parte del control ayuda a disminuir el sobre pico de error

2.3. Control por Lógica Difusa

Para definir el control difuso debemos previamente definir que es la lógica difusa, la cual la podemos describir como si fuese un sistema de tipo interpretativo, en el cual el entorno o los objetos son relacionados con funciones o conjuntos de fronteras no son claramente definidas, de una manera mucho mas generalizada podemos expresar que existe una especie de interpolación entre fronteras. [9] El control Difuso, se puede expresar de una mejor manera como un controlador por medio de palabras que interpretan la lógica común, en vez de los números y/o ecuaciones. Recordemos que la fusificación, otorga los datos de entrada un grado de membresia dentro de las distintas expresiones posibles, para ello se busca la correspondencia entre el estado de las variables y las funciones de pertenencia definidas para tal propósito. [2]

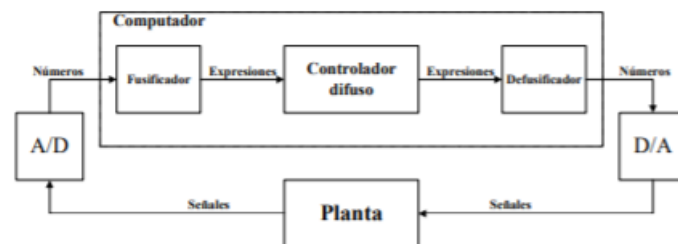


Figura 2.3: Estructura de la Logica Difusa Fuente: [2]

2.3.1. Funciones de Membresia

La función de pertenencia representa el grado de pertenencia del elemento al subconjunto definido por la etiqueta. La función de pertenencia tiene muchas formas, las más comunes son trapezoide, triángulo, singleton y S. A continuación se van a enlistar las clases de funciones de membresia previamente mencionadas, describiendo los diferentes parámetros que dentro de un sistema difuso y que la mayoría de las veces son usadas para describir la función de mebresia ya sea en una o en dos dimensiones

Forma Triangular

Este tipo de función de membresia se describe mediante tres parametros (a,b,c) de la siguiente manera:

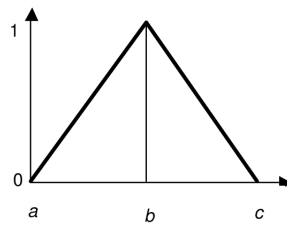


Figura 2.4: *Función de Membresia Triangular* - Fuente: [3]

$$Triangular(x, a, b, c) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ \frac{c-x}{c-b} & b \leq x \leq c \\ 0 & c \leq x \end{cases} \quad (2.8)$$

Forma Trapezoidal

Este tipo de función de membresia viene especificada por cuatro parametros a,b,c,d de la siguiente manera:

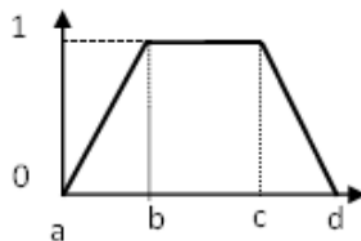


Figura 2.5: *Funcion de Membresia Trapezoidal* - Fuente: [3]

$$Trapezoidal(x, a, b, c, d) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ 1 & b \leq x \leq c \\ \frac{c-x}{c-b} & c \leq x \leq d \\ 0 & d \leq x \end{cases} \quad (2.9)$$

Función Gaussiana

Esta función de membresía viene especificada por dos parámetros $[c, \sigma]$ donde c representa el centro de la función de membresía, mientras que σ representa el ancho de la banda de la función de membresía

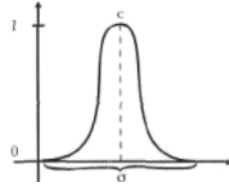


Figura 2.6: Función de Membresía Gaussiana - Fuente: [3]

$$\text{Gaussiana}(x, c, \sigma) = e^{-\frac{1}{2}(\frac{x-c}{\sigma})^2} \quad (2.10)$$

Función Campana

Esta función está especificada mediante tres parámetros (a, b, c) donde usualmente b es positivo, si este parámetro es negativo la forma de la función de membresía será una campana inversa.

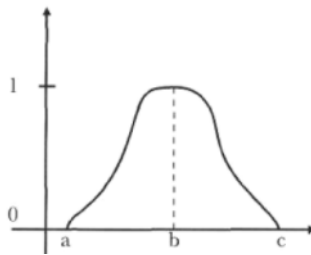


Figura 2.7: Función de Membresía Campana - Fuente: [3]

$$\text{Campana}(x; a, b, c) = \frac{1}{1 + \left|\frac{x-c}{a}\right|^{2b}} \quad (2.11)$$

Función Sigmoideal

Esta función se define dentro de dos parámetros los cuales son (a, c) donde a es el encargado de controlar la cuesta al punto donde se obtiene el cruce a $x = c$. Cabe resaltar que dependiendo del signo del parámetro a , la función de membresía sigmoideal será abierta por la derecha o abierta por la izquierda

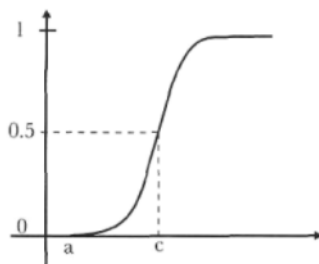


Figura 2.8: Función de Membresía Sigmoideal - Fuente: [3]

$$\text{Sigmoideal}(x; a, c) = \frac{1}{1 + e^{-a(x-c)}} \quad (2.12)$$

2.3.2. Fuzzificacion

En la teoría de control difuso siempre se ve involucrado este proceso de Fuzzificacion, esta operación se realiza en todo momento, y es la puerta de entrada al sistema de inferencia difusa. Es un proceso matemático en el que un elemento del campo del discurso (una variable de medida del proceso) se transforma en un valor en cada función de pertenencia, donde pertenece. [10]

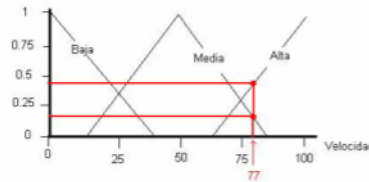


Figura 2.9: Ejemplo de Fuzzificacion de una variable Fuente: [3]

2.3.3. Reglas

El control mediante lógica difusa implementa un conjunto de reglas que combinan uno o mas funcione de membresia de entrada los cuales reciben el nombre de antecedentes o premisas y estos se asocian a una función de membresia de salida que se denomina como consecuencia. A este conjunto de reglas se les denomina como fuzzy rules. La estructura de dichas reglas son del tipo SI-ENTONCES.

Reglas Difusas de Mamdani

La estructura que comúnmente sigue este tipo de reglas difusas es: **IF X_1 IS A AND X_2 IS B AND X_3 IS C THEN U_1 IS D , U_2 IS E** [5], esto teniendo en cuenta que nuestro sistema difuso cuente con tres variables de entrada las cuales son x_1, x_2 y x_3 donde cada una de estas entradas pueden ser el error, la derivada del error y por ultimo la segunda derivada del error, ahora , A, B, C vienen siendo las funciones de membresia de entrada ; u_1 y u_2 , son las acciones de control que en un ejemplo en particular podría ser el voltaje de un motor y/o la velocidad del mismo, en cierto sentido estas constantes todavía son variables lingüísticas es decir que no toman un valor numero, D Y E son las funciones de membresia de salida, en la mayorías de los casos se emplea singleton por la versatilidad computacional, y finalmente el uso de la AND juega con el papel de ser un operador lógico difuso, donde cabe resaltar se podría implementar otro.

Ventajas:

- Es Intuitivo
- Cuenta con alto margen de aceptación
- Esta bien optimizado para la incorporación de conocimiento y experiencia

Reglas Difusas de Takagi-Sugeno

La estructura que comúnmente sigue este tipo de reglas difusas es: **IF X_1 IS A AND X_2 IS B AND X_3 IS C THEN $U_1 = f(x_1, x_2, x_3)$, $u_2 = g(x_1, x_2, x_3)$** [5], en este tipo de reglas difusas se cuenta con la posibilidad de emplear dos funciones con la característica que pueden ser no lineales $f()$ y $g()$, el inconveniente con la implementación de dichas es que la elección de las funciones pueden ser complejas.

Ventajas

- Alta eficiencia computacional.

- Adecuado para tecnología lineal (por ejemplo, se puede utilizar para Controlador PID).
- Adecuado para tecnología de optimización y control adaptativo.
- Se Asegura de tener una superficie de control continua.
- Muy adecuado para análisis matemático.

2.4. Inferencia

El conjunto de las reglas difusas nos simbolizan la inteligencia y la estrategia de control, partiendo de la premisa de haber información objetiva en las variables de entrada. La inferencia difusa juega un papel importante en el control difuso dado que sirve para calcular en que conjunto difuso de salida esta cada una de las reglas, después de eso, junto con la salida de otras reglas, el sistema se cierra.

Hay muchos métodos de inferencia difusa, pero hay cuatro que producen mejores resultados en el campo del control, los cuales son, inferencia mínima de Mamdani, RM, inferencia de producto de Larsen, RL e inferencia de producto drástico, RDP y razonamiento de producto acotado, RBP [5] [11]

Metodo de Inferencia	Definicion
Mamdani minimum inference, Rm	$\min(\mu, \mu_w(z), \forall z)$
Larsen product inference, RL	$\mu \times \mu_w(z), \forall z$
Drastic Product inference Rdp	$\begin{cases} \mu & \text{para } \mu_w(z) = 1 \\ \mu_w(z) & \text{para } \mu = 1 \\ 0 & \text{para } \mu < 1 \text{ y } \mu_w(z) < 1 \end{cases}$
Bounded product inference, Rbp	$\max(\mu + \mu_w(z) - 1, 0)$

Cuadro 2.1: Definición de los cuatro métodos de inferencia mas populares [5]

2.5. Defusificacion

La defusificacion es una transformación matemática para convertir un conjunto difuso en un numero real, esto se logra mediante la obtención de una conclusión por parte del sistema de inferencia teniendo en cuenta la información de entrada. Esta conclusión o salida difusa se obtiene mediante la etapa de inferencia difusa, que genera un conjunto difuso, pero los datos de salida del sistema deben ser números reales y deben representar el conjunto completo obtenido en la etapa de agregación, por ende en el mundo de la lógica difusa hay varios tipos de defusificacion que entre estos diferentes tipos de densificación el que mas destaca es el centroide. En la siguiente ecuación podremos ver la expresión matemática para la defusificacion por centro de gravedad

$$y_d = \frac{\int_s \mu_y dy}{\int_s \mu_y dy} \quad (2.13)$$

Donde:

- μ_y = Funcion de pertenencia del conjunto de salida Y
- y variable de salida
- S = Dominio o rango de integración

En la ecuación 2.13 podemos observar como matemáticamente esta descrita la defusificacion por centro de gravedad, como se dijo anteriormente la metodología que se usa con mas frecuencia para defusificar es el centro de área (*COA*, *center of area*), la ecuación 2.13 es una aproximación de un arreglo de masas puntuales, esto haciéndolo en contraste con el centroide se obtiene que el centro de masa corresponde a cada regla del grado de pertenencia. A continuación observaremos la definición matemática para el centro de área.

$$\frac{\sum_{l=1}^R \delta_l \mu_{B_1}(\delta l)}{\sum_{l=1}^R \mu_{B_1}(\delta l)} \quad (2.14)$$

Donde:

- δ_l = Centro de gravedad del conjunto difuso
- B_1 = l-esima regla

Ahora veremos los diferentes tipos de defusificacion que existe ademas del centroide.

- **Criterio de máximo** donde la salida es el máximo valor que alcanza dentro de la función de membresia
- **El metodo de centro de area o de gravedad COA o COG** La salida es correspondiente a las ecuaciones 2.13 y 2.14
- **Metodo de la media de Maximo MOM** donde la salida es valor medio de las funciones de membresia que alcanzan el valor máximo
- **Metodo del maximo valor mas pequeño SOM** La salida es el mínimo valor de todos aquellos que generan el valor más alto de la función de membresía [12]
- **Metodo del maximo mas grande** Donde la salida es el valor de todas aquellas funciones de membresia que generan el valor mas alto.
- **Bisector de área** Donde la salida es el valor el cual separa de manera equidistante el área bajo la curva convirtiendo así el área en dos áreas iguales.

A continuación se podrán observar tanto como se obtienen los valores de manera gráfica, como, los tipos de curvas que describen cada uno de estas metodologias.

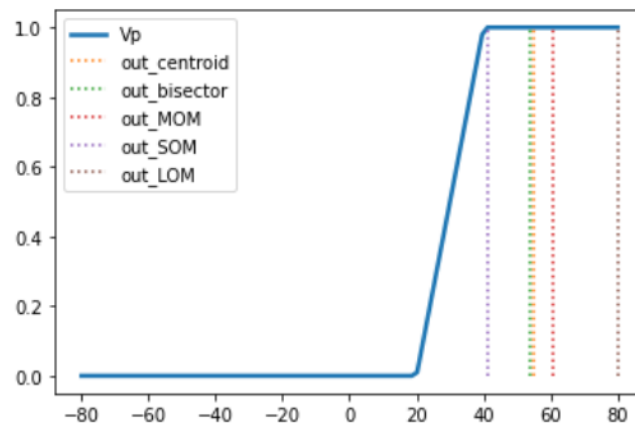


Figura 2.10: Ejemplo de Desfusificacion de una variable - Fuente: Autor

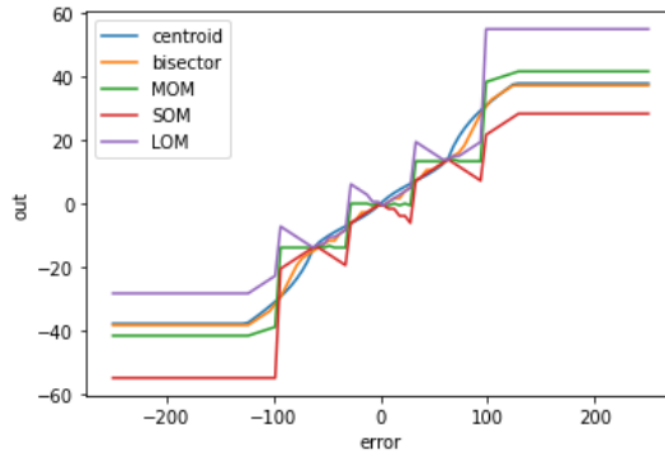


Figura 2.11: Curvas que describen cada tipo de defusificación - Fuente: Autor

Cabe resaltar que estas graficas cuentan con valores de defusificación como los siguientes:

- out centroid = 55.07673132351795
- out bisector = 54.14141414141415
- out MOM = 60.6060606060606
- out SOM = 41.212121212121204
- out LOM = 80.0

2.6. Estado del Arte

Titulo	Autores	Objetivo del Proyecto	Año
Control de un Drone Basado en seguimiento de objetos	Esteban Rivera	Diseñar e implementar el sistema de control de posición para un Drone basado en seguimiento de Objetos	2015 Bogota
Diseño y construcción de un dron para el seguimiento de un objeto de forma autónoma mediante visión artificial	Francisco Fernandez	Diseñar y construir un dron para el seguimiento de objetos mediante vision artificial	2017 Alcala
Seguimiento Automático de Objetivos con Drones Mediante Algoritmos de Tracking	Jesus Bescos	Garantizar el funcionamiento de algoritmos de videovigilancia de un seguimiento concretamente , a largo plazo y/o en grandes áreas	2016 Madrid

Cuadro 2.2: Marco de antecedentes Fuente: Autor

Con respecto al estado del arte de este proyecto, se han desarrollado con anterioridad, los cuales podemos observar en el cuadro 2.2 cabe resaltar que existen varios proyectos de este tipo de los cuales solo se seleccionaron tres como podemos observar en el cuadro 4.1, teniendo en cuenta que hicieran uso de una plataforma UAV para el respectivo seguimiento de objetos. El proyecto «Control de un Drone basado en seguimiento de objetos» desarrollado por Esteban Rivera, propone una problemática que tiene que ver con la importancia de la implementación de sistemas de control para seguimiento de objetos haciendo énfasis en la industria, y como esta se ve beneficiada con el uso de plataformas UAV donde este toma mediciones y de forma remota las envía a usuario o al operario, evitando así que el operario tenga la tediosa tarea de tener que ir de forma manual a tomar dichas variables, plantea el diseño e implementación una estrategia de control por medio de un controlador PID para el seguimiento de objetos, mediante el uso del software de ROS y

OpenCv. [6] Por ultimo se propone la implementación de un filtro de kalman para que funcione como un estimador de optimización en el sentido que minimiza el error de estimación dentro del controlador. Por ultimo se propone la elaboración de un algoritmo para seguimiento de objetos mediante visión artificial, al igual que Francisco Fernandez en su proyecto *«Diseño y construcción de un drone para el seguimiento de un objeto de forma autónoma mediante visión artificial»* pero el enfoque que este le da a su proyecto se basa en la elaboración del diseño y la construcción de un UAV o quadcopter ligero robotizado con la finalidad de seguimiento de objetos, para este caso el autor hace uso del software de LibrePilot y junto con una placa procesadora de imagen suministra los comandos para el respectivo seguimiento de objetos. [13]

Como se puede observar cada uno de estos proyectos se enfocan en el mejoramiento de algoritmos de tracking aplicando control y/o elaborando por si mismo la plataforma del UAV, la mayoría de los proyectos buscan aprovechar la ventaja de usar camaras moviles que estan ubicadas en los drones en lugar de las camaras fijas con fines de videovigilancia como lo es el seguimiento de objetos , tal como lo plantea Jesus Bescos Cano en su proyecto *«Seguimiento automático de objetivos con drones mediante algoritmos de tracking»* que se enfoca en el objetivo de desarrollar y mejorar algoritmos de videovigilancia y de seguimiento de objetos concretamente en ambientes no controlados. [14]

2.7. Marco Legal

Dentro del marco de regulación legal para el uso de los drones en Colombia la Aerocivil por medio de la norma 04201 [15] genero la respectiva normatividad que rige sobre el uso de Drones en Colombia dentro de las restricciones mas importantes que rige el proyecto están:

- Altura de vuelo máxima de 150m
- Distancia máxima de 750m
- Se prohíbe volar sobre aglomeraciones de gente
- Se prohíbe operar a menos de 1800m de instalaciones gubernamentales
- Se prohíbe la operación de sistemas de mas de 25 Kg de peso

Teniendo en cuenta que el proyecto de investigación que se esta llevando a cabo es de dimensiones limitadas, podemos afirmar que no cuenta con algún problema que prohíba el uso del UAV en lo que respecta a la regulación de impuesta por la Aerocivil, dado que nuestro proyecto se va llevar a cabo en ambientes controlados y las distancias de operación son cortas

3.1. Arquitectura de Control

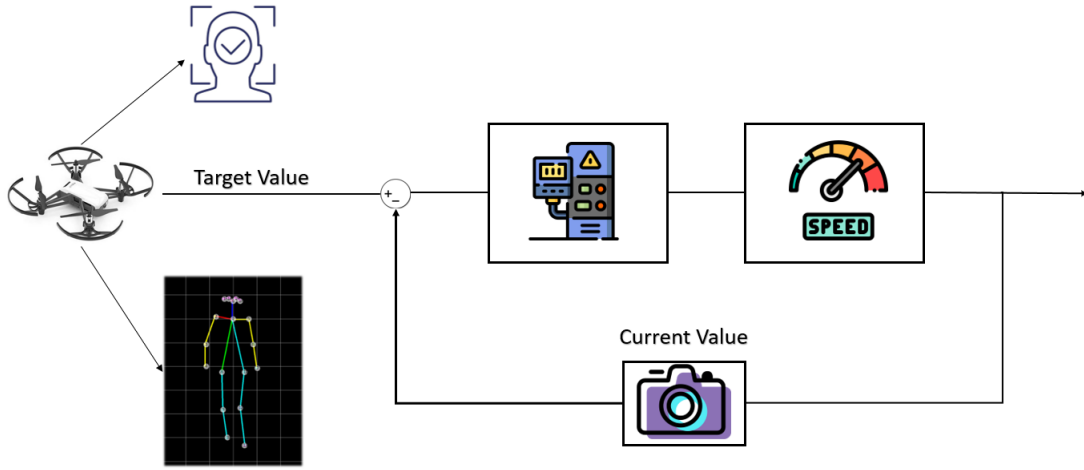


Figura 3.1: Arquitectura de control - Fuente: Autor

Previo al desarrollo del diseño e implementación de la estrategia de control tenemos que tener en cuenta una arquitectura de control, en la cual se plantea que se hará un control de alto nivel, es decir que no se afectará el controlador interno que cuenta el dron en cada uno de sus motores, en una primera instancia se hace el reconocimiento del objeto a seguir, para el desarrollo de esta investigación se optó por Face Tracking y Body Tracking, para así por medio de un BBOX (*Bounding box*) obtener las coordenadas de posición en píxeles, teniendo en cuenta que el centro del frame o de la captura de vídeo del dron va ser la variable de referencia o valor objetivo a controlar y el centro del *Bounding Box* va ser el valor actual el cual va funcionar, cabe resaltar que para esta investigación se optó por generar un controlador independiente para cada uno de los movimientos que describe el dron, dado que, un solo controlador para todos los movimientos generaría conflicto entre cada uno de los movimientos, es decir, que se contara con diferentes valores de referencia e independientes constantes de control y para hallar los diferentes errores con respecto a la variable a controlar se hace de la siguiente manera:

$$\text{Error en } x = w_i/2 - cx \quad (3.1)$$

$$\text{Error en } y = h_i/2 - cy \quad (3.2)$$

$$\text{Error en area} = \text{Area definida por el usuario} - \text{area actual} \quad (3.3)$$

Donde :

- $hi//2$ = el centro de la imagen teniendo en cuenta el alto de la misma
- $wi//2$ = el centro de la imagen teniendo en cuenta el ancho de la misma
- cx = el valor actual del centroide con respecto al ancho del recuadro delimitador
- cy = el valor actual del centroide con respecto al alto del recuadro delimitador

Teniendo en cuenta la ecuación 3.3 y 3.2 se da a entender que el sensor que realimenta el sistema va ser la cámara, es por eso que se plantea el control de alto nivel, dado que control a bajo nivel el lazo de realimentación viene dado por los diferentes sensores con los que cuenta el drone. Una vez que obtenemos el error de nuestro sistema este se multiplica por la estrategia de control que vayamos a implementar que para esta investigación se manejo el control clásico con un controlador PID y el control no convencional que en este caso es la lógica difusa.

3.2. Detección de objetos

3.2.1. Reconocimiento Facial

La detección de rostros es una solución de detección de rostros ultrarrápida con 6 puntos de referencia y admite múltiples rostros. Está basado en BlazeFace, que es un detector facial liviano con buen rendimiento, diseñado para inferencia de GPU móviles. El rendimiento súper en tiempo real del detector permite que se aplique a cualquier región facial precisa de interés como modelo para otras tareas específicas (como puntos clave faciales en 3D o estimación geométrica, clasificación de rasgos o expresiones faciales y segmentación de regiones faciales) . La red de extracción de características de magnitud está inspirada pero diferente de MobileNetV1 / V2, un detector de caja múltiple de disparo único con un esquema de anclaje compatible con GPU mejorado y una estrategia de resolución de enlace mejorada que reemplaza la supresión no máxima. BlazeFace. En el caso del código implementado se uso la librería de **CVZONE** para el reconocimiento facial, teniendo en cuenta que esta librería hace uso de Media Pipe con la misma finalidad.

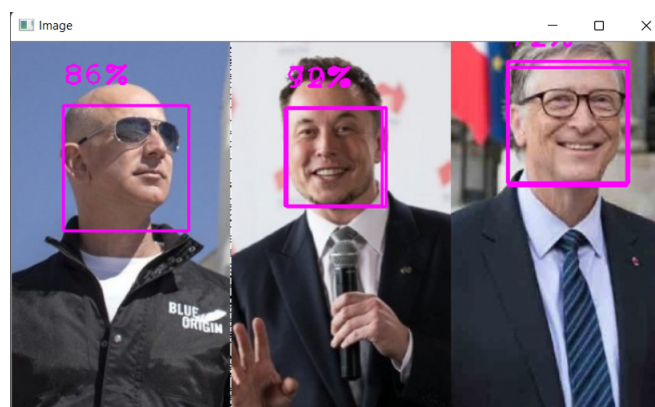


Figura 3.2: Reconocimiento Facial mediante el uso de Media Pipe - Fuente: Autor

Como podemos observar en la figura 3.2 el algoritmo de reconocimiento facial funciona de una manera eficiente, y cuenta con la capacidad de dibujar un bounding box en cada una de las caras que detecta, la cual sirve para obtener el área y las coordenadas en píxeles de la cara que reconoce, dado que como se menciona previamente estas variables serán funcionales en el momento de diseñar e implementar el controlador

3.2.2. Detección de Cuerpo

Dentro la estimación de pose humana a partir de la captura de un frame, se puede validar la importancia que tiene este sobre varias aplicaciones, como lo es la cuantificación de varios ejercicios físicos, el reconocimiento de lenguaje para personas sordo mudas y por ultimo el control de gestos del cuerpo, un ejemplo sencillo es la aplicación que puede tener con el yoga la danza o el fitness. Mediante el uso de Media Pipe Pose, se puede reconocer de una manera fiable la pose del cuerpo, los cuales cuentan con 33 puntos de referencia 3D sobre todo el cuerpo a partir de frames de vídeo en RGB.

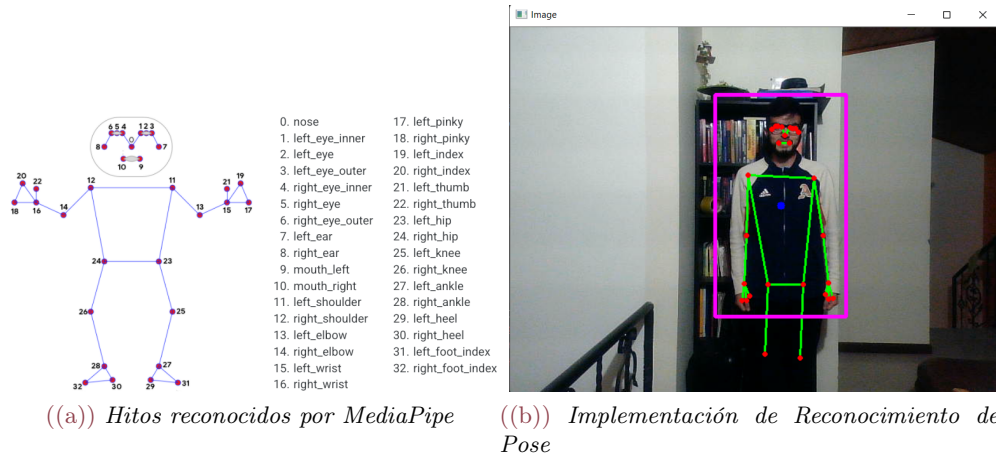


Figura 3.3: Reconocimiento de Pose - Fuente : Autor

De igual manera mediante el uso de la librería de **CVZONE** y teniendo en cuenta que esta librería se basa en la implementación de Media Pipe, se hace un reconocimiento de pose, y se dibuja un cuadro delimitador como podemos observar en la figura 3.3(b) para así de igual manera que con el reconocimiento facial obtener las diferentes variables de área y coordenadas en píxeles para así poder tener los valores de referencia para el diseño y simulación de las dos estrategias de control que se plantean.

3.3. Diseño y simulación de controlador PID

En una primera instancia se tiene que observar cuales y como son las variables que se van a controlar por medio de los recuadros delimitadores que se dibujaron, esto se desarrollo por partes como se muestra a continuación:

3.3.1. Área

Para el caso del movimiento de adelante y hacia atrás, se maneja con respecto al área del recuadro delimitador. Como se puede apreciar en la figura 3.6 a medida que el sujeto o la persona se acerca a la cámara del Dji Tello, este va dibujar un recuadro mucho mas grande, es decir que a medida que la persona se acerca el área del recuadro delimitador va ser mucho mayor y de manera inversa si este se aleja, es decir, al alejarse el área se disminuye. Dado que lo que prima es la seguridad del usuario, se busca implementar un controlador para que que el UAV no colisione con la cara del usuario y/o con el cuerpo del usuario por eso se plantean los movimientos que podemos observar en la 3.5 para el caso del área:

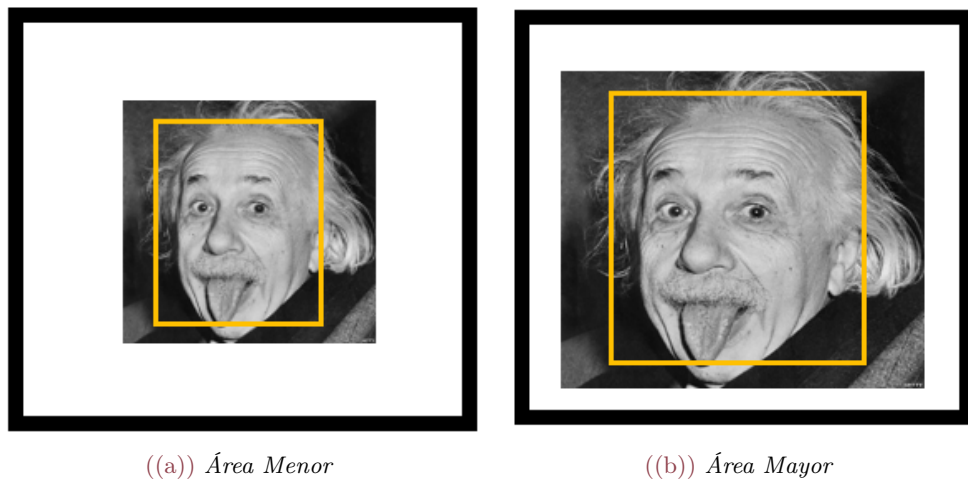


Figura 3.4: Interpretación del área del recuadro delimitador con respecto al Frame - Fuente: Autor

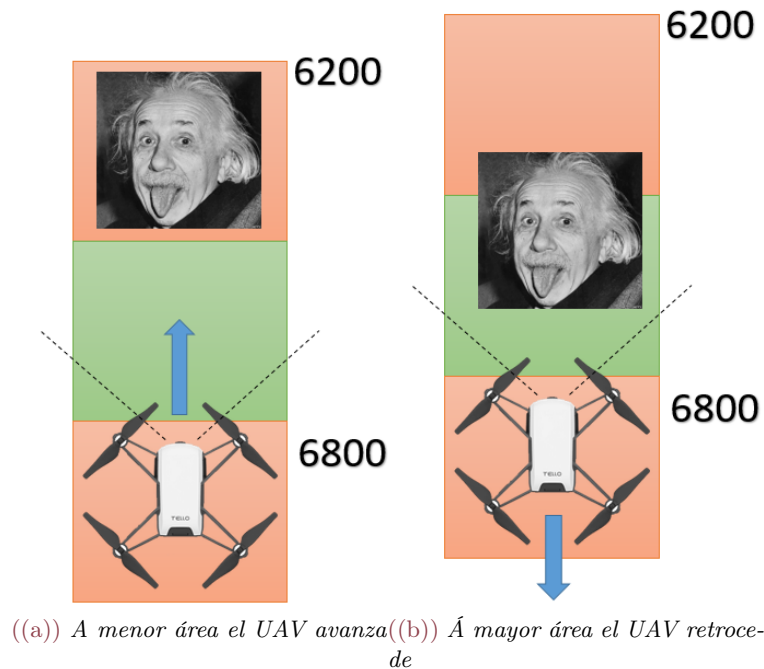


Figura 3.5: Movimientos que describe el UAV con respecto al área detectada - Fuente: Autor

3.3.2. Coordenadas de CX Y CY

Lo que se busca en el desarrollo de esta investigación fue que se hiciera de manera completa el tracking de una persona o la cara de una persona, es por eso que con respecto al movimiento en el eje X, se descarta y/o se reemplaza por el movimiento angular del Yaw con la finalidad que el UAV procure rotar con la cara del usuario. Para el control de dicho movimiento se utiliza la variable de C_x , y la variable de referencia va ser el ancho de la imagen del frame es decir, el frame o la imagen cuenta con un ancho y una alto, (h_i , w_i) para este caso la variable de referencia va ser $w_i // 2$, mientras que en el caso del que el UAV se mueva de manera ascendente o descendente, la variable de control va ser el alto de la imagen es decir $h_i // 2$ y con la variable de C_y para el lazo de realimentación del sistema.

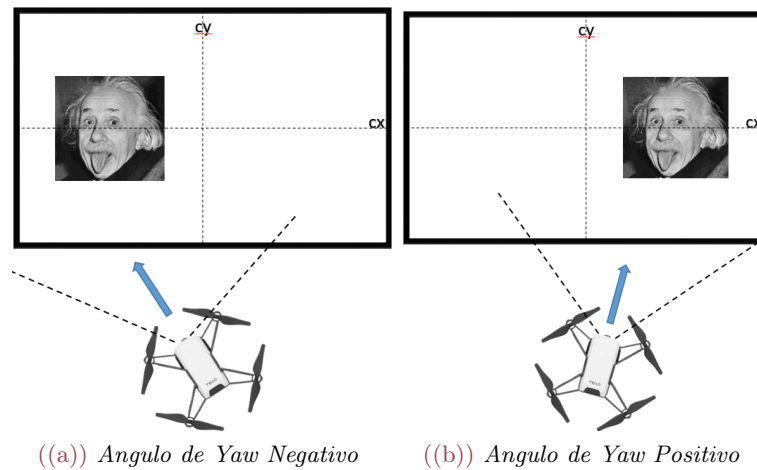


Figura 3.6: Movimientos que describe el UAV con respecto a la posición del Recuadro delimitador para el ángulo en Yaw - Fuente: Autor

3.3.3. Identificación del Sistema

Para el diseño de un controlador PID, se tiene que partir de la premisa de reconocer el sistema, es decir obtener la función de transferencia de la planta con la que se vaya a trabajar, teniendo en cuenta que como se planteo previamente se va hacer control en alto nivel mas no a bajo nivel dado que para un control a bajo nivel es necesario saber la caracterización matemática de todo el UAV para trabajar con respecto a los sensores del UAV, en esta investigación el único sensor con el que se trabajo fue el de la cámara es decir que el lazo de realimentación del sistema es la cámara dando los diferentes parámetros que se necesitan para diseñar cada uno de los controladores ,para ello por medio de datos experimentales y por medio del aplicativo de MATLAB *System identification Toolbox*, la metodología que se hizo para hallar la función de transferencia de cada uno de los movimientos se hizo mediante la respuesta al escalón. Para cada uno de los ejes el escalón es de diferente valor dado que como se menciona anteriormente cada movimiento cuenta con un controlador por separado.

Función de Transferencia para Yaw

Para hallar la función de transferencia para Yaw, se hizo la respuesta al paso con respecto al movimiento, es decir , un paso de 320 píxeles, teniendo en cuenta una posición inicial de 20 píxeles positivos, y se observo como el UAV hacia el movimiento hasta que este quedara en los 320 píxeles, y se obtuvo la siguiente gráfica.

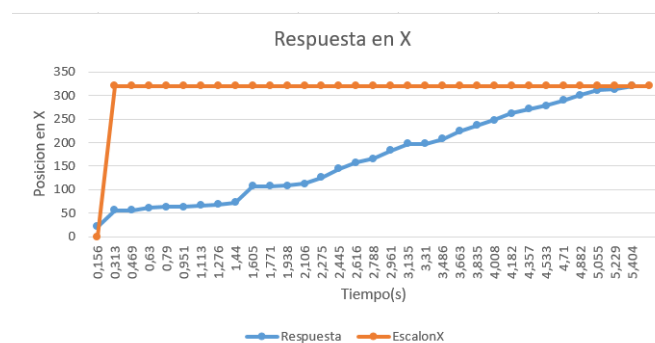


Figura 3.7: Respuesta al paso en Yaw - Fuente: Autor

Como se puede observar en la figura 3.7 se ve tanto el escalón de 320 píxeles como la respuesta del sistema la cual tuvo un retraso aproximado de 5 segundos, obteniendo estos datos, se uso el aplicativo de MATLAB de *System identification Toolbox*, para hacer el respectivo reconocimiento de el sistema, en el aplicativo se eligió la opción de Procesamiento de modelos haciendo que el sistema tuviera dos polos y ningún zero, para ello el porcentaje de similitud dio de un 94 % , a continuación se muestra una aproximación del sistema identificado.

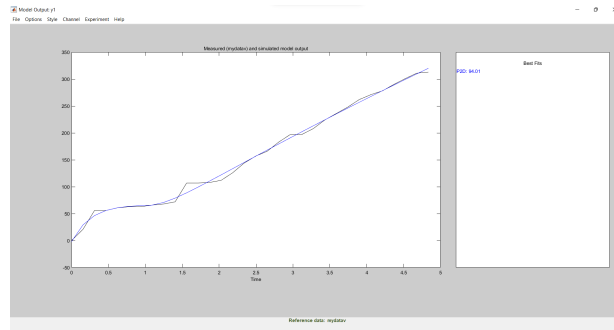


Figura 3.8: Identificación del Sistema con respecto al movimiento en Yaw - Fuente: Autor

Teniendo el sistema identificado se procede a sacar la función de transferencia, dado que este aplicativo ya cuenta con la opción de entregar la función de transferencia como resultado se obtuvo:

$$G(s) = \frac{5,88}{3,58s^2 + 24,02s + 1} \quad (3.4)$$

Contando con la función de transferencia se simulo mediante *simulink* si la respuesta del sistema era acorde, y dado que contamos con la opción de poner un bloque de control o un bloque en este caso de PID para observar las constantes que funcionen para este sistema o esta Planta, para este caso se procuro hacerlo por un ensayo y error con el fin de conseguir las constantes adecuadas para la planta.

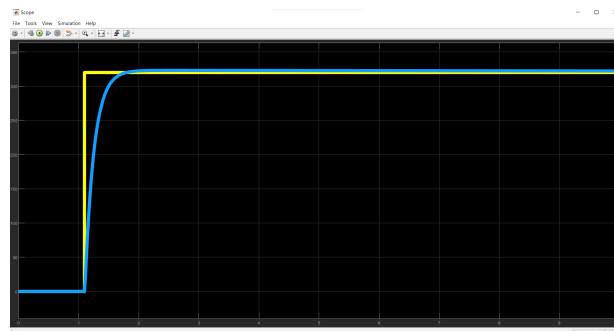


Figura 3.9: Respuesta del Sistema con las constantes - Fuente: Autor

Para observar la respuesta del sistema se procedió a utilizar las constantes para el PID para observar cuando el sistema es estable y cuanto tiempo se demora en estabilizarse, partiendo de la premisa que el sobre pico sea de un valor mínimo para que el sistema no se comporte de manera oscilante:

- $K_P = 0,17$
- $K_I = 0$
- $K_D = 0,1$

y la respuesta del sistema es la que podemos observar en la figura 3.9

Función de Transferencia para el movimiento de Subir y Bajar

Para hallar la función de transferencia para subir y bajar el drone dependiendo de la posición del recuadro delimitador, de igual manera se hizo la respuesta al paso con respecto al movimiento pero para este caso se tuvo en cuenta que el paso fuera equivalente al valor de $h_i/2$ es decir de 240 píxeles, claro esta que se hizo teniendo en cuenta una posición inicial de 380 píxeles, y se observo como el UAV hacia el movimiento hasta que este quedara en los 240 píxeles, y se obtuvo la siguiente gráfica.

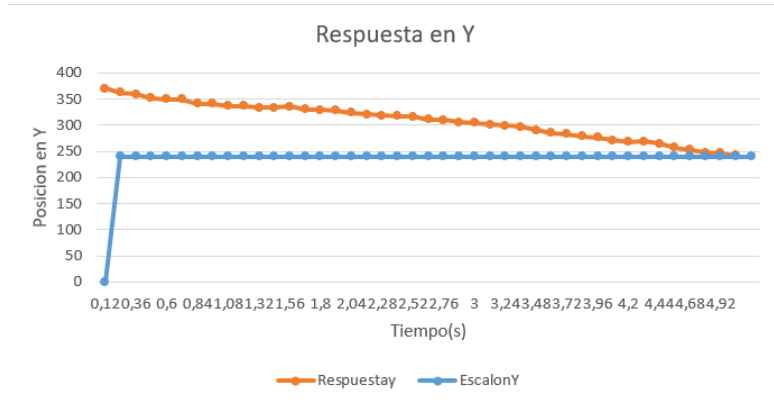


Figura 3.10: Respuesta al paso con respecto al movimiento de subir y bajar el UAV - Fuente: Autor

De igual manera se aplico la misma metodología con respecto al movimiento anterior es decir al Yaw, pero en esta gráfica se hizo la respuesta fuera de manera descendente es decir desde un valor mayor hacia un valor menor en la figura 3.10 se ve tanto el escalón de 240 píxeles como la respuesta del sistema la cual tuvo el mismo tiempo de retraso con respecto al movimiento anterior, obteniendo estos datos, de igual manera se uso el aplicativo de MATLAB de *System identification Toolbox*, para hacer el respectivo reconocimiento de el sistema, para este caso de igual manera en el aplicativo se eligió la opción de Procesamiento de modelos haciendo que el sistema tuviera dos polos y ningún cero, para ello el porcentaje de similitud dio de manera similar en este caso se obtuvo un 87% , a continuación se muestra una aproximación del sistema identificado.

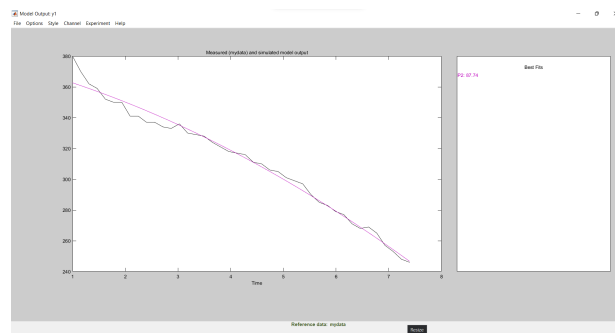


Figura 3.11: Identificación del Sistema con respecto al movimiento de subir y bajar con el UAV - Fuente: Autor

De igual manera ya que se identifica el sistema se puede obtener la función de transferencia para este movimiento.

$$G(s) = \frac{5,18}{5,49s^2 + 24,05s + 1} \quad (3.5)$$

Podemos observar que la función de transferencia es similar de subida y bajada del UAV tiene cierta similitud, esto es normal por la cercanía en valor del escalón, además que recordando la velocidad máxima que puede describir el drone mediante el SDK de python es de -100 a 100 por

medio del comando de *send rc control(self, left right velocity, forward backward velocity, up down velocity, yaw velocity)*. de igual manera siguiendo la metodología de el movimiento anterior procedemos a simular la respuesta al escalón teniendo en cuenta las constantes del PID obteniendo como respuesta la siguiente gráfica. Para observar la respuesta del sistema se procedió

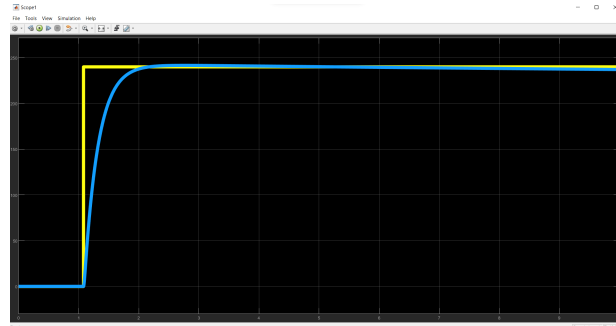


Figura 3.12: Respuesta del Sistema con las constantes - Fuente: Autor

a utilizar las constantes para el PID para observar cuando el sistema es estable y cuanto tiempo se demora en estabilizarse, partiendo de la premisa que el sobre pico sea de un valor mínimo para que el sistema no se comporte de manera oscilante:

- $K_P = 0,27$
- $K_I = 0$
- $K_D = 0,1$

Función de Transferencia para el Área

Para la identificación de la función de transferencia se tuvo en cuenta en este caso la respuesta al paso pero el valor del paso es de 12000 píxeles que el valor objetivo a controlar, de igual manera se obtuvo la siguiente gráfica, cabe resaltar que para el caso de reconocimiento de pose, es una misma función de transferencia pero tratando de conservar la misma relación con esta función de transferencia dado que tanto para el reconocimiento facial, como para el detector de pose se limita la velocidad. Como se puede observar en la figura 3.16 , se puede observar el comportamiento del

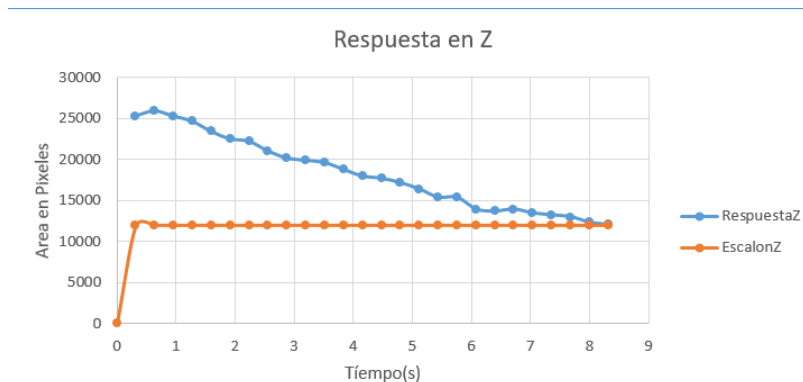


Figura 3.13: Respuesta al paso con respecto al movimiento de alejar y acercar con respecto al área del recuadro delimitador el UAV - Fuente: Autor

UAV con respecto al escalón, cabe resaltar que este controlador se tiene que hacer con mas cautela dado que este movimiento si podría perjudicar al usuario, y como se dijo previamente lo que prima es la seguridad del usuario por ende el este movimiento se limita la velocidad a (20,-15) teniendo

en cuenta que se sacrifica la velocidad de respuesta del sistema pero se asegura la integridad del usuario. Luego de tener los datos de igual manera nos dirigimos al *System identification Toolbox* para subir cada uno de los datos que son necesarios para la identificación del sistema, teniendo en cuenta que este tiene un tiempo de muestreo muchísimo mayor ya que como se menciono previamente se limito la velocidad de los motores es decir el tiempo de respuesta va ser mas lento, y para esta toma de datos el UAV tardo al rededor de 8 segundos en completar el escalón, para esta toma de datos se identifico mediante el procesador de modelos, teniendo en cuenta una misma estructura que las anteriores funciones de transferencia, obteniendo un porcentaje de llenado del 92.66 %

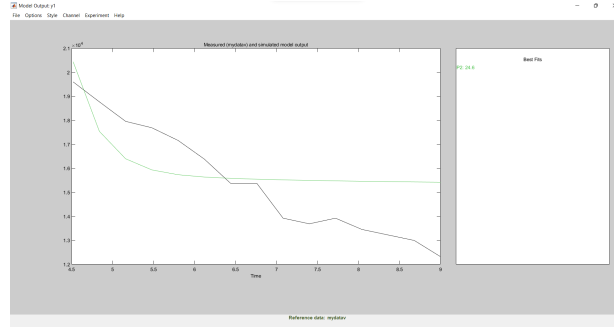


Figura 3.14: Identificación del Sistema con respecto al movimiento de acercarse y alejarse con el UAV

Así finalmente se obtuvo la siguiente función de transferencia:

$$G(s) = \frac{1,2776}{0,8871s^2 + 2,9849s + 1} \quad (3.6)$$

Luego de obtener la función de transferencia se observo su comportamiento mediante *simulink* y con la ayuda del bloque de PID se simularon las constantes a tener en cuenta para el controlador, teniendo en cuenta que el controlador va funcionar con un tiempo de respuesta mucho menor que el de los otros dos controladores que se diseñaron previamente.

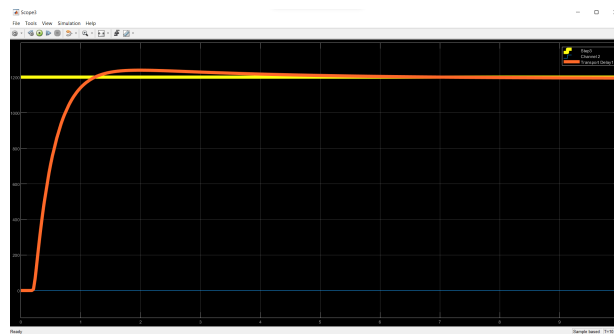


Figura 3.15: Respuesta del Sistema con las constantes

Para observar la respuesta del sistema se procedió a utilizar las constantes para el PID para observar cuando el sistema es estable y cuanto tiempo se demora en estabilizarse, partiendo de la premisa que el sobre pico sea de un valor mínimo para que el sistema no se comporte de manera oscilante:

- $K_P = 0,005$
- $K_I = 0$
- $K_D = 0,0013$

Para el caso del área con respecto al detector de pose dado que como se menciono anteriormente el área es mucho mayor con respecto al recuadro delimitador también se tiene que hacer el reconocimiento de la planta aplicando la misma dinámica que se hizo previamente. Esto se hace dado que también para dicho programa el valor objetivo es bastante diferente. Como se puede observar se hizo la respuesta al paso, teniendo como valor objetivo 150000 píxeles obteniendo así:

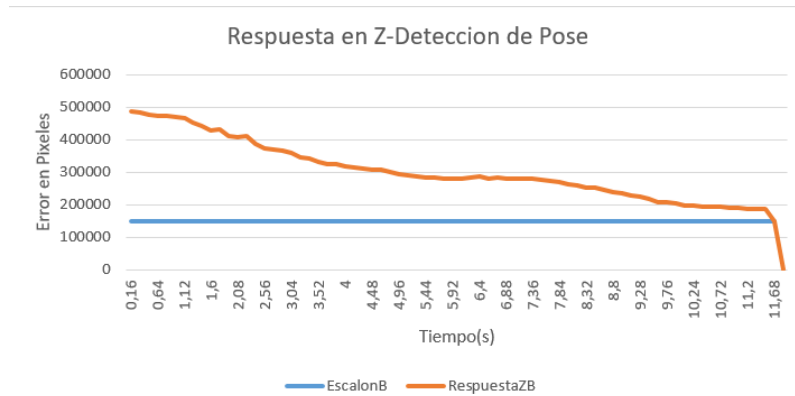


Figura 3.16: Respuesta al paso con respecto al movimiento de alejar y acercar con respecto al área del recuadro delimitador de la detección de Pose

De igual manera mediante el *System identification Toolbox* se identifico el sistema obteniendo un porcentaje de similitud de un 90 % Obteniendo la siguiente función de transferencia y una

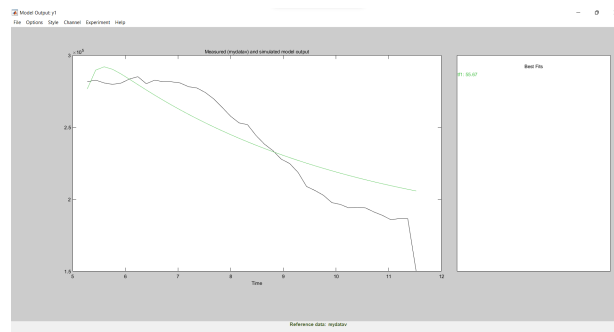


Figura 3.17: Identificación del Sistema con respecto al movimiento de acercarse y alejarse con el UAV

respuesta al escalón con las siguientes constantes:

$$G(s) = \frac{21,42s + 1,566}{s^2 + 6,216s + 1,353} \quad (3.7)$$

- $K_P = 0,00016$
- $K_I = 0$
- $K_D = 0,000011$

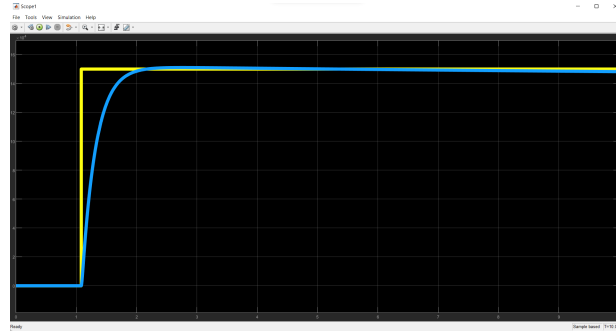


Figura 3.18: Respuesta del Sistema con las constantes

3.4. Diseño y simulación de controlador Lógica Difusa

Como se planteó en el desarrollo de esta investigación, se busca contrastar el control clásico con el control no convencional, a lo largo de esta sección se mostrara el proceso por el cual se hizo el diseño y simulación de un controlador por lógica difusa.

Para el diseño del controlador difuso se uso de la librería **Fuzzy.py** dado que esta cuenta con la expresión matemática de cada unas de las funciones de membresia que existen y que se pueden simular y graficar en python, cabe resaltar que se decidió trabajar por medio de Python dado que en este software es mucho mas fácil hacer procesamiento de imágenes, reconocimiento facial y el reconocimiento de pose del usuario como ya se visualizo previamente, cabe resaltar que el metodo que se uso para las reglas difusas se hizo mediante **Mamdani**. En una primera etapa se definió el universo de discurso para el error, observando cuanto es el error máximo que puede haber dentro del sistema, para cada movimiento del que describe el UAV existen diferentes margenes de error es decir cuenta con un diferente grupo de funciones de membresia. Para el error se trabajo con las siguientes variables lingüísticas:

- ENG : *Error negativo grande*
- ENP : *Error negativo pqueño*
- EC : *Error cero*
- EPP : *Error positivo pequeño*
- EPG : *Error positivo Grande*

con estas mismas variables lingüísticas se trabajo para cada uno de los movimientos, y de igual manera la respuesta del sistema es decir un universo de velocidad donde las variables lingüísticas son:

- VNG : *Velocidad negativa grande*
- VNP : *Velocidad negativa pequeña*
- VC : *Velocidad cero*
- VPP : *Velocidad positiva pequeña*
- VPG : *Velocidad positiva grande*

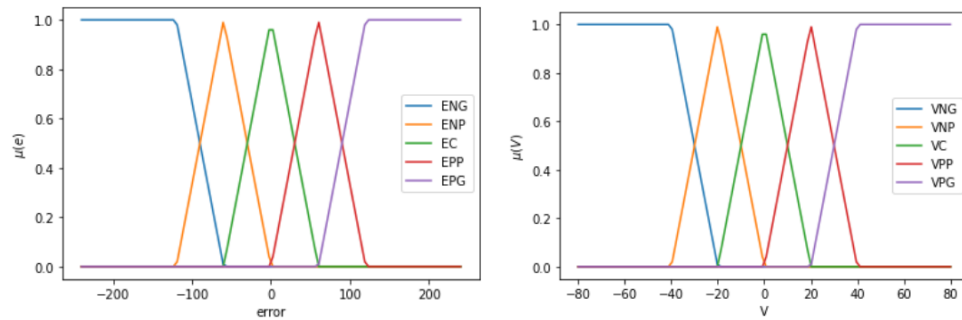
El conjunto de reglas como se dijo anterior mente se trabajaron por el método de Mamdani, y para cada conjunto difuso se implementaron las mismas reglas, las reglas que se usaron se muestran a continuación:

- si el error negativo es grande, entonces la velocidad es negativa grande.
- si es error negativo pequeño, entonces la velocidad es negativa pequeña.
- si el error es cero, entonces la velocidad es cero
- si el error es positivo pequeño, entonces el voltaje es positivo pequeño
- si el error es positivo grande, entonces el voltaje es positivo grande

3.4.1. Funciones de Membresia para Yaw

Mediante el uso de la librería y el uso de **Jupyter.Lab** podemos simular los universos y las diferentes variables lingüísticas, obteniendo tanto las funciones como los rangos con la que cuenta cada una de las funciones de memebresia, Para el caso del movimiento en Yaw se obtuvo un error máximo de 250 píxeles definiendo así el universo y cada una de las variables lingüísticas. De igual manera para el caso de la velocidad en este caso se definió un universo de de velocidad máxima de 80, es decir la velocidad ira de -80 hasta 80 dado que el UAV no alcanza su velocidad máxima que es 100, por ende se definió este universo. Cabe resaltar que las funciones que se usaron tanto para el error como para la velocidad fueron:

- ENG - VNG = *Función trapezoidal*
- ENP - VNP = *Función triangular*
- EC - VC = *Función triangular*
- EPP - VPP = *Función triangular*
- EPG - VPG = *Función trapezoidal*



((a)) Funciones de membresia para el Error en Yaw ((b)) Funciones de Membresia para la velocidad en Yaw

Figura 3.19: Funciones de membresia tanto para el error como para la velocidad en Yaw - Fuente: Autor

3.4.2. Funciones de Membresia para el eje Y

De la misma manera que se hizo con Yaw se diseño para el eje Y cabe resaltar que se hizo un controlador difuso independiente para cada uno de los movimiento que describe el UAV, para este caso el universo del error fue de 190 como máximo, y la velocidad se manejo de la misma manera es decir de 80 a -80, con los mismos tipos de curva, el error se tomo con respecto a la variable de control es decir con respecto al Cy del recuadro delimitador, cabe resaltar que lo que diferencia

cada uno de los controladores difusos es la manera de defuzzificar, dado que existen diferentes métodos como se pudo apreciar en el marco teórico, además teniendo en cuenta que todas estas se hacen por medio del método de inferencia.

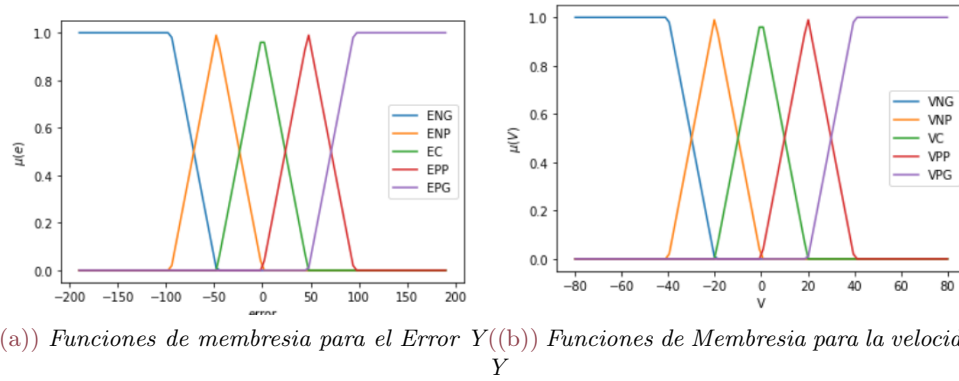
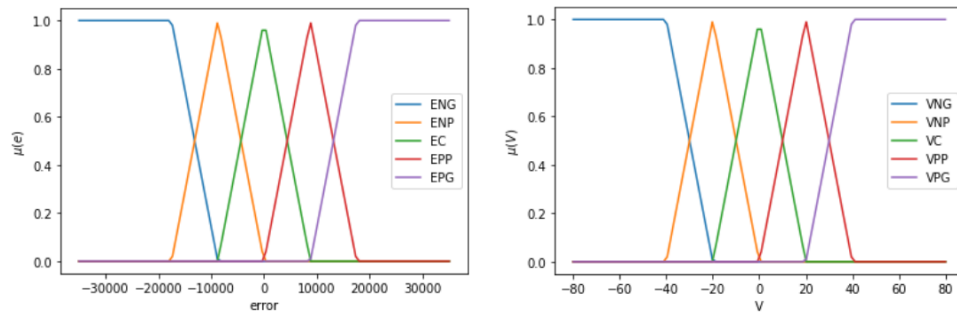


Figura 3.20: Funciones de membresia tanto para el error como para la velocidad en Y - Fuente: Autor

3.4.3. Funciones de Membresia para el Área

Para el caso del área partiendo de la misma premisa de garantizar la seguridad del usuario el universo de la velocidad cambia y en este caso solo la ponemos de 20 a -20, en el caso del error del área se propone que el máximo error que este puede obtener es de aproximadamente 35000 cabe resaltar que en el caso de body following esta función de membresia si cambia dado que el área del recuadro delimitador es mucho mayor en el caso del detector de pose que en el de detector facial, es decir, que son universos diferentes pero que cuentan con las mismas variables lingüísticas en el caso de body following se tomo un error máximo de 350000 píxeles, pero en el caso de las funciones de membresia de la velocidad si se conservo la misma relación



((a)) Funciones de membresia para el error con respecto al área del recuadro delimitador de Face Following ((b)) Funciones de Membresia para la velocidad

Figura 3.21: Funciones de membresia tanto para el error como para la velocidad en Z - Fuente: Autor

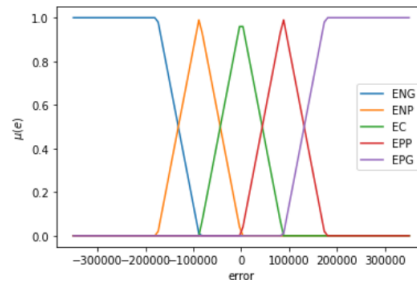


Figura 3.22: Funciones de membresia para el error con respecto al área del recuadro delimitador de Body Following - Fuente: Autor

3.4.4. Método de inferencia

Para encontrar el valor con respecto a las reglas difusas, se hizo por el método de inferencia, como se puede mostrar en la figura 3.23 esta es la lógica que contempla el método de inferencia para un sistema de tipo mamdani teniendo en cuenta que para el método de inferencia con sistemas de tipo mandami se contempla el valor puntual que se ingresa en el sistema difuso que en este caso es el error actual, para así luego poder convertir este en un nivel de pertenencia para cada regla. A cada una de las reglas el operador difuso respectivo se aplica al valor de membresía asociado con la entrada. Utilizando los resultados anteriores y las respectivas funciones de pertenencia de la salida asociada a la regla, se aplica el operador difuso correspondiente al método de implicación. En seguida mediante la implementación de un operador difuso se incluyen todas las implicaciones, para que finalmente mediante un método de defusificación se obtenga un resultado [4]

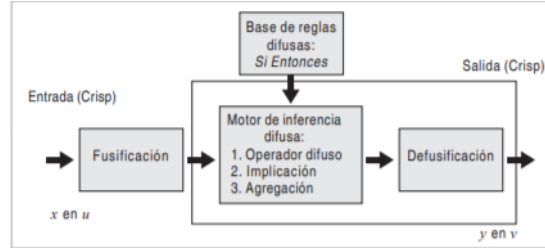


Figura 3.23: Lógica que se maneja en la inferencia difusa Fuente: [4]

Para observar el comportamiento del método de inferencia se procedió hacer una simulación por medio de Jupyter notebook donde se puede observar tanto las funciones de membresía del error y de la velocidad, y por medio del ingreso de un valor de error actual se puede obtener y visualizar como el programa hace el método de inferencia para este sistema de tipo mandami, como se muestra a continuación. Para la simulación se hizo mediante el universo del error para Face Following, en el caso de el movimiento de Yaw y se ingreso un error actual de 75 pixeles de error, en la simulación se puede apreciar como el programa fuzzifica es decir encuentra el valor de pertenencia para el valor de error actual que para este caso es de 75 pixeles, y se visualiza la posición de los valores de pertenencia además de la grafica del método de inferencia como se muestra a continuación

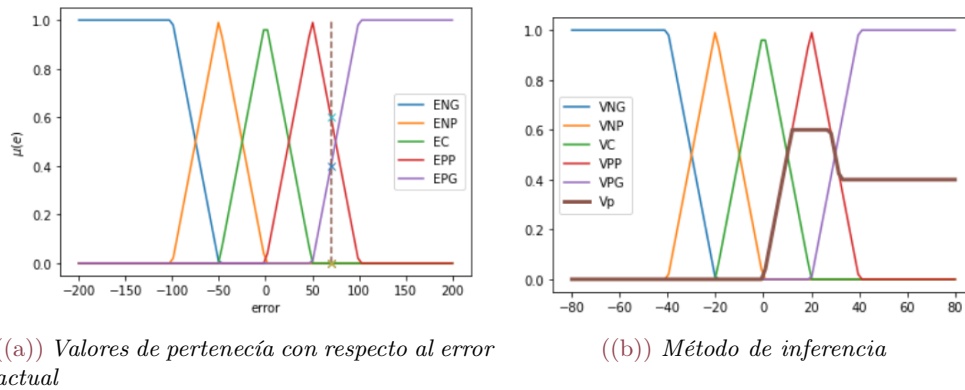


Figura 3.24: Gráfica de Valor de pertenencia y método de inferencia - Fuente: Autor

Los valores de pertenencia que se obtuvieron fueron:

- val ENG = 2.2479472311238644e-12
- val ENP = 0.0
- val EC = 0.0
- val EPP = 0.6
- val EPG = 0.4

3.4.5. Defuzzificación

Para el método de defuzzificación se busco un método por el cual el comportamiento fuera lineal dado que existen diferentes tipos de defuzzificación, previamente se hace un análisis de donde están ubicados los valores de defuzzificación teniendo en cuenta el ejemplo anterior para el método de inferencia, y gracias al software de igual manera podemos simular todos los tipos

de defuzzificación y mirar cual es el valor que mas se ajusta al controlador, además de visualizar cuales son los valores que ese obtienen con los distintos tipos de defuzzificar.

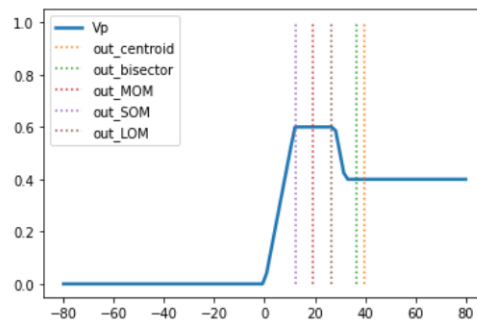


Figura 3.25: Gráfica de defuzzificación - Fuente: Autor

los valores que se obtuvieron con respecto a la figura 3.25 fueron:

- out centroid = 39.77899293653051
- out bisector = 36.36363636363636
- out MOM = 19.393939393939394
- out SOM = 12.121212121212125
- out LOM = 26.666666666666667

el valor que se selecciona depende de como este se comporte con respecto al control, es decir observando de que manera se ajusta mejor para disminuir el erro de una manera mucho mejor.

Capítulo 4

Análisis de Resultados

Para el análisis de los resultados se hicieron mediante la elaboración de gráficas de error , y en adición viendo superpuesta en la misma gráfica, la acción de control con respecto al error actual que captura el sensor que como se dijo previamente es la cámara dado que se esta hablando de control de alto nivel. A continuación se va observar tanto la gráfica que describió el UAV mediante control clásico es decir PID, como la gráfica con control no convencional por lógica difusa. Todas las pruebas que se hicieron, se aplicaron con un rango de tiempo de al rededor de 25 segundos, dado que la respuesta del UAV es de manera casi inmediata. Cabe resaltar que cada gráfica tiene una respuesta diferente, dado que se hicieron las pruebas de manera aleatoria es decir observar el comportamiento sin una ruta en especifico, de igual manera el controlador funciona de manera uniforme cual sea el movimiento por eso se planteo de esta manera.

4.0.1. Gráfica de error y acción de control en Yaw

PID Yaw

Para la obtención de las gráficas se hizo mediante manera practica, exportando los datos de python a excel, esto se hace mediante la implementación de la librería de Pandas importando dicha librería, para luego crear un dataframe de los datos dentro de python para luego poder visualizar los datos y poder contrastar tanto el control clásico como el control no convencional. Como se puede observar en la figura 4.1 se puede observar que el control obedece de una manera efectiva el comportamiento del UAV con respecto a la trayectoria que describe, cabe resaltar que estos experimentos se elaboraron sin tener en cuenta una ruta en especifico, por ende la irregularidad de las gráficas, pero se puede observar que para este experimento se movió de izquierda a derecha, por eso es que la curva describe una función similar a una función oscilante. Teniendo en cuenta esta gráfica podemos observar que el tiempo de respuesta es casi que inmediato, pero si se desea que el sistema opere de una manera mucho mas rápida se tiene que variar la constante integral del controlador

Control Difuso Yaw

Como se puede observar en la figura 4.2 para el control difuso se observa un comportamiento similar pero en este caso es un tanto mas lenta la respuesta, dado que el universo del para la velocidad es un poco mas pequeño es decir que va desde -80 hasta 80 y esto hace que la respuesta del sistema sea un poco tardía, además que influye mucho el método de defusificación que se este implementando, para esta salida se eligió el método de bisección dado que era el que mejor respondía con el sistema , pero se puede evidenciar que en el caso del control difuso sirve de una manera lineal, es decir cuando el UAV detecta un error cercano al 0 este envía al UAV una velocidad de 0 (cm/s) cabe resaltar que el segundo eje horizontal viene expresado en cm/s es decir la velocidad, para verificar que el comportamiento del controlador sea el óptimo por medio de las gráficas se puede observar que las curvas son similares y cuentan con una misma forma de onda, cabe resaltar que las pruebas se hicieron en un ambiente controlado donde no hay perturbaciones de ninguna

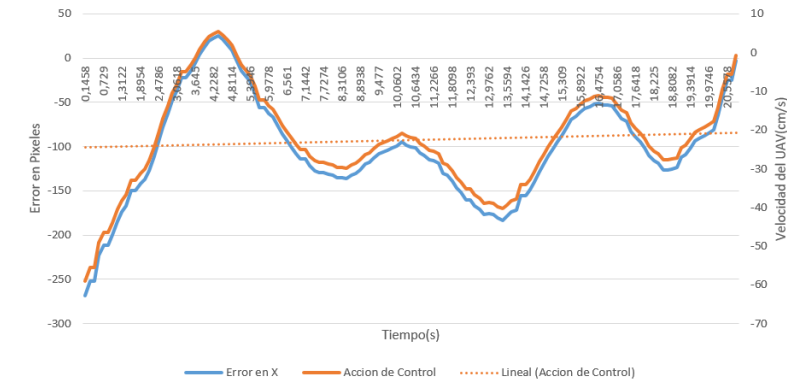


Figura 4.1: Gráfica de Respuesta del error y la acción de control por medio de Control Clásico - Fuente: Autor

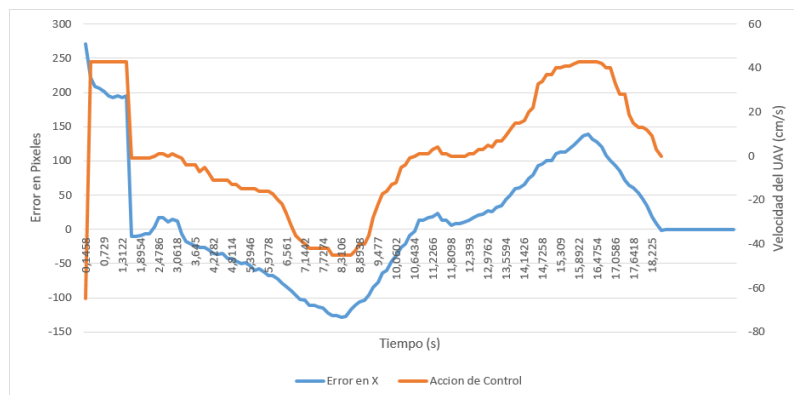


Figura 4.2: Gráfica de Respuesta del error y la acción de control por medio de Control no convencional - Fuente: Autor

clase. Esta gráfica corresponde tanto para el reconocimiento facial como para el reconocimiento de pose, dado que en los dos métodos se hizo con las mismas funciones de membresía y el mismo controlador PID dado que cuentan con el mismo valor objetivo para Yaw y para el movimiento ascendente y descendente del UAV, el único movimiento que es diferente es el de alejar y acercar dado que el área del recuadro delimitador si es mucho mayor y/o menor

4.0.2. Gráfica de error y acción de control para el eje Y

De igual manera para la gráfica de error y acción de control con respecto al eje Y, se manejo con el mismo tiempo de muestreo que fueron al rededor de 24 segundos, hay que tener en cuenta que se busco que el comportamiento de la acción de control fuera de manera oscilatoria es decir en este caso subiendo y bajando la cara y/o el cuerpo haciendo que el centro del recuadro delimitador creara un error positivo y negativo para así poder observar como se comportaron las estrategias de control

PID Eje Y

Como se puede observar en la figura 4.4 en la parte izquierda se puede apreciar la curva de control que describe el UAV con respecto al error, mediante el método de control clásico se ve que hay una respuesta de manera óptima obteniendo errores pequeños, para este experimento se observo que el comportamiento del UAV es un poco lento dado que las condiciones de reconocimiento en el momento de tomar el experimento eran un poco opacas esta, cabe resaltar que la

respuesta del sistema podría ser mucho mas eficiente teniendo en cuenta un ambiente controlado mucho mas optimizado para la prueba. Pero los resultados obtenidos que se pueden observar son óptimos dado que la curva de acción de control corresponde de igual manera a la curva del error

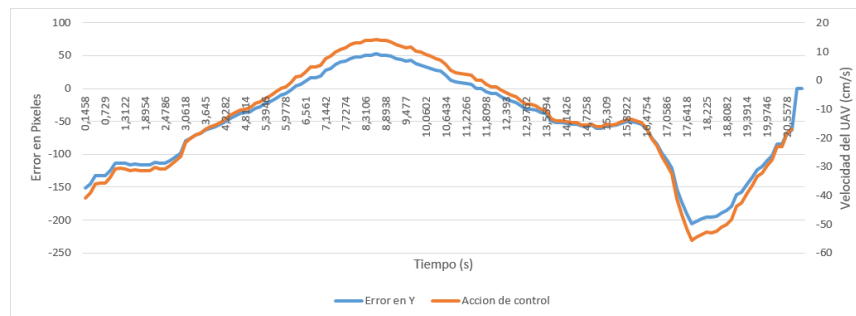


Figura 4.3: Gráfica de Respuesta del error y la acción de control por medio de Control Clásico - Fuente: Autor

Control Difuso Eje Y

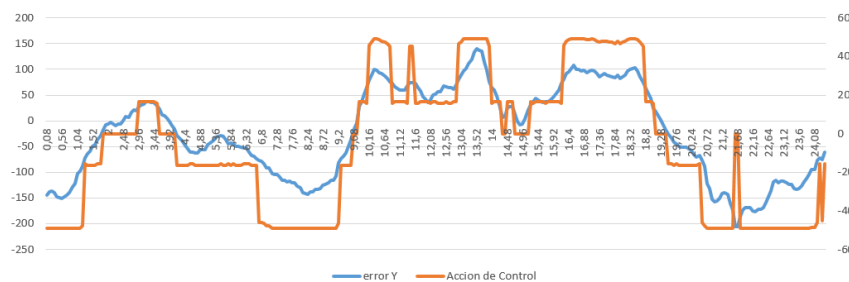


Figura 4.4: Gráfica de Respuesta del error y la acción de control por medio de Control Clásico - Fuente: Autor

Por el lado de el controlador difuso se ve varios margenes de error pero, vemos que la curva que describe la acción de control corresponde al comportamiento del error, esta gráfica se ve de manera escalonada ya que se aplico el método de defuzzificacion de **Método de la media de Máximo** y en la figura 2.11 se puede observar el tipo de curva que describe este método , se eligió este método ,dado que en el caso del movimiento de ascender y descender esta fue la mejor opción respecto a las funciones de membresia que se plantearon, cabe resaltar que el movimiento no sufre de cambios tan bruscos pero se puede acondicionar de una mejor manera

4.0.3. Gráfica de error y acción de control para el eje Z

Reconocimiento de Pose

Para el eje Z se cuenta con una característica en particular , para este caso si se observo tanto el comportamiento con el reconocimiento facial, como el reconocimiento de pose, implementando de igual manera las dos estrategias de control y observar cual es su comportamiento, dado que estos dos métodos de reconocimiento de objetos y/o personas son diferentes con respecto al debido a que el recuadro delimitador cuenta con un área mucho mayor para el caso del detector de pose. Para este experimento se hicieron mediante diferentes constantes en la parte del control clásico Para el caso del control difuso, se desarrollo por medio de el método de bisección, dado que por este método se pudo obtener una mejor respuesta del sistema de control difuso. A continuación se muestra la respuesta del sistema con respecto a la detección de pose, como se dijo previamente

la ruta para la obtención de datos no se tuvo en cuenta por ende la gráfica de error es diferente pero, de igual manera podemos observar el comportamiento de los controladores al someterlos a diferentes pruebas como se muestra a continuación.

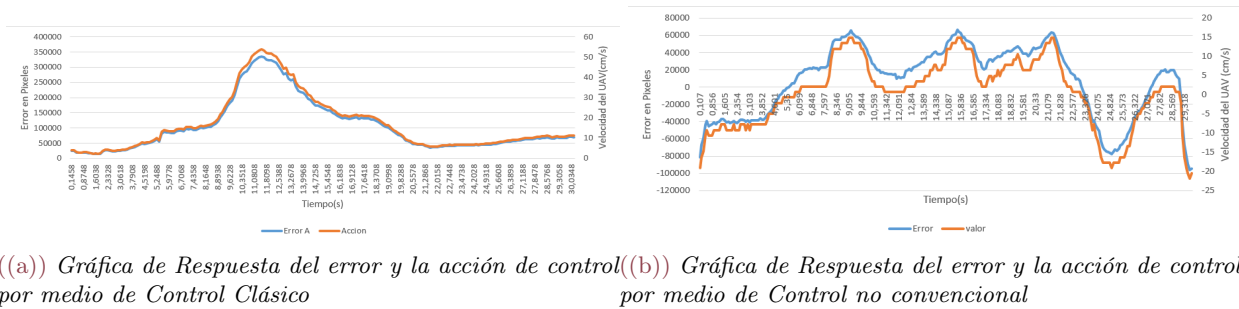


Figura 4.5: Gráfica de Respuesta del sistema mediante dos estrategias de control - Fuente: Autor

Reconocimiento Facial

Por otro lado el reconocimiento facial funciona de una manera menos eficiente, esto se debe al entorno y a la iluminación para el reconocimiento facial dado que se hace por medio de hardcascaade, para mejorar la identificación del reconocimiento facial se recomienda la implementación de una Red Neuronal como YoLo para que el reconocimiento sea de una manera mas eficiente y el control por ende va funcionar de manera mas eficiente, en este caso se obtuvo una gráfica con el controlador clásico con bastante ruido, pero que aun así funciona el controlador, y por parte del control no convencional, se encuentra una mejora frente a la respuesta del sistema como se puede observar a continuación:

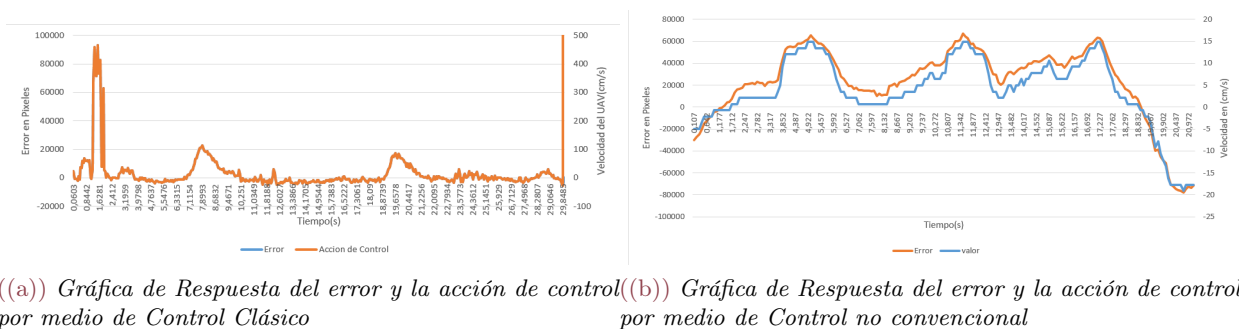


Figura 4.6: Gráfica de Respuesta del sistema mediante dos estrategias de control - Fuente: Autor

Finalmente podemos observar en la figura 4.6(a) que el control sirve de manera muy eficiente dado que la curva de control esta superpuesta con la gráfica del error, pero se cuenta con una falencia en cuestiones de ruido dado que el sistema reconoció un área muy grande y tenemos un sobre pico demasiado alto, lo que puede ocasionar que el UAV pierda la referencia del reconocimiento facial, y por ende tener una respuesta un tanto lenta, pero en este caso la velocidad de respuesta se puede solucionar aumentando los rangos de velocidad para el movimiento en Z.

4.0.4. Conclusiones

Como conclusión general se pudo observar el comportamiento de las dos estrategias de control tanto clásico y no convencional para el seguimiento de objetos, se observo que el seguimiento funciona de una mejor manera para el caso de seguimiento de personas dado que el recuadro delimitar no genera tanto erro, y de igual manera que la implementación de control por lógica difusa es mucho mas fácil e intuitivo para plantas que no sea tan fácil hacer el modela miento matemático de la misma. En su medida los objetivos específicos de igual manera también se cumplieron, dado que estos se formularon como los principales pasos a tener en cuenta para

cumplir el objetivo principal. En adición se cumplió con el objetivo de comunicar el UAV mediante el software de Python para hacer el respectivo reconocimiento facial y el reconocimiento de pose para obtener así los recuadros delimitadores que brindaron las diferentes variables para hacer el control, cabe resaltar que para trabajos futuros se recomienda hacer un control a bajo nivel, es decir, mediante el uso de los diferentes sensores que cuenta el DJI Tello, Además de implementar diferentes funciones de membresía y diferentes tipos de Fuzzyficacion y Defuzzificacion, para así obtener una respuesta más óptima. además una conclusión en general, se pudo observar que el control clásico funciona de una manera un tanto más eficiente que el control no convencional pero la principal ventaja que cuenta el control no convencional frente al control clásico es que el control no convencional se puede diseñar de manera mucho más intuitiva sin tener la necesidad de contar con el modelo matemático de la planta, es decir, que no se necesita ser un gran experto en la materia de control, si no tener claras Cuáles van a ser las variables lingüísticas, el universo, y las diferentes reglas de justificación para implementar el sistema, además, implementar otro tipo de conjunto de reglas difusas es decir Takagi Sugeno, dado que en esta investigación se trabajó con Mamdani. por último también se concluyó que la iluminación reconoció reconocimiento facial, y qué este influye a la hora de aplicar una estrategia de control

Apéndice **A**

Participación en eventos

A.1. V Encuentro Zonal de Semilleros de Investigación

El 23 de abril del 2021 se presentó la ponencia nombrada igual que la presente investigación, en la que se presentaron avances de la primer etapa del proyecto, recibiendo un concepto positivo por parte de los evaluadores, recomendando a su vez tener claras las categorías de análisis y los referentes que las acompañan.

A.2. VI Congreso Internacional de Instrumentación, Control y Telecomunicaciones

Ponencia presentada el 19 de octubre en el marco del VI Congreso Internacional de Instrumentación, Control y Telecomunicaciones - III Congreso internacional en diseño, fabricación y nuevos materiales donde se presentó el desarrollo del proyecto hasta la fecha.



Figura A.1: *Certificado de participación*

Bibliografía

- [1] E. D. N. Guerrero and F. A. V. De La Torre, “Desarrollo de un modelo matemático, cinemático y dinámico con la aplicación de software, para modificar el funcionamiento de un dron, para que este realice monitoreo automático,” *RECIMUNDO*, vol. 4, no. 1 (Esp), pp. 332–343, 2020.
- [2] S. Kouro and R. Musalem, “Control mediante lógica difusa,” *Técnicas Modernas en Automática*, vol. 1, pp. 1–7, 2002.
- [3] J. C. García Infante, *Sistemas con lógica difusa*, 2009.
- [4] H. E. E. Cuchango and J. J. S. Méndez, “Sistema de inferencia difusa basado en relaciones booleanas,” *Ingeniería*, vol. 15, no. 2, 2010.
- [5] C. Guerra. (2018) Logica difusa y sistemas de control. [Online]. Available: <https://cayetanoguerra.github.io/ia/logicadifusa/Logica%20difusa%20y%20sistemas%20de%20control.pdf>
- [6] E. Rivera Guerrero *et al.*, “Control de un drone por seguimiento de objetos,” thesis, Uniandes, 2016.
- [7] O. Escobar Díaz, O. ESCOBAR DIAZ *et al.*, “Seguimiento de objetos con vehículos aéreos no tripulados,” Master’s thesis, 2015.
- [8] A. Gauthier, *El regulador pid industrial*, SicuaPlus, Ed., 2015.
- [9] T. del Cerro Sánchez and P. N. Laina, “Logica difusa,” *Universidad Carlos III de Madrid*, vol. 8, 2014.
- [10] B. MARTIN and A. SANZ MOLINA, *Redes neuronales y sistemas difusos*, 2002.
- [11] H. Ying, “The simplest fuzzy controllers using different inference methods are different non-linear proportional-integral controllers with variable gains,” *Automatica*, vol. 29, no. 6, pp. 1579–1589, 1993.
- [12] M. A. P. Jacques, M. Pursula, J. Niittymäki, and I. Kosonen, “The impact of different approximate reasoning methods on fuzzy signal controllers,” *Anais do 13th Mini-Euroconference*, vol. 10, pp. 184–192, 2002.
- [13] P. A. Revenga de Toro *et al.*, “Diseño y construcción de un dron para el seguimiento de un objeto de forma autónoma mediante visión artificial,” 2017.
- [14] R. Rico Díaz *et al.*, “Seguimiento automático de objetivos con drones mediante algoritmos de tracking,” B.S. thesis, 2016.
- [15] D. Conde. (2018) Regulación de drones en colombia: Conversación con la aerocivil. [Online]. Available: <https://apd.org/regulacion-de-drones-en-colombia/1>

