ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DESARROLLO DE UNA HERRAMIENTA COMPUTACIONAL PARA LA SINTONIZACIÓN DE PARÁMETROS DE CONTROLADORES PID Y SMC PARA EL SEGUIMIENTO DE TRAYECTORIA DE UN CUADRICÓPTERO BASADO EN ALGORITMOS GENÉTICOS

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN "ELECTRÓNICA Y CONTROL"

VANESSA DEL CISNE OROSCO OROZCO
vanessa.orosco@epn.edu.ec

DIRECTOR: Dr. ING. GEOVANNY DANILO CHÁVEZ GARCÍA, Ms.
danilo.chavez@epn.edu.ec

Quito, junio 2018

AVAL

Certifico que el presente trabajo fue desarrollado por Vanessa del Cisne Orosco Orozco, bajo mi supervisión.

Dr. ING. GEOVANNY DANILO CHÁVEZ GARCÍA, Ms.
DIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo Vanessa del Cisne Orosco Orozco, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

VANESSA DEL CISNE OROSCO OROZCO

DEDICATORIA

Dedico este trabajo a mi madre, porque a ella se lo debo todo, ella es mi ejemplo de fuerza, perseverancia y el impulso que tengo para superarme y ser mejor cada día, a mi hermano porque estoy segura de que logrará grandes metas, y estoy muy orgullosa de lo que ha logrado hasta ahora, esto es por y para ustedes.

“Soy de las que piensan que la ciencia tiene una gran belleza. Un científico en su laboratorio no es sólo un técnico, es también un niño colocado ante fenómenos naturales que le impresionan como un cuento de hadas.” Marie Curie

Vanne.

AGRADECIMIENTOS

A mi madre Cristina por haberme inculcado y formado con valores, a ella le debo todo lo que soy, gracias por esa dedicación con la que nos criaste a mi hermano y a mí, te amo mamita.

A mi padre Jaime por confiar en mí, y haberme apoyado en esta etapa.

A mi hermano Darío por apoyarme siempre, darme concejos, contarme sus problemas y poder confiarle los míos, gracias por haber sido un pilar fundamental en esta etapa de mi vida.

A mi novio Jef, gracias por haber compartido toda esta etapa universitaria, estoy segura de que nos esperan muchos más logros por compartir.

A mi hijo Benji por recibirme en casa de la mejor forma, por estar ahí siempre en las buenas y las malas, por brindarme tantas alegrías y hacerme conocer el verdadero amor.

A mis tutores, Dr. Chávez y Dr. Camacho por su paciencia y apoyo, por el tiempo dedicado y sus palabras de aliento, y al Dr. Aguilar por haberme guiado en este proyecto, muchas gracias.

A mi querida amiga y profesora Raquel Vásquez (+), gracias por haber encendido en mi la chispa de querer ser ingeniera aun cuando no entendía el significado de esa palabra, a Zoilo Cárdenas, Jacqueline Rivera, Mónica Acosta, William Chamorro, Alicia Carabalí, gracias a todos Uds, porque siento que me ayudaron a ser mejor persona y a cambiar una parte de mi para bien, que mejor momento que este para brindarles mi admiración, gratitud y aprecio.

A mi amiga Lisbeth, gracias por tantos momentos juntas, sabes que te quiero mucho y gracias por ser mi mejor amiga, a mis compañeros de clase que hicieron que Quito y la Politécnica sea mi segundo hogar, Danny G, Carlitos R, Danny P. (chimbo), Ricky F, gracias por hacer de esta etapa más llevadera, gracias muchachos.

Vanne

ÍNDICE DE CONTENIDO

AVAL I
DECLARACIÓN DE AUTORÍA II
DEDICATORIA III
AGRADECIMIENTOS IV
ÍNDICE DE CONTENIDO V
RESUMEN VIII
ABSTRACT IX
1. INTRODUCCIÓN 1
1.1 Objetivos 2
1.2 Alcance 2
1.3 Marco Teórico 3
Vehículos Aéreos no Tripulados 3
Cuadricóptero 3
Mecánica del Cuadricóptero 4
Controladores 6
Control PID 6
Sintonización de Controladores PID 7
Control por Modos Deslizantes (SMC) 9
Análisis de Estabilidad 11
Algoritmos Genéticos 13
Base Biológica 13
Estructura de Los Algoritmos Genéticos Clásicos 15
Parámetros de los Algoritmos Genéticos 16
Etapas del Algoritmo Genético 17
Índices de Desempeño de un Sistema Controlado 21
2. METODOLOGÍA 23
2.1 Modelo Matemático del Cuadricóptero 23
Matriz de Rotación 24
Modelo Cinemático 25
Modelo Dinámico 26
Dinámica Traslacional 27
Dinámica Rotacional 30

2.2 Algoritmos de Control 36
Control de Orientación y Altitud por Modos Deslizantes (SMC) 36
Control de Altitud 36
Control para el Sistema de Orientación 39
Control para el Seguimiento de Trayectorias 39
Control para el Seguimiento de Trayectorias mediante PID 39
Control para el Seguimiento de Trayectorias mediante SMC 40
Transformación del Sistema de Traslación a Rotación 41
2.3 Algoritmos Genéticos 42
Sintonización de un PID para una planta de Primer Orden con AG 42
Sintonización de controladores PID y SMC, para el control de altura y seguimiento de trayectorias de un Cuadricóptero mediante AG. 47
Costo Computacional de los Algoritmos Genéticos 51
Herramienta Computacional para la sintonización de Controladores PID y SMC para el seguimiento de trayectoria de un cuadricóptero. 52
2.4 Generación de Trayectorias 54
2.5 Interfaz Gráfica de Usuario 55
3. RESULTADOS Y DISCUSIÓN 59
3.1 Parámetros del Cuadricóptero 59
3.2 Control de Orientación mediante SMC 60
3.3 Control de Altura para un Cuadricóptero SMC Y SMC- AG 62
3.4 Control de Seguimiento de Trayectoria de un Cuadricóptero 64
Seguimiento de Trayectoria para una Curva de Lissajous. 66
Seguimiento de Trayectoria para una Sinusoide. 72
Seguimiento de Trayectoria para una Trayectoria Cuadrada. 78
Seguimiento de Trayectoria para una Trayectoria Helicoidal. 83
3.5 Pruebas de Robustez 87
Cambios de Masa 87
Rechazo a Perturbaciones 88
Perturbaciones a una Curva de Lissajous 88
Perturbaciones a una Sinusoide 90
Perturbaciones a una Trayectoria Cuadrada 92
4. CONCLUSIONES 95
4.1 CONCLUSIONES 95
4.2 RECOMENDACIONES 97
5. REFERENCIAS BIBLIOGRÁFICAS 98

6. ANEXOS..... 101
ORDEN DE EMPASTADO..... 115

RESUMEN

El procedimiento de establecer valores a los parámetros de un controlador es conocido como sintonización. En el caso de los controladores tipo PID existen métodos de sintonización propuestos, por ejemplo, por Zieglers Nichols; sin embargo, al trabajar con sistemas no lineales muchas de las veces se requiere un método de ajuste heurístico. Por otro lado, para la sintonización de los parámetros de un SMC (*Sliding Mode Control*) es fundamental el conocimiento que el diseñador tenga del sistema. Debido a la naturaleza no lineal del controlador, los SMC son más difíciles de sintonizar.

Para solucionar este problema, se plantea realizar una sintonización de los parámetros de los controladores PID y SMC para el seguimiento de trayectoria de un cuadricóptero mediante Algoritmos Genéticos. El objetivo de esta técnica es brindar una solución a un problema a través de la optimización de un criterio en específico conocido como función objetivo. Esto es posible utilizando un espacio de búsqueda definido al inicio del algoritmo, el mismo que evoluciona mientras se ejecute dicho algoritmo.

En el presente trabajo, se plantea el desarrollo de una herramienta computacional, que permita sintonizar de forma automática controladores PID y SMC, aplicados al seguimiento de trayectoria de un cuadricóptero. La base para el diseño de un controlador radica en escoger el modelo adecuado, por ello en el presente se trabaja con el modelo dinámico del cuadricóptero sin despreciar los efectos giroscópicos del mismo.

Los controladores sintonizados mediante Algoritmos Genéticos presentan mejores resultados al analizar los índices de desempeño obtenidos, la programación se desarrolla en Matlab y Simulink.

PALABRAS CLAVE: Sintonización, UAV, PID, SMC, Algoritmos Genéticos, ISE.

ABSTRACT

This work describes the process of selecting the best parameters of a controller applied trajectory tracking of a quadcopter based in genetic algorithms. For PID controllers, there are tuning methods studied by Ziegler and Nichols; however, many times a method of heuristic adjustment requires, when working with non-linear systems. The experience of the designer is critical for the tuning of the parameters of an SMC (*Sliding Mode Controller*). Because of the non-linear nature of controller, controllers based on the sliding mode approach are more difficult to tune.

That is why in this work, we propose to use Genetic Algorithms as a tool for the tuning of PID and SMC controllers for trajectory tracking of a quadcopter. This optimization technique is based on the evolutionary principle of the species, and its objective is to provide a solution through the optimization of a specific criterion known as objective function. This is possible by using a search space defined at the beginning of the algorithm which evolves under specific rules while executing said algorithm.

To solve this problem, we propose to develop a computational tool that allows an automatic adjustment of the tuning parameters of PID and SMC controllers applied to trajectory tracking of a quadcopter, to make the simulation more realistic, considers the gyroscopic effects of the drone produced on the X and Y axes.

A graphical user interface is also developed, which allows comparing the performance indices obtained by the tunings made to the different trajectories, resulting in the controllers tuned by Genetic Algorithms with better performance.

KEY WORDS: Tuning, UAV, PID, SMC, Genetic Algorithms, ISE.

1. INTRODUCCIÓN

Uno de los principios que determinan el desempeño de un sistema es la calibración o ajuste de los parámetros de control; de los valores de dichos parámetros dependerá en gran parte el triunfo o fracaso de las leyes de control. La estrategia de control más utilizada actualmente en la industria sigue siendo la PID,[1] debido a que se pueden ajustar sus parámetros fácilmente. Sin embargo, los controladores robustos como el SMC (*Sliding Mode Control*) están tomando ventaja debido a su mejor respuesta frente a perturbaciones o cambio de parámetros de la planta.

El diseño y calibración de los controladores se efectúa en base a métodos de sintonización existentes como el de Ziegler y Nichols para el caso de los controladores PID; sin embargo, muchas de las veces, cuando la planta no responde de manera adecuada para la aplicación de estos métodos, la sintonización es realizada en función del conocimiento que se tenga del sistema. En el caso de los controladores SMC debido a su utilización en plantas no lineales la calibración de sus parámetros llega a convertirse en un proceso heurístico. Por otra parte, es importante señalar que se han desarrollado nuevas técnicas de optimización basadas en inteligencia artificial. Entre estas, se destacan los Algoritmos Genéticos (AG), que mediante la optimización de la función objetivo que representa al sistema, se encuentran las posibles soluciones de valores para los parámetros de las leyes de control, convirtiendo a los algoritmos genéticos en una herramienta útil para la sintonización de controladores. Los Algoritmos Genéticos constituyen un método de optimización inspirados en comportamientos biológicos y evolutivos de las especies.

La importancia del control de posición de los vehículos aéreos no tripulados ha hecho que se compare entre diversas técnicas de control para los mismos. En el presente trabajo se diseña un controlador clásico tipo PID y un controlador robusto SMC, para el seguimiento de trayectoria de un cuadricóptero. Ambos controladores son diseñados a partir del modelo dinámico del drone, y posteriormente se desarrolla una herramienta de sintonización basada en Algoritmos Genéticos para dichos controladores. Los AG son un método que difiere de algoritmos tradicionales, [2] puesto que utiliza valores de una función aptitud o función objetivo. En el caso de sistemas de control los índices de desempeño como el ISE, son utilizados como medidas cuantitativas del desempeño del sistema, por esto tomar al ISE, IAE, ITAE, entre otros, como función aptitud para la sintonización de parámetros de controladores mediante Algoritmos Genéticos, resulta una opción adecuada.

En el presente proyecto se utiliza un Algoritmo Genético simple para obtener los parámetros de sintonización de los controladores PID y SMC, posteriormente, se realiza un análisis de los índices de desempeño (ISE, IAE) al sintonizar los parámetros de los controladores con y sin AG, así como pruebas de robustez en ambos sistemas. Se desarrolla una herramienta que permite cambios en los parámetros de los Algoritmos Genéticos y del cuadricóptero, finalmente, los resultados son mostrados en una interfaz gráfica creada en GUI (*Graphical User Interface*) de Matlab.

1.1 Objetivos

El Objetivo General de este Estudio Técnico es: Desarrollar una herramienta computacional para la sintonización de parámetros de un Controlador PID y SMC para el seguimiento de trayectoria de un cuadricóptero basado en Algoritmos Genéticos.

Los objetivos específicos de este Estudio Técnico son:

- Realizar una recopilación bibliográfica y escoger el modelo del cuadricóptero a trabajar en el presente.
- Diseñar un control PID y SMC para el seguimiento de trayectoria del cuadricóptero.
- Desarrollar un programa basado en Algoritmos Genéticos para la sintonización de los parámetros del controlador PID y SMC para el seguimiento de trayectoria del cuadricóptero.
- Analizar la sintonización con AG, tomando en cuenta un índice de desempeño como el ISE.
- Desarrollar una Interfaz Gráfica como una herramienta de visualización de resultados.

1.2 Alcance

El modelo será escogido luego de una recopilación bibliográfica, se trabajará con el modelo dinámico, no se desprecian los efectos giroscópicos del cuadricóptero, así como datos de longitud de brazo e inercias. El control de Orientación en roll, pitch y yaw será controlado mediante SMC al igual que el control de altura. Para el seguimiento de trayectoria se generará al menos un caso de estudio utilizando una curva de Lissajous, y para su control se podrá escoger el tipo de control: PID, SMC, PID-AG, SMC-AG, cabe recalcar que toda la programación será desarrollada en Matlab- Simulink. Además, para analizar la sintonización de cada controlador se utilizará al menos un índice de desempeño del sistema. Se desarrollará una Interfaz Gráfica GUIDE, en la que se podrá escoger el controlador con el que se desee trabajar y se visualizará la gráfica del seguimiento de trayectoria con el controlador escogido, también se tendrá una opción para visualizar el índice de desempeño del sistema.

1.3 Marco Teórico

Vehículos Aéreos no Tripulados

Los UAV (*unmanned aerial vehicle*) o vehículos aéreos no tripulados, son conocidos comúnmente como drones, ver Figura 1.1. El término UAV se hizo famoso en la década de los 90 para describir a las aeronaves robóticas las cuales reemplazaron a los vehículos aéreos pilotados de forma remota “RPA” (*Remotely Piloted Aircraft*) [3]. Los UAV son robots a los cuales se les asignan misiones de vuelo para ejercicios civiles y militares sin presencia de piloto alguno. Su principio de funcionamiento se basa en fuerzas aerodinámicas para producir movimiento, entre las aplicaciones en las que podemos encontrarlos son fotografía aérea, agricultura y silvicultura, mapeo aéreo y meteorología, tareas policiales y militares, investigaciones realizadas por laboratorios de universidades, entre otros.

Figura 1.1. Vehículo aéreo no tripulado [4]

Cuadricóptero

Es un vehículo aéreo no tripulado (UAV), formado por cuatro rotores separados de forma simétrica (90 grados entre sí), como se observa en la Figura 1.2. En cada rotor se encuentra una hélice, las cuales son las responsables de la fuerza de propulsión para lograr los seis grados de libertad, tres traslacionales (x, y, z) y tres rotacionales (roll ϕ, pitch θ, yaw ψ),

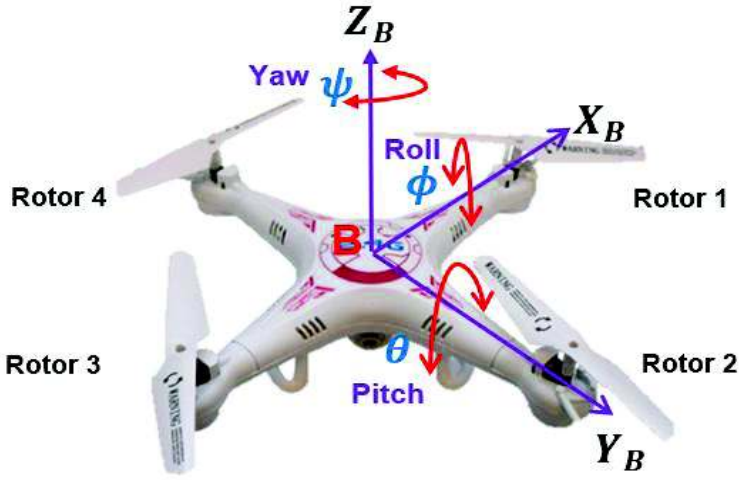


Figura 1.2. Diagrama de un cuadricóptero

Mecánica del Cuadricóptero

El cuadricóptero está formado por cuatro hélices, dos giran en sentido de las manecillas del reloj y dos en sentido contrario a las manecillas del reloj, como se observa en la Figura 1.3, para crear la fuerza de empuje necesaria para que el sistema se eleve. Para analizar la mecánica del cuadricóptero se plantea un sistema de referencia fijo a tierra (***I)*** y un marco fijo al cuerpo del cuadricóptero (***B).*** La mecánica del cuadricóptero se rige por cuatro movimientos que son: empuje o elevación vertical (altitud), roll o alabeo (eje X), pitch o cabeceo (eje Y) y yaw o guiñada (eje Z), dados en el sistema de referencia ***B*** respecto a ***I*** [5]. En el eje X_B se ubican los rotores 1 y 3, y en el eje Y_B los rotores 2 y 4, como se muestra en la Figura 1.3.

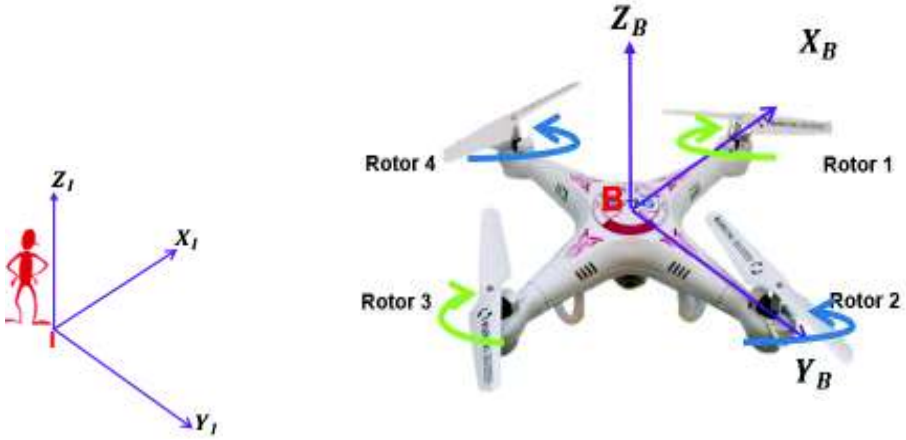


Figura 1.3. Sistemas de referencia del cuadricóptero

Empuje o Elevación Vertical

El empuje o elevación vertical, el cual se lo expresa como U1, ver Figura 1.4. Es el resultado de lograr que la sumatoria de las fuerzas generada por los 4 pares motores-hélices sea mayor que el peso generado por la acción gravitatoria; es decir se puede ascender (aumento de la velocidad de los rotores de forma equitativa), descender (disminución de la velocidad de los rotores de forma equitativa) o mantener el vuelo estable a cierta altitud dependiendo de la fuerza U1 aplicada al drone.

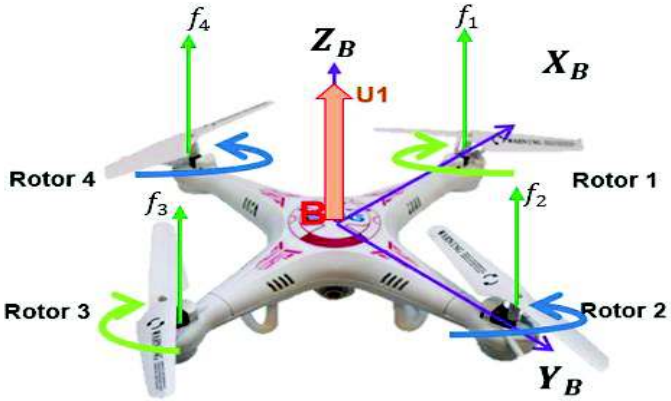


Figura 1.4. Empuje o elevación vertical

Movimiento en Roll (ϕ) o Alabeo

El movimiento en Roll o de alabeo (movimiento de derecha a izquierda) se realiza en torno al eje X y es representado por ϕ [rad/s], como se muestra en la Figura 1.5. Este movimiento se consigue al realizar un incremento de velocidad en el rotor 2 y disminución de velocidad en el rotor 4 o viceversa, el aumento o disminución de velocidad de los rotores mencionados debe darse en la misma proporción de tal forma que el empuje se mantenga. [6]

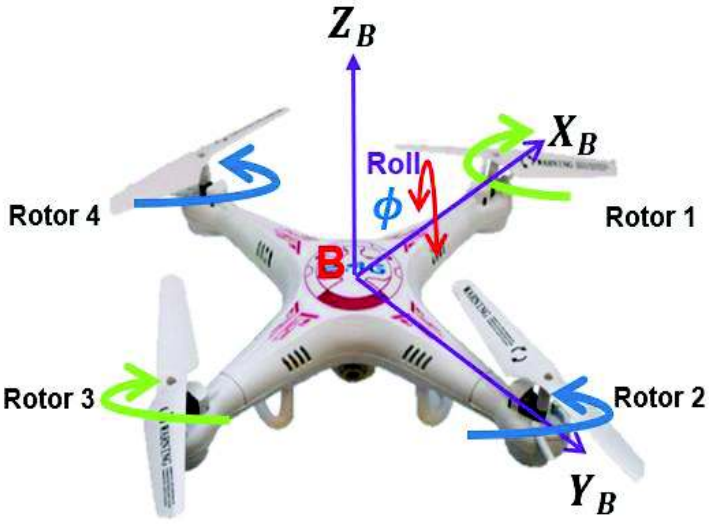


Figura 1.5. Movimiento en Roll o Alabeo

Movimiento en Pitch (θ) o Cabeceo

El movimiento en Pitch o Cabeceo (movimiento de adelante hacia atrás) se da en torno al eje Y y es representado por θ [rad/s], como se observa en la Figura 1.6. Este movimiento se produce de forma similar al movimiento en roll al realizar un incremento de velocidad en el rotor 1 y disminución de velocidad en el rotor 3 o viceversa, el aumento o disminución de velocidad debe darse en la misma proporción de tal forma que el empuje se mantenga.

Figura 1.6. Movimiento en Pitch o Cabeceo

Movimiento en Yaw (ψ) o Guiñada

El movimiento en Yaw o Guiñada se da alrededor del eje Z y es representado por ψ [rad/s], se produce al realizar un incremento en las velocidades de los rotores 1 y 3 y un decremento en las velocidades de los rotores 2 y 4, tomando como referencia las posiciones de los rotores mostradas en la Figura 1.7.

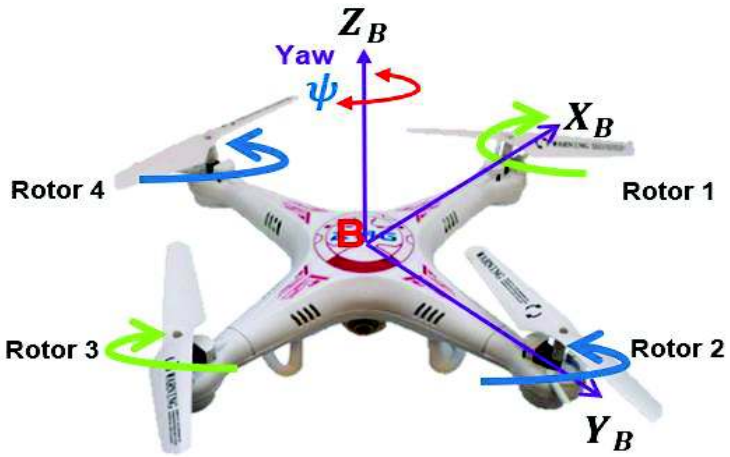


Figura 1.7. Movimiento en Yaw o Guiñada

Controladores

Los controladores han sido creados para determinar y minimizar los errores producidos al comparar el valor de referencia con el valor medido, la forma en la cual el controlador produce la señal de control se denomina acción de control [7]. Un controlador clásico es el de tipo PID; en la actualidad, muchos sistemas son operados con controladores robustos como el SMC que se adaptan mejor a sistemas no lineales.

Control PID

El control PID es uno de los más utilizados a nivel industrial, combina las tres acciones: proporcional, integral y derivativa, ver Figura 1.8. Dependiendo de las aplicaciones que vaya a ser usado un controlador tipo PID puede ser P, PI, PD o PID. [8]

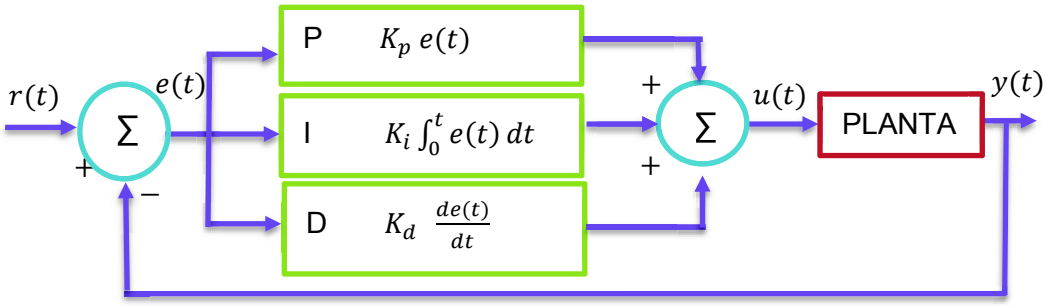


Figura 1.8. Estructura de un Controlador PID

Control Proporcional (P)

Este tipo de controlador es uno de los más sencillos, consiste en añadir al sistema una constante K_p conocida comúnmente como ganancia.

Control Integral (I)

Este tipo de control efectúa la acción correctora mediante la integral del error $e(t)$, permitiendo eliminar la señal del error en régimen permanente. Donde K_i se describe como una ganancia integral ajustable.

Control Derivativo (D)

Este controlador tiene como principal característica su efecto de anticipación, el cual consiste en producir una corrección antes de que el error crezca. Donde K_d es la ganancia derivativa, si K_d no se calcula bien puede llevar a saturación a alguno de los elementos del sistema.

Finalmente, la ley de control PID en función del tiempo está definida por la Ecuación 1.1.

$$u(t) = K_p\, e(t) + K_i \int_0^t e(t)\, dt + K_d \frac{de(t)}{dt}$$

Ecuación 1.1 Ley de control para un PID

De donde:

$$K_i = \frac{K_p}{\tau_i}$$

Ecuación 1.2 Ganancia de la parte integral

Y:

$$K_d = K_p \tau_d$$

Ecuación 1.3 Ganancia de la parte derivativa

Siendo, τ_i la constante de tiempo de integración y τ_d la constante de tiempo de derivación. Si el lector desea un análisis exhaustivo acerca de este controlador, se recomienda revisar la bibliografía citada en [7].

Sintonización de Controladores PID

El ajuste de valores de los parámetros K_p, K_i, K_d se conoce como sintonización, Ziegler y Nichols propusieron una serie de reglas para afinar controladores PID y sin suponer conocimiento previo de la planta. A continuación, se revisarán algunos de los métodos más utilizados para la sintonización de controladores PID propuestos por Ziegler y Nichols. [9]

Método de la Curva de Reacción

Se lo realiza en lazo abierto y se basa en la respuesta al escalón. Este método es aplicable si al someter a la planta ante una entrada paso se obtiene una respuesta en forma de S, como se observa en la Figura 1.9. Posteriormente se traza una recta tangente al punto de inflexión de la curva, donde L es el tiempo en atraso y τ la constante de tiempo, los parámetros de sintonización son obtenidos según la Tabla 1.1.

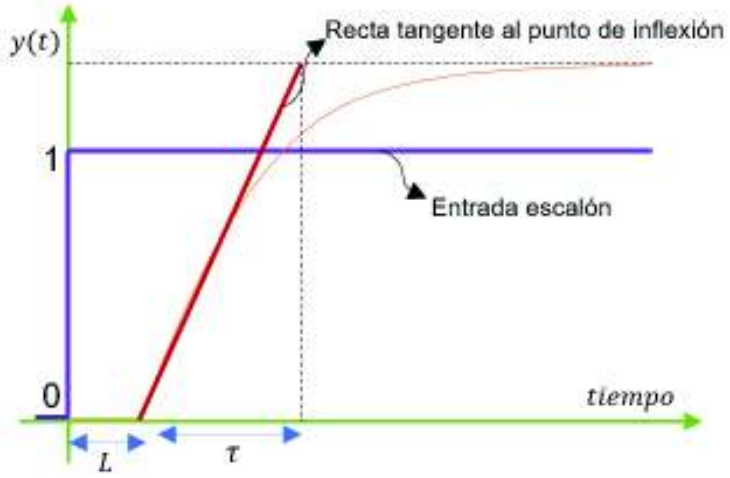


Figura 1.9. Curva de Reacción en forma de "S"

Mediante L y τ, se calculan los parámetros del controlador PID utilizando la Tabla 1.1.

Tabla 1.1. Parámetros de sintonización, método Curva de Reacción

Tipo de Controlador	K_p	τ_i	τ_d
P	$\frac{\tau}{L}$	∞	0
PI	$0.9\frac{\tau}{L}$	$\frac{L}{0.3}$	0
PID	$1.2\frac{\tau}{L}$	$2L$	$0.5L$

Método de Oscilaciones Sostenidas

Este método es aplicable a sistemas que puedan tener oscilaciones sostenidas, consiste en eliminar los efectos de las partes integral y derivativa, y utilizando solo la parte proporcional del controlador K_p se consigue que el sistema oscile de forma sostenida como se muestra en la Figura 1.10. El valor de ganancia con el cual se logra oscilar el sistema se conoce como ganancia crítica K_{cr}, que corresponde a un periodo crítico P_{cr}.

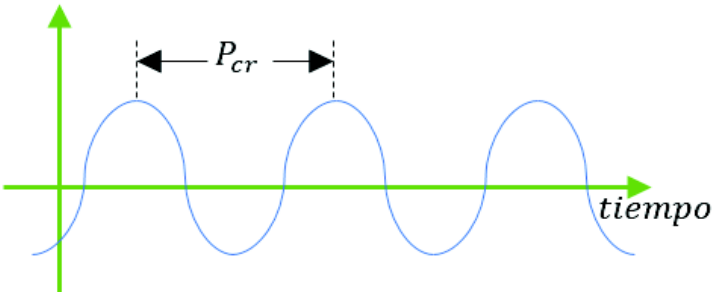


Figura 1.10. Oscilación Sostenida de una Planta

Mediante K_{cr} y P_{cr}, se calculan los parámetros del controlador PID utilizando la Tabla 1.2.

Tabla 1.2. Parámetros de sintonización, método Oscilaciones Sostenidas

Tipo de Controlador	K_p	τ_i	τ_d
P	$0.5K_{cr}$	∞	0
PI	$0.45K_{cr}$	$\frac{1}{2}P_{cr}$	0
PID	$0.6K_{cr}$	$\frac{1}{2}P_{cr}$	$0.125P_{cr}$

Control por Modos Deslizantes (SMC)

El SMC (Sliding Mode Control) o Control por Modos Deslizantes es una técnica de control robusto derivada del Control por Estructura Variable o VSC (Variable Structure Controller) [10].Debido a su robustez controla sin problemas plantas con incertidumbres y no linealidades. En el presente se utilizará las siglas SMC para referirnos al Control en Modo Deslizante.

Ley de Control SMC

Su ley de control se compone de dos partes: función equivalente o continua y función switching o discontinua.

$$u(t) = \mu_C(t) + \mu_D(t)$$

Ecuación 1.4. Ley de control para un controlador SMC

Donde $u(t)$ es la ley de control, $\mu_C(t)$ es la función continua y $\mu_D(t)$ es la función discontinua. Para el diseño del SMC se define una superficie de deslizamiento $s(t)$, que representa la dinámica del sistema en lazo cerrado, para lo cual se tiene una ecuación diferencial que actué sobre el error como se observa en la Ecuación 1.5 y la Figura 1.11.

$$s(t) = \left(\frac{d}{dt} + \lambda\right)^n e(t)dt$$

Ecuación 1.5. Superficie deslizante de orden n

Donde n representa el orden del sistema, λ es un parámetro de sintonización y $e(t)$ es el error. Si el sistema es de orden $n = 1$, entonces la Ecuación 1.5, se puede reescribir: [11]

$$s = \dot{e} + \lambda e$$

Ecuación 1.6. Superficie deslizante de orden 1

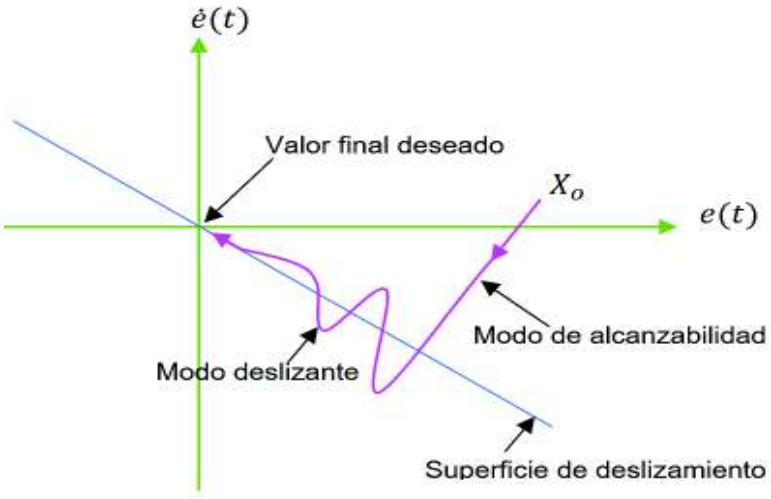


Figura 1.11. Interpretación gráfica de un SMC [10]

Función Equivalente o Continua

La función continua del SMC es la encargada de seguir la dinámica que describe a la superficie de deslizamiento, para encontrar $\mu_C(t)$ se utiliza el modelo matemático de la planta. El objetivo de la parte equivalente es mantener al sistema limitado a la superficie de deslizamiento, es decir se debe cumplir que $\dot{s}(t) = 0$.

Función Discontinua o Switching

La parte discontinua del controlador $\mu_D(t)$ es la responsable de forzar al error hacia la superficie de deslizamiento. Esta función contiene un elemento no lineal que actúa como un switch de la ley de control, permitiendo alcanzar la superficie de deslizamiento a gran velocidad. La función discontinua se describe como:

$$\mu_D(t) = K_D sign(s(t))$$

Ecuación 1.7. Parte discontinua del SMC

Donde K_D es el parámetro de sintonización encargado de la velocidad de alcance de la superficie deslizante, y $sign(s(t))$ es una función no lineal de la superficie de deslizamiento $s(t)$. Dado que la ley de control de la parte discontinua está diseñada con la función signo y para disminuir el fenómeno de "chattering" que crea oscilaciones de alta frecuencia, [11] se suaviza la función discontinua aproximándola a una función sigmoide, como se describe:

$$sigmoide\big(s(t)\big) = \frac{s(t)}{|s(t)| + \delta}$$

Ecuación 1.8. Aproximación a una sigmoide

Donde δ es el parámetro de sintonización positivo escalar de ajuste de la sigmoide cuyo trabajo es reducir las oscilaciones de alta frecuencia. Sustituyendo la Ecuación 1.8 en la Ecuación 1.7, la función discontinua se reescribe como:

$$\mu_D(t) = K_D \frac{s(t)}{|s(t)| + \delta}$$

Ecuación 1.9. Parte discontinua del SMC aproximada a una sigmoide

Análisis de Estabilidad

La teoría de estabilidad juega un rol muy importante en los sistemas de ingeniería y robótica. La teoría de estabilidad de Lyapunov es una de las técnicas más utilizadas en el análisis de estabilidad, así como para el diseño de controladores. Existen varias técnicas para el análisis de estabilidad de Lyapunov, en el presente trabajo se estudiará el método directo de Lyapunov.

Método Directo de Lyapunov

Un punto de equilibrio de un sistema dinámico es estable en el sentido de Lyapunov si todas las soluciones que nacen en las cercanías del punto de equilibrio permanecen en dichas cercanías; de otra forma resulta inestable. El punto de equilibrio es, además, asintóticamente estable si las soluciones a más de permanecer en las cercanías de este tienden hacia el punto de equilibrio a medida que transcurre el tiempo. [12]

A) Estable[13]

El punto de equilibrio $x = x_e$ de la Figura 1.12, se dice que es estable en el sentido de Lyapunov si $\forall\, \varepsilon > 0,\ \exists\, \delta(\varepsilon) > 0,$ tal que todas las soluciones cumplen:

$$\|x(0) - x_e\| < \delta \ \Rightarrow\ \|x(t) - x_e\| < \varepsilon, \qquad \forall\, t \geq t_0,$$

Caso contrario, se dice que el punto de equilibrio es inestable

B) Asintóticamente Estable [13]

El punto de equilibrio $x = x_e$ de la Figura 1.12, es asintóticamente estable, si es estable en el sentido de Lyapunov y si, además $\exists\ \delta > 0$ tal que:

$$\|x(0) - x_e\| < \delta \Rightarrow \lim_{t\to\infty} x(t) = x_e$$

quedando dentro de la esfera determinada por δ, caso contrario, se dice que el punto de equilibrio es inestable.

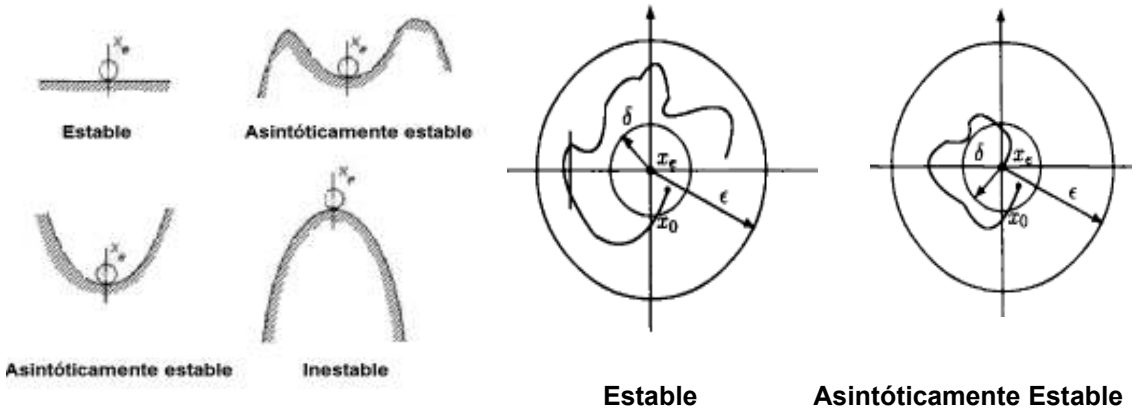


Figura 1.12. Definiciones de Estabilidad en el sentido de Lyapunov. [13],[14].

Para afirmar que un sistema es estable se deben cumplir las siguientes condiciones:

1) Se escoge una función candidata de Lyapunov V, tal que $V(x_e)$ sea definida positiva.

$$V(x_e) > 0\ ;\ \forall\ x_e\ > 0$$

Ecuación 1.10. Función candidata de Lyapunov

2) La candidata de Lyapunov evaluada en un punto de equilibrio del sistema x_e, debe ser igual a 0.

$$V(x_e) = 0$$

Ecuación 1.11. Candidata de Lyapunov evaluada en el punto de equilibrio

3) Finalmente, la derivada de la función de Lyapunov debe ser menor a cero, como se indica en la Figura 1.13.

$$\dot{V}(x_e) < 0$$

Ecuación 1.12. Derivada de la candidata de Lyapunov

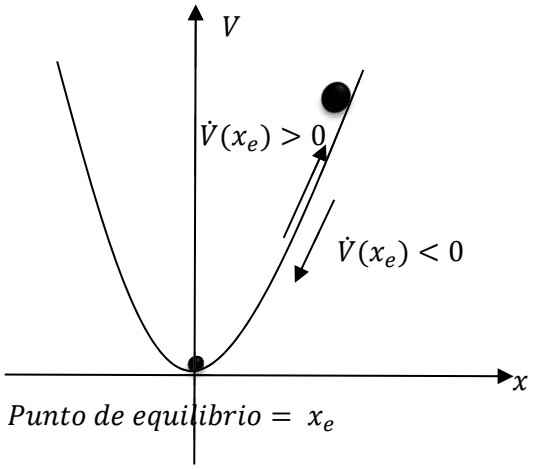


Figura 1.13. Condiciones de Estabilidad de Lyapunov [15]

Algoritmos Genéticos

Base Biológica

Todos los seres vivos están formados por células, cada célula contiene el mismo número de cromosomas, los cuales son cadenas de ADN (Ácido Desoxirribonucleico) y definen cómo es un ser vivo. Cada cromosoma está formado por genes que constituyen bloques de ADN, cada gen define una característica del individuo, por ejemplo, el color de piel. Al conjunto de todos los cromosomas se lo conoce como genoma. A la manifestación externa de un ser vivo (color de ojos, estatura, tamaño de pies) se la conoce como fenotipo. Al darse la reproducción en los seres vivos se produce una etapa denominada cruce, en esta etapa los genes de los padres se combinan para generar un nuevo cromosoma (retoño u offspring). Este nuevo cromosoma puede sufrir una etapa conocida como mutación. Los genes sufren cambios normalmente por errores al ser copiados de sus padres. Otra definición interesante es el fitness de un ser vivo, este indica que tan bueno es un individuo en función del grado de adaptación al entorno, asegurando así su supervivencia [16].

Los algoritmos genéticos, son técnicas de optimización que se basan en la teoría de la evolución de Darwin, en donde de forma natural se selecciona a los individuos que presenten mejores características genéticas[17]. Los AG permiten que una población compuesta por muchos individuos evolucione bajo reglas de selección especificadas a un estado que maximiza la “aptitud”. El matemático Hans-Joachin Bremermann fue el primero en ver la evolución como un proceso de optimización, además de realizar una de las primeras simulaciones con cadenas binarias que se procesaban por medio de reproducción, selección y mutación [18]. Los AG en la automatización son utilizados principalmente para optimizar

movimientos de manipuladores, en la regulación y sintonización de parámetros, seguimientos de trayectoria y evasión de obstáculos [19], [20]. Antes de entrar en la teoría de los AG es necesario conocer la siguiente terminología, ilustrada en la Figura 1.14.

Rasgos Fenotípicos

Son las características físicas y de comportamiento que diferencian a un individuo de otro, son parcialmente determinadas por la herencia.

Alelo

Son los diferentes valores con los que puede representarse un gen.

Gen

Es el material que contiene información genética dentro de un cromosoma y dispuesto por un orden fijo.

Cromosoma

Es un arreglo de genes que contienen material genético, también se lo conoce como individuo.

Índice

Es la posición que ocupa un individuo dentro de una población.

Población

Conjunto de individuos o cromosomas

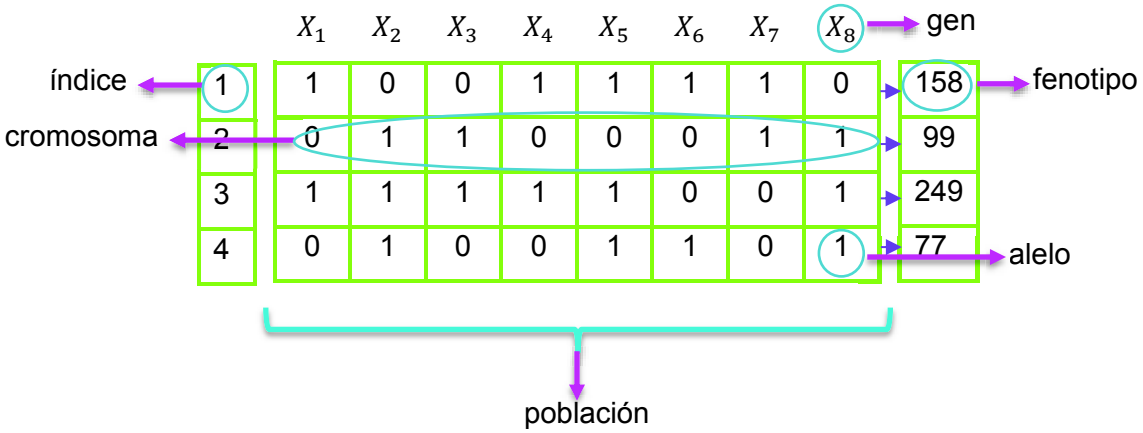


Figura 1.14. Componentes de una Población[21]

Estructura de Los Algoritmos Genéticos Clásicos

Para desarrollar buenas soluciones e implementar la selección natural, se necesita un identificador que permita evaluar y diferenciar entre buenos y malos resultados, dicho indicador se lo conoce como función objetivo o función aptitud [22]. El valor de la función aptitud se denomina costo e indica el desempeño del sistema; es decir, entre más pequeño sea el costo, la optimización es mejor. Para la programación de los AG se empieza como en todo programa declarando las variables a utilizar, sin embargo, durante el desarrollo del código, el AG es bastante diferente pues debe seguir una secuencia estricta de pasos que son ilustrados en la Figura 1.15 a manera de diagrama de flujo y posteriormente se detalla cada etapa.

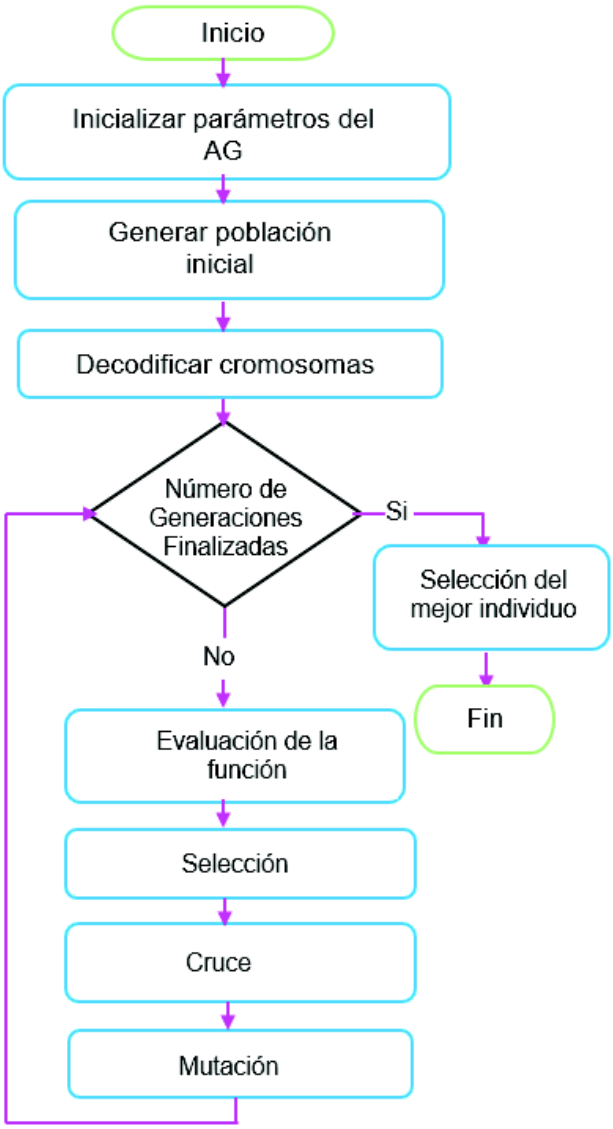


Figura 1.15. Diagrama de Flujo de un AG

Parámetros de los Algoritmos Genéticos

A continuación, se describen los principales parámetros a tomar en cuenta para el desarrollo de un AG: [16], [23], [24].

Tamaño del individuo o Longitud del cromosoma

El tamaño del individuo dependerá de los bits con los que se logre una posible solución de acuerdo con el problema a resolver.

Tamaño de la Población

El tamaño de la población indica el número de individuos que existe en la población para una generación determinada. Si el tamaño de la población es demasiado pequeño, el AG tiene pocas posibilidades de realizar reproducciones lo que significa una búsqueda de soluciones escasa y poco óptima, y si por el contrario el tamaño de la población es elevado, el AG se volverá muy lento ocasionando un costo computacional elevado y no significa que se encuentren mejores soluciones debido a que el algoritmo cae en óptimos locales. Lo idóneo es establecer un tamaño de población adecuada en función del problema.

Número de Generaciones

A medida que pasan las generaciones la población evoluciona y se obtiene individuos con mejores características genéticas, sin embargo, el número de generaciones se debe considerar de acuerdo con el problema y costo computacional.

Porcentaje de Cruce

Este es un indicador de con qué frecuencia los individuos se cruzarán, si este indicador es de 0% los hijos serán idénticos a los padres y serán alterados solo por la mutación, y si el indicador es del 100% significa que todos los individuos nuevos son producto del cruce, sin embargo, es recomendable dejar que algunos padres pasen a las siguientes generaciones por lo que se debe elegir correctamente el porcentaje de cruce

Porcentaje de Mutación

El Porcentaje de Mutación nos indica la frecuencia con la que son mutados los individuos, si este indicador es del 0% significa que los individuos luego del cruce no tendrán ninguna alteración genética y si este indicador es del 100% significa que todos los individuos sufrieron alteraciones luego del apareamiento. La mutación trata de impedir que la búsqueda del mejor

individuo caiga en extremos locales, sin embargo, no es recomendable que tenga un porcentaje grande de mutación debido a que la búsqueda cambiaria de inteligente a aleatoria.

Etapas del Algoritmo Genético

Población Inicial

Se define una población aleatoria binaria de individuos, la misma que está determinada por el número de cromosomas y la longitud de bits que contiene cada cromosoma. El número de poblaciones para algoritmos genéticos multivariables es directamente proporcional al número de variables existentes

Decodificación

En el proceso de decodificación cada cromosoma binario se convierte en decimal, con el objetivo de ser evaluado a través de la función objetivo, cuando los cromosomas ya se encuentran en sus valores decimales se los conoce como fenotipos, ver Figura 1.16.

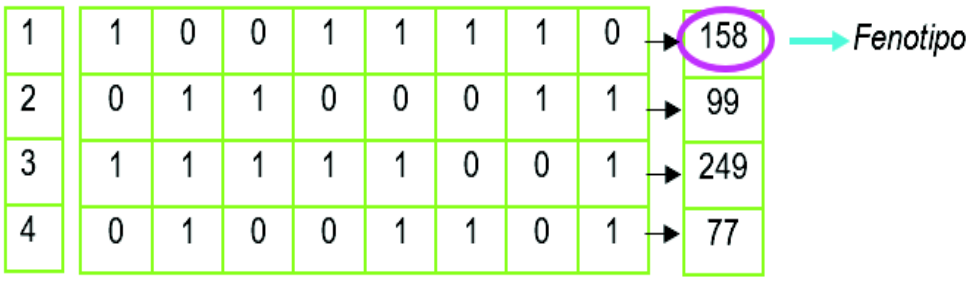


Figura 1.16. Decodificación

Evaluación

Una vez que la población es inicializada y decodificada, se evalúa a la población recién creada y a sus descendientes mediante la función objetivo. Por lo cual la designación de la función objetivo se convierte en una de las partes más importantes de los AG. Por lo tanto, dicha función debe reflejar el valor del individuo de forma real, sin embargo, en la mayoría de los problemas de optimización donde hay varias condiciones de restricción, gran cantidad de los puntos de búsqueda representan a individuos no válidos.

Selección

Mediante la etapa de selección se decide cuáles son los cromosomas que presentan mejores características genéticas para sobrevivir y posiblemente para reproducirse en la próxima generación. El proceso de selección natural debe darse en cada iteración del algoritmo, para

dar lugar a la evolución de la población a lo largo de las generaciones. No todos los sobrevivientes son considerados aptos para la reproducción, solo los superiores (aquellos que resuelvan mejor el problema) se mantienen para el apareamiento y el resto se descartan para dejar espacio a la nueva descendencia. Existen varios métodos para la selección de los mejores individuos:

Emparejamiento de Arriba abajo

Este método de selección es el más sencillo de programar, consiste en ordenar cada individuo respecto a su índice, se empieza en la parte superior de la lista de individuos vinculando de dos en dos, de esta forma el algoritmo empareja filas impares con filas pares.

Emparejamiento aleatorio

Este sistema utiliza un generador de números aleatorios para la selección de los cromosomas y su posterior apareamiento, no es un método recomendado para problemas de mayor grado.

Elitismo

Este sistema indica que el mejor individuo de la generación actual pase a la siguiente generación sin modificarse, con el fin de no perder al mejor cromosoma de la generación presente.

Selección por Rueda de La Ruleta

Para efectuar esta selección todos los individuos son colocados de acuerdo con sus índices y se calcula la probabilidad de cada individuo mediante la siguiente ecuación:

$$P_n = \frac{N_p - n + 1}{\sum_{n=1}^{N_p} n}$$

Ecuación 1.13. Fórmula para calcular la probabilidad de cada individuo

Donde:

n: Es el índice de cada cromosoma.
N_p: Es el número de individuos existentes en la población.

Luego se calcula las probabilidades acumulativas de cada individuo, y se genera un número aleatorio entre 0 y 1, y el individuo cuyo valor de probabilidad acumulativa sea más alto o cercano al número aleatorio será el individuo seleccionado.

Selección por Torneo

Este sistema de selección imita fielmente la competencia por apareamiento de la naturaleza, se selecciona un subconjunto de cromosomas y se inicia una pequeña competencia o torneo, el cromosoma que genere un costo más bajo se convierte en padre, cabe recalcar que los cromosomas que consigan un costo bajo tienen mayor probabilidad de reproducción y los cromosomas con un costo más alto tienen menos probabilidades de reproducción. La selección por torneo es la más indicada para problemas con tamaños de poblaciones considerables.

Cada esquema de selección de padres arroja un grupo de padres diferentes para el apareamiento, por lo tanto, la característica de la nueva generación dependerá del sistema de selección a utilizarse. La Rueda de la Ruleta y Selección por Torneo son las más utilizadas en los problemas de AG.

Apareamiento o Cruce

El apareamiento o cruce es la creación de uno o más descendientes a partir de los padres previamente seleccionados, la forma más común de apareamiento involucra a dos padres para producir dos crías, así el numero de la población se mantiene, el apareamiento o cruce se lo puede realizar de varias formas, entre las más comunes se tiene:

Un Punto de Cruce

Se divide la longitud del cromosoma de ambos padres de igual forma, y se selecciona al azar entre el primero y el ultimo fragmento de los cromosomas de los padres, la madre pasa la primera parte de su cromosoma a su primer descendiente y el padre pasa la segunda parte de su cromosoma a su primer descendiente, de igual forma para su segundo descendiente el padre pasa primero la parte de su cromosoma y luego pasa la madre su parte de cromosoma como se ilustra en la Figura 1.17.

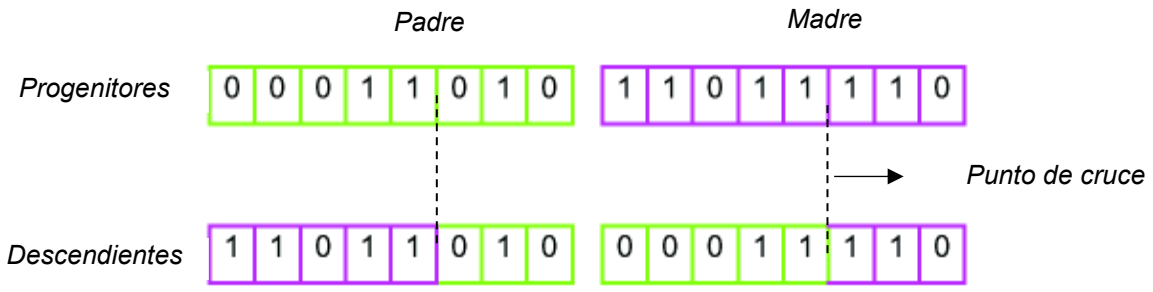


Figura 1.17. Cruce basado en un Punto [16]

Dos Puntos de Cruce

En este sistema de cruce se divide al cromosoma en tres partes, es decir los cromosomas de ambos padres son divididos en tres segmentos de igual forma y siguiendo la lógica del sistema de cruce de un punto se va intercambiando material genético para dar origen a un nuevo descendiente, como se observa en la Figura 1.18.

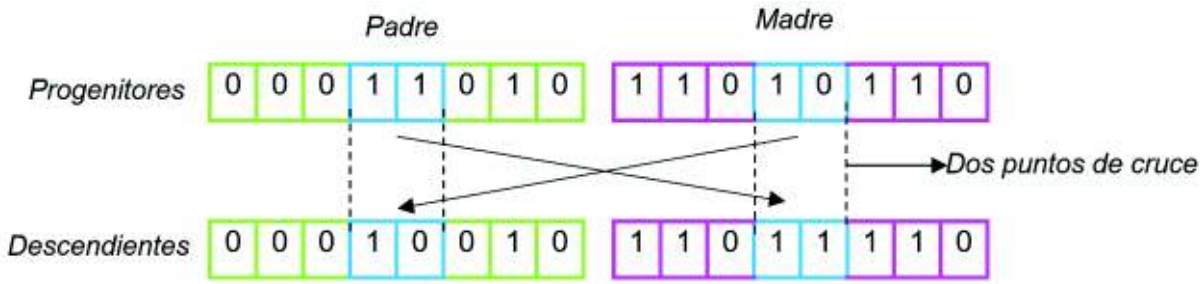


Figura 1.18. Cruce basado en dos Puntos

Mutación

Se entiende por mutación a una variación de la información genética contenida por los nuevos individuos o descendientes, alterando un cierto porcentaje de los bits que contienen los cromosomas. Comúnmente la mutación que se realiza es de punto único que consiste en cambiar un 1 a 0 y viceversa, como se muestra en Figura 1.19, los genes para cambiar son seleccionados aleatoriamente. Con la mutación un AG puede introducir rasgos que no se encuentran en la población inicial, generalmente las mutaciones no se dan en la última iteración ni se permite que muten las mejores soluciones. En la mayoría de los problemas el porcentaje de mutación no pasa de 0.15, esto se debe a que la búsqueda de la mejor solución podría pasar a convertirse de inteligente en aleatoria.

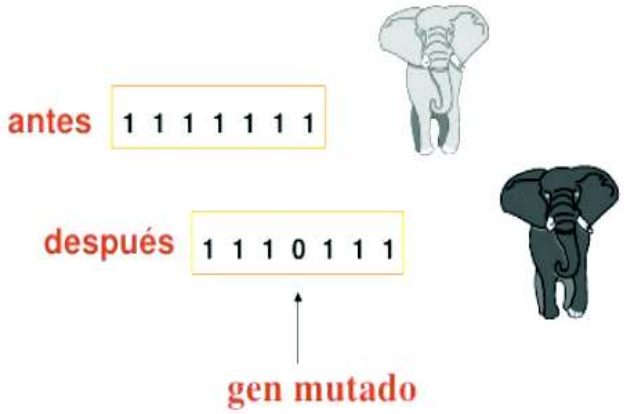


Figura 1.19. Representación gráfica de la Mutación [23]

Índices de Desempeño de un Sistema Controlado

La función objetivo debe evaluar al individuo y reflejar que tan bueno o malo es dicho individuo para el sistema, en los sistemas de control los índices de desempeño son los indicadores de cuán bien funciona el sistema, es por esto, que los índices de desempeño representan una buena opción para ser escogidos como función objetivo. Los índices de desempeño son utilizados en sistemas controlados para evaluar la respuesta del sistema frente a perturbaciones o cambios de parámetros. Al realizar un análisis de estos índices es posible determinar que controlador brinda un mejor rendimiento para el proceso que se está efectuando. Los índices de desempeño son considerados como una medida cuantitativa del desempeño del sistema, y los que más se utilizan en la optimización de sistemas de control son:

Integral del Error Absoluto IAE

La integral del error absoluto IAE está dada por la ecuación:

$$IAE = \int_0^\infty |e(t)|dt$$

Ecuación 1.14. Índice de desempeño IAE

Este criterio es de fácil aplicación y proporciona una respuesta aceptable a la salida del lazo de control, sin embargo, no es capaz de optimizar sistemas sobre amortiguados, para considerar que el sistema controlado tiene un buen desempeño el IAE idealmente debe tender a cero.

Integral del Error Cuadrático ISE

Este criterio está definido por:

$$ISE = \int_0^\infty e(t)^2 dt$$

Ecuación 1.15. Índice de desempeño ISE

Este criterio da mayor ponderación a los errores elevados del sistema, que comúnmente se dan al inicio de la respuesta, y con menos peso significativo evalúa los errores pequeños que se dan al final de la respuesta.[25]

Integral del Tiempo por el Error Absoluto ITAE

Este criterio está definido por la ecuación:

$$ITAE = \int_0^\infty t\,|e(t)|dt$$

Ecuación 1.16. Índice de desempeño ITAE

Al estar el valor absoluto del error multiplicado por el tiempo, permite que errores grandes que ocurren al inicio de la respuesta donde aún los tiempos son pequeños tengan poco peso, y los errores pequeños que se dan al final de la respuesta sean más grandes, se caracteriza por ser un criterio muy selectivo pero difícil de evaluar analíticamente. [25]

Integral del Tiempo del Error Cuadrático ITSE

El criterio de la integral del tiempo por el error cuadrático está determinado por la ecuación:

$$ITSE = \int_0^\infty t\,e(t)^2 dt$$

Ecuación 1.17. Índice de desempeño ITSE

Este criterio da poco peso a los errores grandes que ocurren al inicio del sistema y se enfoca en los errores presentes en el tiempo luego de la entrada al sistema.

2. METODOLOGÍA

En el presente proyecto la investigación realizada es de tipo aplicada, ya que los Algoritmos Genéticos son empleados en la sintonización de los parámetros de control para el seguimiento de trayectoria del cuadricóptero, y mediante experimentación se obtienen los índices de desempeño (ISE, IAE), para determinar si existe mejora en el método de sintonización planteado. En la presente sección se estudia el modelo dinámico del cuadricóptero, el control de orientación y altura es efectuado mediante un controlador robusto como el SMC (*Sliding Mode Control*), mientras que para el control de trayectoria se puede escoger entre PID o SMC, posteriormente los Algoritmos Genéticos son utilizados como herramienta de sintonización.

2.1 Modelo Matemático del Cuadricóptero

Para definir las ecuaciones dinámicas del cuadricóptero, es necesario conocer los sistemas de referencia a los que está regido: estos son: el sistema de referencia fijo a tierra ***I*** y el sistema de referencia móvil ***B*** ubicado en el centro de masa del cuadricóptero, como se observa en la Figura 2.1, [26],[27]. Para la obtención del modelo dinámico, el cuadricóptero es considerado como un sólido rígido con estructura simétrica de masa *m,* las coordenadas traslacionales del centro de masa del cuadricóptero con respecto al eje I quedan definidas por el vector ξ.

$$\xi = \{x, y, z\} \in \mathbb{R}^3$$

Ecuación 2.1. Coordenadas traslacionales

La orientación del cuadricóptero se obtiene al conocer los ángulos de Euler de B respecto a I, que nos indica la rotación del cuadricóptero en el espacio [28].

$$\eta = \{\phi, \theta, \psi\} \in \mathbb{R}^3$$

Ecuación 2.2. Coordenadas Rotacionales

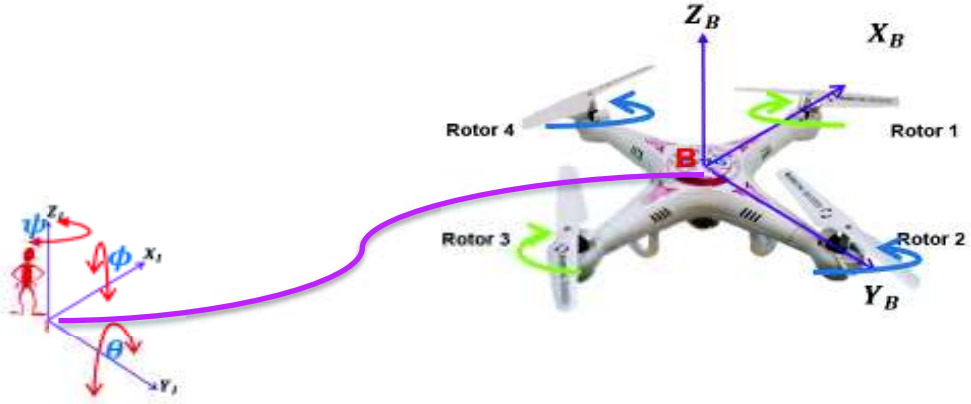
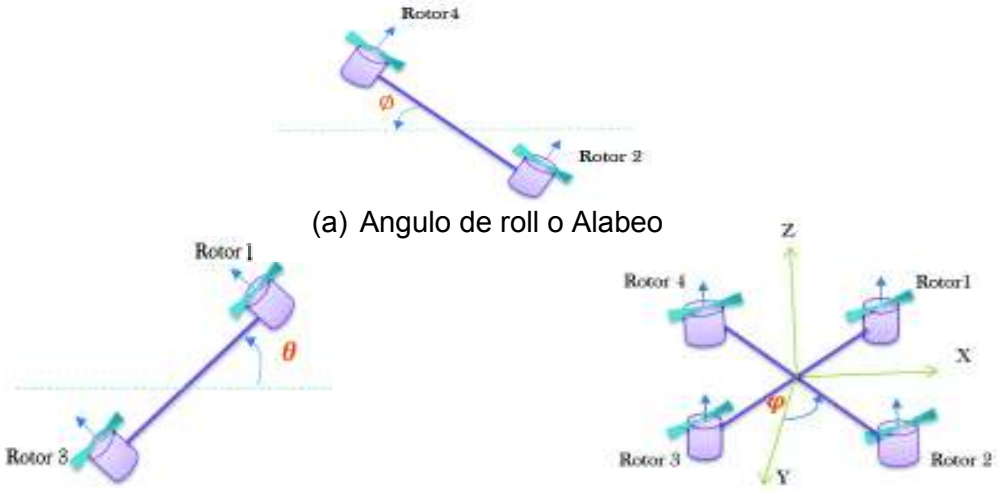


Figura 2.1. Sistemas de Referencia del Cuadricóptero

Matriz de Rotación

Como se indicó, se tiene dos sistemas de referencia, uno en tierra (I) y otro en el centro de masa del cuadricóptero (B). Para obtener el modelo matemático del cuadricóptero es necesario enlazar los dos sistemas de referencia, esto se logra rotando el primer eje de referencia para coincidir con la rotación del segundo, y aplicando las tres rotaciones alrededor de los ejes en un orden dado (ϕ, θ, ψ) [28], [29].

En la Figura 2.2, se observan los ángulos de rotación respecto al sistema referencial I.



(a) Angulo de roll o Alabeo

(b) Angulo de Pitch o Cabeceo

(c) Angulo de Yaw o Guiñada

Figura 2.2. Ángulos de Euler

Rotación de Roll

Al girar el sistema de referencia B un ángulo ϕ, con respecto al eje X_I del marco de referencia I, se obtiene la matriz de rotación de roll $R(x, \phi)$.

$$R(x,\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}$$

Ecuación 2.3. Matriz de rotación de Roll

Rotación de Pitch

Al girar el sistema de referencia B un ángulo θ, con respecto al eje Y_I del marco de referencia I, se obtiene la matriz de rotación de roll $R(y, \theta)$.

$$R(y,\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

Ecuación 2.4. Matriz de rotación de Pitch

Rotación de Yaw

Al girar el sistema de referencia B un ángulo ψ, con respecto al eje Z_I del marco de referencia I, se obtiene la matriz de rotación de roll $R(z,\psi)$.

$$R(z,\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Ecuación 2.5. Matriz de rotación de Yaw

La matriz de rotación completa del sistema de referencia del centro de masa B respecto al sistema en tierra I, se la obtiene al realizar el producto de las matrices de rotaciones de roll, pitch y yaw, la convención escogida es $(z-y-x)$ [30].

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

Ecuación 2.6. Producto de las matrices de rotaciones

$$R(\psi,\theta,\phi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}$$

Resolviendo, se tiene:

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

Ecuación 2.7. Matriz de rotación completa del sistema B respecto a I

La matriz de rotación completa $R(\psi,\theta,\phi)$ y su inversa, es útil para rotar un vector desde el sistema del centro de masa al sistema de referencia fijo en tierra o viceversa.

Modelo Cinemático

El modelo cinemático de un cuadricóptero se expresa en función de las velocidades lineales de cada eje traslacional $V = [V_x, V_y, V_z]$, que son las entradas del sistema, y de las velocidades en el sistema del cuerpo del cuadricóptero $\dot{\xi} = [\dot{x}_B, \dot{y}_B, \dot{z}_B]$, como se muestra en la Ecuación 2.8.

$$\begin{bmatrix} \dot{x}_B \\ \dot{y}_B \\ \dot{z}_B \end{bmatrix} = \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} = V$$

Ecuación 2.8. Velocidades lineales

Luego se enlaza las velocidades del sistema de referencia (I) $\dot{\xi} = [\dot{x}_I, \dot{y}_I, \dot{z}_I]$, con las velocidades del sistema de referencia ubicado en el centro de masa del cuadricóptero (B), para ello se utiliza la matriz de rotación completa $R(\psi, \theta, \phi)$ con lo que se obtiene:

$$\dot{\xi} = R(\psi, \theta, \phi)\, V$$

Ecuación 2.9. Relación de las velocidades con la matriz de rotación completa

Sustituyendo la Ecuación 2.8 en la Ecuación 2.9, se tiene:

$$\begin{bmatrix} \dot{x}_I \\ \dot{y}_I \\ \dot{z}_I \end{bmatrix} = R(\psi, \theta, \phi) \begin{bmatrix} \dot{x}_B \\ \dot{y}_B \\ \dot{z}_B \end{bmatrix}$$

Remplazando la matriz de rotación completa de la Ecuación 2.7, se tiene:

$$\begin{bmatrix} \dot{x}_I \\ \dot{y}_I \\ \dot{z}_I \end{bmatrix} = \begin{bmatrix} \cos\psi\ \cos\theta & -\sin\psi\cos\phi + \cos\psi\sin\theta\sin\phi & \sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi \\ \sin\psi\ \cos\ \theta & \cos\psi\cos\phi + \sin\psi\sin\theta\sin\phi & -\sin\phi\cos\psi + \cos\phi\sin\psi\sin\theta \\ -\sin\theta & \sin\phi\cos\theta & \cos\theta\cos\phi \end{bmatrix} \begin{bmatrix} \dot{x}_B \\ \dot{y}_B \\ \dot{z}_B \end{bmatrix}$$

Para la obtención del modelo cinemático reducido del cuadricóptero, se considera que los ángulos de roll y pitch permanecen muy cercanos a cero durante el tiempo de vuelo, esto permite una simplificación trigonométrica sin reducir la precisión de aproximación del modelo, estableciendo que $\phi \to 0$ y $\theta \to 0$, por lo tanto, $\cos\phi \cong 1$, $\cos\theta \cong 1$ y $\sin\phi \cong 0$, $\sin\theta \cong 0$, entonces se obtiene:

$$\begin{bmatrix} \dot{x}_I \\ \dot{y}_I \\ \dot{z}_I \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_B \\ \dot{y}_B \\ \dot{z}_B \end{bmatrix}$$

Sustituyendo con la Ecuación 2.8, se tiene:

$$\begin{bmatrix} \dot{x}_I \\ \dot{y}_I \\ \dot{z}_I \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix}$$

Ecuación 2.10. Modelo cinemático del cuadricóptero

Modelo Dinámico

El modelo dinámico se obtiene al representar el cuadricóptero como un sólido rígido en tres dimensiones mediante las ecuaciones de Euler-Lagrange y las Leyes de Newton. La dinámica de un cuerpo relaciona: fuerzas, velocidades, aceleraciones, dimensiones, masas, momentos

de inercia, entre otros. Se define las coordenadas generalizadas para el cuadricóptero formadas por las posiciones lineales (x, y, z) en el sistema de referencia I respecto a B y las posiciones angulares (ϕ, θ, ψ) en el sistema de referencia B respecto a I.
Partiendo de la Ecuación 2.1 y la Ecuación 2.2, se tiene:

$$q = \{x, y, z, \phi, \theta, \psi\} \in \mathbb{R}^6$$

Ecuación 2.11. Coordenadas generalizadas del cuadricóptero

Ecuaciones de Euler-Lagrange[31],[32],[33]

La función langraniana se describe como la energía cinética total menos la energía potencial total, definida por:

$$L(q, \dot{q}) = T - V,$$

Ecuación 2.12. Función langraniana

donde:

L Es la función Langraniana

T Es la energía cinética total

V Es la energía potencial total

q Es la coordenada generalizada

$\dot{q}$ Es la primera derivada de la coordenada generalizada

Dado que la traslación de $\xi = \{x, y, z\}$ es independiente de la rotación de $\eta = \{\phi, \theta, \varphi\}$ y el Langraniano no contiene energía cinética combinacional, la dinámica del sistema se divide en dinámica traslacional y rotacional.

$$\begin{bmatrix} F_\varepsilon \\ T_\eta \end{bmatrix} = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q}$$

Ecuación 2.13. Dinámica traslacional y rotacional del sistema

Donde F_ε es la fuerza traslacional aplicada al cuadricóptero y T_η son los momentos angulares de roll, pitch y yaw.

Dinámica Traslacional

Define la dinámica del sistema debido a los movimientos que describen $\xi = \{x, y, z\}$ [28]. La energía traslacional cinética, donde m es la masa del cuadricóptero, es:

$$T_{tras} = \frac{1}{2} m \dot{\xi}^T \dot{\xi}$$

Ecuación 2.14. Energía traslacional cinética

La energía potencial se describe como:

$$V = mgz,$$

Ecuación 2.15. Energía potencial

donde:

g: Es la gravedad

z: Es la altura

De la Ecuación 2.12, se conoce que el langraniano traslacional se describe como:

$$L_{tras}(\xi, \dot{\xi}) = T_{tras} - V$$

Ecuación 2.16. Langraniano traslacional simplificado

Sustituyendo la Ecuación 2.14 y la Ecuación 2.15 en la Ecuación 2.16, se tiene:

$$L_{tras}(\xi, \dot{\xi}) = \frac{1}{2}\, m\dot{\xi}^T \dot{\xi} - \, mgz$$

Remplazando las coordenadas traslacionales, se tiene:

$$L_{tras}(\xi, \dot{\xi}) = \frac{1}{2}\, m\, [\dot{x}\ \ \dot{y}\ \ \dot{z}] \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} - \, mgz$$

Resolviendo el producto de matrices, el langraniano traslacional queda definido, como:

$$L_{tras}(\xi, \dot{\xi}) = \frac{1}{2}\, m\, (\dot{x}^2 + \dot{y}^2 + \dot{z}^2) - \, mgz$$

Ecuación 2.17. Langraniano traslacional

De la Ecuación 2.13 de Euler -Lagrange:

$$F_{\xi} = \frac{d}{dt}\left(\frac{\partial L_{tras}}{\partial \dot{\xi}}\right) - \frac{\partial L_{tras}}{\partial \xi}$$

Ecuación 2.18. Dinámica de la fuerza traslacional

Derivando la Ecuación 2.17, se obtiene:

$$\frac{d}{dt}\left(\frac{\partial L_{tras}(\xi, \dot{\xi})}{\partial \dot{\xi}}\right) = \ \ m\,\ddot{x} + m\,\ddot{y} + m\,\ddot{z}$$

$$\frac{\partial L(\xi, \dot{\xi})}{\partial \xi} = -mg$$

Sustituyendo las derivadas en la Ecuación 2.18, se aprecia:

$$F_\xi = m\,\ddot{x} + m\,\ddot{y} + m\,\ddot{z} + mg$$

Agrupando:

$$F_\xi = m\,(\ddot{x} + \ddot{y} + \ddot{z}) + mg$$

Remplazando las coordenadas traslacionales, se tiene:

$$F_\xi = m\ddot{\xi} + mgE_z$$

Donde $E_z = \begin{bmatrix}0\\0\\1\end{bmatrix}$, entonces:

$$F_\xi = m\begin{bmatrix}\ddot{x}\\\ddot{y}\\\ddot{z}\end{bmatrix} + mg\begin{bmatrix}0\\0\\1\end{bmatrix}$$

Ecuación 2.19. Dinámica de la fuerza traslacional en función de las aceleraciones

F_ξ es la fuerza suministrada al cuadricóptero en el sistema de referencia I, y se la obtiene mediante el producto de la matriz de rotación completa reducida $R(\psi,\theta,\phi)$ y la fuerza lineal aplicada por cada hélice ($\hat{F}$).

Se tiene entonces:

$$\hat{F} = \begin{bmatrix}0\\0\\f_1 + f_2 + f_3 + f_4\end{bmatrix} = \begin{bmatrix}0\\0\\U_1\end{bmatrix}$$

Ecuación 2.20. Fuerza lineal aplicada a cada eje

Donde f_i son las fuerzas de empuje suministradas a cada hélice, y U_1es la sumatoria de las fuerzas de empuje.

Resolviendo F_ξ se tiene:

$$F_\xi = R(\psi,\theta,\phi)\begin{bmatrix}0\\0\\U_1\end{bmatrix}$$

Ecuación 2.21. Fuerza traslacional suministrada al cuadricóptero

Planteando el producto de las matrices correspondientes:

$$F_\xi = \begin{bmatrix}\cos\psi\ \cos\theta & -\sin\psi\cos\phi + \cos\psi\sin\theta\sin\phi & \sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi\\ \sin\psi\ \cos\ \theta & \cos\psi\cos\phi + \sin\psi\sin\theta\sin\phi & -\sin\phi\cos\psi + \cos\phi\sin\psi\sin\theta\\ -\sin\theta & \sin\phi\cos\theta & \cos\theta\cos\phi\end{bmatrix}\begin{bmatrix}0\\0\\U_1\end{bmatrix}$$

Ecuación 2.22. Planteamiento de la fuerza traslacional suministrada al cuadricóptero

Sustituyendo la Ecuación 2.19 en la Ecuación 2.22, se presenta:

$$m\begin{bmatrix}\ddot{x}\\\ddot{y}\\\ddot{z}\end{bmatrix} + mg\begin{bmatrix}0\\0\\1\end{bmatrix} = \begin{bmatrix}(\sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi\ )U_1\\(-\sin\phi\cos\psi + \cos\phi\sin\psi\sin\theta)U_1\\\cos\theta\cos\phi\ \ U_1\end{bmatrix}$$

Igualando términos:

$$m\ddot{x} = sin\,\psi\,sin\,\phi + cos\,\psi\,sin\,\theta\,cos\,\phi\;\;U_1$$
$$m\ddot{y} = cos\,\phi\,sin\,\psi\,sin\,\theta - sin\,\phi\,cos\,\psi\;\;U_1$$
$$m\ddot{z} + mg = cos\,\theta\,cos\,\phi\;\;U_1$$

Finalmente, el modelo que describe la dinámica traslacional del cuadricóptero es:

$$\ddot{x} = \frac{\sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi}{m}\;\;U_1$$

Ecuación 2.23. Dinámica de la aceleración en X del cuadricóptero

$$\ddot{y} = \frac{\cos\phi\sin\psi\sin\theta - \sin\phi\cos\psi}{m}\;\;U_1$$

Ecuación 2.24. Dinámica de la aceleración en Y del cuadricóptero

$$\ddot{z} = -g + \frac{\cos\theta\cos\phi}{m}\;\;U_1$$

Ecuación 2.25. Dinámica de la aceleración en Z del cuadricóptero

Dinámica Rotacional

Para describir la dinámica rotacional recurrimos a la ecuación de Euler-Lagrange que describe los momentos angulares de roll, pitch y yaw. [28]

$$\tau_\eta = \frac{d}{dt}\left(\frac{\partial Lrot}{\partial\dot{\eta}}\right) - \frac{\partial Lrot}{\partial\eta}$$

Ecuación 2.26. Dinámica de la fuerza rotacional

Para simplificar su desarrollo, se ha dividido la Ecuación 2.26 en dos partes, donde:

$$\frac{\partial Lrot}{\partial\dot{\eta}} = \Omega^T I\frac{\partial\Omega}{\partial\dot{\eta}}$$

Ecuación 2.27. Dinámica rotacional parte 1

Y

$$\frac{\partial Lrot}{\partial\eta} = \Omega^T I\frac{\partial\Omega}{\partial\eta}$$

Ecuación 2.28. Dinámica rotacional parte 2

Se sabe que I es la matriz de inercias explicada más adelante, y además Ω es el vector de velocidad angular del cuadricóptero $\Omega = [p \quad q \quad r]^T$ y para enlazarlo con la velocidad de los ángulos de rotación $\dot{\eta} = [\dot{\phi} \quad \dot{\theta} \quad \dot{\psi}]^T$, se utiliza la relación:

$$\Omega = W_\eta\,\dot{\eta}$$

Ecuación 2.29. Relación entre Ω y $\dot{\eta}$

Donde W_η es el Jacobiano respecto al sistema de referencia B y se define como:

$$W_\eta = \begin{bmatrix} 1 & 0 & -sin\theta \\ 0 & cos\,\phi & sin\,\phi cos\theta \\ 0 & -sin\,\phi & cos\,\phi\, cos\theta \end{bmatrix}$$

Ecuación 2.30. Jacobiano respecto al sistema B

Reemplazando la Ecuación 2.30 en la Ecuación 2.29, se obtiene:

$$\Omega = \begin{bmatrix} 1 & 0 & -sin\theta \\ 0 & cos\,\phi & sin\,\phi cos\theta \\ 0 & -sin\,\phi & cos\,\phi\, cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

Resolviendo el producto de matrices, tenemos:

$$\Omega = \begin{bmatrix} \dot{\phi} - \dot{\psi} sin\theta \\ \dot{\theta}\ cos\,\phi + \ \dot{\psi}\ sin\,\phi cos\theta \\ -\dot{\theta}\ sin\,\phi + \ \dot{\psi}\, cos\,\phi\, cos\theta \end{bmatrix}$$

Ecuación 2.31. Velocidad angular del cuadricóptero

Derivando la Ecuación 2.31, respecto a $\dot{\eta}$:

$$\frac{d\Omega}{\partial \dot{\eta}} = \begin{bmatrix} 1 & 0 & -sin\theta \\ 0 & cos\,\phi & sin\,\phi cos\theta \\ 0 & -sin\,\phi & cos\,\phi\, cos\theta \end{bmatrix}$$

Ecuación 2.32. Derivada de la velocidad angular respecto a $\dot{\eta}$

Derivando la Ecuación 2.31, respecto a η :

$$\frac{d\Omega}{\partial \eta} = \begin{bmatrix} 0 & -\dot{\psi}\, cos\theta & 0 \\ -\dot{\theta}\ sin\phi + \dot{\psi} cos\,\phi\, cos\theta & -\dot{\psi}\ sin\,\phi sin\theta & 0 \\ -\dot{\theta}\ cos\,\phi - \ \dot{\psi}\ sin\,\phi cos\theta & -\dot{\psi}\ cos\,\phi sin\theta & 0 \end{bmatrix}$$

Ecuación 2.33. Derivada de la velocidad angular respecto a η

Considerando que la estructura del cuadricóptero es simétrica, se tiene la matriz de inercias I dada por:

$$I = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix}$$

Ecuación 2.34. Matriz de inercias del cuadricóptero

Donde

I_x : Es la Inercia en el eje X

I_y : Es la Inercia en el eje Y

I_z : Es la Inercia en el eje Z

Sustituyendo la Ecuación 2.31, la Ecuación 2.34 y la Ecuación 2.32 en la Ecuación 2.27, se obtiene:

$$\Omega^T I \frac{\partial \Omega}{\partial \dot{\eta}} = \begin{bmatrix} \dot{\phi} - \dot{\psi} sin\theta \\ \dot{\theta}\ cos\phi + \dot{\psi}\ sin\phi cos\theta \\ -\dot{\theta}\ sin\phi + \dot{\psi} cos\phi\ cos\theta \end{bmatrix}^T \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} 1 & 0 & -sin\theta \\ 0 & cos\phi & sin\phi cos\theta \\ 0 & -sin\phi & cos\phi\ cos\theta \end{bmatrix}$$

Finalmente tenemos:

$$\Omega^T I \frac{\partial \Omega}{\partial \dot{\eta}} = \begin{bmatrix} I_x(\dot{\phi} - \dot{\psi} sin\theta\ ) \\ I_y cos\phi\ (\dot{\theta}\ cos\phi + \dot{\psi}\ sin\phi cos\theta\ ) + I_z sin\phi\ (\dot{\theta}\ sin\phi - \dot{\psi} cos\phi\ cos\theta\ ) \\ I_x sin\theta(\dot{\psi} sin\theta - \dot{\phi}\ ) + I_y sin\phi cos\theta\ (\dot{\theta}\ cos\phi + \dot{\psi}\ sin\phi cos\theta\ ) + \cdots \\ \ldots\ldots + I_z cos\phi\ cos\theta\ (\dot{\psi} cos\phi\ cos\theta - \dot{\theta}\ sin\phi) \end{bmatrix}^T$$

Ecuación 2.35. Parte 1 de la dinámica rotacional desarrollada

De forma similar, sustituyendo la Ecuación 2.31, la Ecuación 2.34 y la Ecuación 2.33 en la Ecuación 2.28, se plantea:

$$\Omega^T I \frac{\partial \Omega}{\partial \eta} = \begin{bmatrix} \dot{\phi} - \dot{\psi} sin\theta \\ \dot{\theta}\ cos\phi + \dot{\psi}\ sin\phi cos\theta \\ -\dot{\theta}\ sin\phi + \dot{\psi} cos\phi\ cos\theta \end{bmatrix}^T \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} 0 & -\dot{\psi} cos\theta & 0 \\ -\dot{\theta}\ sin\phi + \dot{\psi} cos\phi\ cos\theta & -\dot{\psi}\ sin\phi sin\theta & 0 \\ -\dot{\theta}\ cos\phi - \dot{\psi}\ sin\phi cos\theta & -\dot{\psi}\ cos\phi sin\theta & 0 \end{bmatrix}$$

Desarrollando por términos, se obtiene:

$$\Omega^T I \frac{\partial \Omega}{\partial \eta} = [a_1 \quad a_2 \quad a_3]$$

$a_1 = I_y(\dot{\theta}\ cos\phi + \dot{\psi}\ sin\phi cos\theta)(\dot{\psi} cos\phi\ cos\theta - \dot{\theta}\ sin\phi) + I_z(-\dot{\theta}\ sin\phi + \dot{\psi} cos\phi\ cos\theta)(-\dot{\theta}\ cos\phi - \dot{\psi}\ sin\phi cos\theta)$

$a_2 = I_x \dot{\psi} cos\theta(\dot{\psi} sin\theta - \dot{\phi}) + I_y(\dot{\theta}\ cos\phi + \dot{\psi}\ sin\phi cos\theta)(-\dot{\psi}\ sin\phi sin\theta) + I_z(-\dot{\theta}\ sin\phi + \dot{\psi} cos\phi\ cos\theta)(-\dot{\psi}\ cos\phi sin\theta)$

$a_3 = 0$

Derivando la Ecuación 2.35 respecto al tiempo, se obtiene:

$$\frac{d}{dt}\left(\frac{\partial Lrot}{\partial \dot{\eta}}\right) = [b_1 \quad b_2 \quad b_3]$$

$b_1 = I_x(\ddot{\phi} - \ddot{\psi} sin\theta - \dot{\psi}\dot{\theta} cos\theta)$

$$b_2 = I_y\left(\ddot{\theta}\, cos\phi^2 - 2\dot{\theta}\,\dot{\phi}Sin\phi cos\phi + \ddot{\psi}sin\phi cos\theta cos\phi + \dot{\psi}\dot{\phi}cos\phi^2 cos\theta - \dot{\psi}\dot{\theta}\, sin\phi sin\theta cos\phi - \dot{\psi}\dot{\phi}\, sin\phi^2 cos\theta\right) + I_z(\ddot{\theta}sin\phi^2 + 2\dot{\theta}\dot{\phi}cos\phi sin\phi - \ddot{\psi}cos\phi cos\theta sin\phi + \dot{\psi}\dot{\phi}sin\phi^2 cos\theta + \dot{\psi}\dot{\theta}cos\phi\, sin\theta\, sin\phi - \dot{\psi}\dot{\phi}\, cos\phi^2 cos\theta)$$

$$b_3 = I_x\left(-\ddot{\phi}sin\theta - \dot{\phi}\dot{\theta}cos\theta + \ddot{\psi}sin\theta^2 + 2\dot{\psi}\dot{\theta}sin\theta cos\theta\right) + I_y\left(\ddot{\theta}cos\phi sin\phi cos\theta - \dot{\theta}\dot{\phi}sin\phi^2 cos\theta + \dot{\theta}\dot{\phi}cos\phi^2 cos\theta - \dot{\theta}^2 cos\phi sin\phi sin\theta + \ddot{\psi}sin\phi^2 cos\theta^2 + 2\dot{\psi}\dot{\phi}sin\phi cos\phi cos\theta^2 - 2\dot{\psi}\dot{\theta}sin\phi^2 sin\theta\, cos\theta\right) + I_z(-\ddot{\theta}sin\phi cos\phi cos\theta - \dot{\theta}\dot{\phi}cos\phi^2 cos\theta + \dot{\theta}\dot{\phi}sin\phi^2 cos\theta + \dot{\theta}^2 cos\phi sin\phi sin\theta + \ddot{\psi}cos\phi^2 cos\theta^2 - 2\dot{\psi}\dot{\phi}\, sin\phi cos\phi cos\theta^2 - 2\dot{\psi}\dot{\theta}cos\phi^2 cos\theta sin\theta)$$

Transcribiendo la Ecuación 2.26, tenemos:

$$\tau_\eta = \frac{d}{dt}\left(\frac{\partial Lrot}{\partial \dot{\eta}}\right) - \frac{\partial Lrot}{\partial \eta}$$

$$\tau_\eta = [b_1 \quad b_2 \quad b_3] - [a_1 \quad a_2 \quad a_3] = [\tau_\phi \quad \tau_\theta \quad \tau_\psi]$$

Ecuación 2.36. Resta de términos para la obtención de torques

Realizando una resta de términos presentados en la Ecuación 2.36 y considerando que los ángulos de roll, pitch y yaw permanecen muy cercanos a cero durante el tiempo de vuelo, se establece que $\phi \to 0$, $\theta \to 0\ y\ \psi \to 0$ por lo tanto, $\cos\phi \cong 1,\ \cos\theta \cong 1\ , cos\,\psi \cong 1$ y $\sin\phi \cong 0,\ \sin\theta \cong 0, \sin\psi \cong 0.$

Simplificando se obtiene:

$$\begin{bmatrix}\tau_\phi \\ \tau_\theta \\ \tau_\psi\end{bmatrix} = \begin{bmatrix} I_x\left(\ddot{\phi} - \dot{\psi}\dot{\theta}\right) - (I_y\dot{\psi}\dot{\theta} - I_z\dot{\psi}\dot{\theta}) \\ I_y\ddot{\theta} + \dot{\psi}\dot{\phi}\, I_y - \dot{\psi}\dot{\phi}I_z - (-I_x\dot{\psi}\dot{\phi}\,) \\ I_x\left(-\dot{\theta}\dot{\phi}\right) + Iy\left(\dot{\theta}\dot{\phi}\right) + I_z(-\dot{\theta}\dot{\phi} + \ddot{\psi})\end{bmatrix}$$

$$\tau_\eta = \begin{bmatrix}\tau_\phi \\ \tau_\theta \\ \tau_\psi\end{bmatrix} = \begin{bmatrix} I_x\left(\ddot{\phi} - \dot{\psi}\dot{\theta}\right) - \dot{\psi}\dot{\theta}\left(I_y - I_z\right) \\ I_y\left(\ddot{\theta} + \dot{\psi}\dot{\phi}\right) - \dot{\psi}\dot{\phi}(I_z - I_x) \\ I_z\left(\ddot{\psi} - \dot{\theta}\dot{\phi}\right) - \dot{\theta}\dot{\phi}(I_x - I_y)\end{bmatrix}$$

Ecuación 2.37. Torques rotacionales

Donde τ_ϕ, τ_θ y τ_ψ son los torques de roll, pitch y yaw, los torques $\tau_\phi\ y\ \tau_\theta$ dependen de las fuerzas lineales de empuje, mientras que el torque τ_ψ depende de la sumatoria de los torques producidos por cada motor τ_{mi}.[6]

Para la obtención del torque en roll τ_ϕ , torque en pitch τ_θ, y el torque en yaw τ_ψ, según la disposición de los rotores como muestra la Figura 2.3, se realiza el procedimiento explicado en la mecánica del cuadricóptero de la sección 1.3.

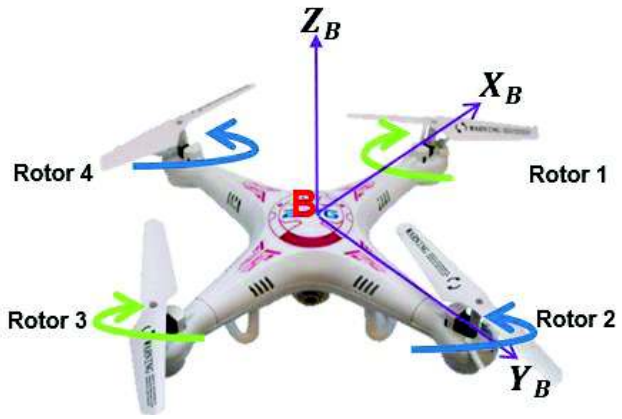


Figura 2.3. Disposición de los rotores en el cuadricóptero

Reescribiendo la obtención de torques de forma matemática y sabiendo que cada movimiento debe ejecutarse con una fuerza principal U_2, U_3, U_4, se tiene:

$$\tau_\eta = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} bl(-{\Omega_2}^2 + {\Omega_4}^2) \\ bl(-{\Omega_1}^2 + {\Omega_3}^2) \\ d(-{\Omega_1}^2 + {\Omega_2}^2 - {\Omega_3}^2 + {\Omega_4}^2) \end{bmatrix} = \begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix}$$

Ecuación 2.38. Obtención de torques y empujes del cuadricóptero

Donde $\Omega_1, \Omega_2, \Omega_3, \Omega_4$ son las velocidades angulares de los rotores, b representa el factor de empuje, d se define como el factor de arrastre y l es la distancia en metros desde el centro de masa del cuadricóptero hasta la hélice de cada rotor.

Sustituyendo la Ecuación 2.37 en la Ecuación 2.38, se obtiene:

$$\tau_\eta = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} I_x(\ddot{\phi} - \dot{\psi}\dot{\theta}) - \dot{\psi}\dot{\theta}(I_y - I_z) \\ I_y(\ddot{\theta} + \dot{\psi}\dot{\phi}) - \dot{\psi}\dot{\phi}(I_z - I_x) \\ I_z(\ddot{\psi} - \dot{\theta}\dot{\phi}) - \dot{\theta}\dot{\phi}(I_x - I_y) \end{bmatrix} = \begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix}$$

Reescribiendo, se tiene:

$$\ddot{\phi} = \dot{\psi}\dot{\theta}\left(\frac{I_y - I_z}{I_x}\right) + \frac{U_2}{I_x} + \dot{\psi}\dot{\theta}$$

Ecuación 2.39. Dinámica de la aceleración en roll del cuadricóptero

$$\ddot{\theta} = \dot{\psi}\dot{\phi}\left(\frac{I_z - I_x}{I_y}\right) + \frac{U_3}{I_y} - \dot{\psi}\dot{\phi}$$

Ecuación 2.40. Dinámica de la aceleración en pitch del cuadricóptero

$$\ddot{\psi} = \dot{\theta}\dot{\phi}\left(\frac{I_x - I_y}{I_z}\right) + \frac{U_4}{I_z} + \dot{\theta}\dot{\phi}$$

Ecuación 2.41. Dinámica de la aceleración en pitch del cuadricóptero

Debido al giro de las hélices con velocidad angular relativa $\Omega = [p \quad q \quad r]^T$, respecto a otro sistema de referencia con velocidad rotacional $\eta = [\phi \quad \theta \quad \psi]^T$, se produce un momento conocido como efecto giroscópico que ocurre en los ejes X y Y, por lo tanto, se tiene:

$$\tau_{giroscopico} = \begin{bmatrix} -J_r\,\dot{\theta}\Omega_r \\ J_r\,\dot{\phi}\Omega_r \\ 0 \end{bmatrix}$$

Ecuación 2.42. Efectos giroscópicos en X y Y

Donde J_r se describe como la inercia rotacional producida por las hélices y Ω_r representa la velocidad total de las hélices, donde:

$$\Omega_r = -{\Omega_1}^2 + {\Omega_2}^2 - {\Omega_3}^2 + {\Omega_4}^2$$

Ecuación 2.43. Velocidad total de las hélices

Al añadir el efecto giroscópico, se obtiene finalmente el modelo dinámico rotacional del cuadricóptero:

$$\ddot{\phi} = \dot{\psi}\dot{\theta}\left(\frac{I_y - I_z}{I_x}\right) + \frac{U_2}{I_x} + \dot{\psi}\dot{\theta} - \frac{J_r\,\dot{\theta}\Omega_r}{I_x}$$

Ecuación 2.44. Dinámica rotacional de la aceleración en roll

$$\ddot{\theta} = \dot{\psi}\dot{\phi}\left(\frac{I_z - I_x}{I_y}\right) + \frac{U_3}{I_y} - \dot{\psi}\dot{\phi} + \frac{J_r\,\dot{\phi}\Omega_r}{I_y}$$

Ecuación 2.45. Dinámica rotacional de la aceleración en pitch

$$\ddot{\psi} = \dot{\theta}\dot{\phi}\left(\frac{I_x - I_y}{I_z}\right) + \frac{U_4}{I_z} + \dot{\theta}\dot{\phi}$$

Ecuación 2.46. Dinámica rotacional de la aceleración en yaw

Se presentan las ecuaciones que describen el modelo dinámico del cuadricóptero:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \dfrac{\sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi}{m}\ U_1 \\ \dfrac{\cos\phi\sin\psi\sin\theta - \sin\phi\cos\psi}{m}\ U_1 \\ -g + \dfrac{\cos\theta\cos\phi}{m}\ U_1 \\ \dot{\psi}\dot{\theta}\left(\dfrac{I_y - I_z}{I_x}\right) + \dfrac{U_2}{I_x} + \dot{\psi}\dot{\theta} - \dfrac{J_r\,\dot{\theta}\Omega_r}{I_x} \\ \dot{\psi}\dot{\phi}\left(\dfrac{I_z - I_x}{I_y}\right) + \dfrac{U_3}{I_y} - \dot{\psi}\dot{\phi} + \dfrac{J_r\,\dot{\phi}\Omega_r}{I_y} \\ \dot{\theta}\dot{\phi}\left(\dfrac{I_x - I_y}{I_z}\right) + \dfrac{U_4}{I_z} + \dot{\theta}\dot{\phi} \end{bmatrix}$$

Ecuación 2. 47 Modelo dinámico del cuadricóptero

2.2 Algoritmos de Control

En esta sección se desarrollan las leyes de control para la dinámica del cuadricóptero, el control de orientación y altitud son diseñados mediante SMC, mientras que para el control de seguimiento de trayectoria se opta por una técnica clásica como PID y una técnica robusta como SMC. A continuación, se presenta un esquema de control del sistema.

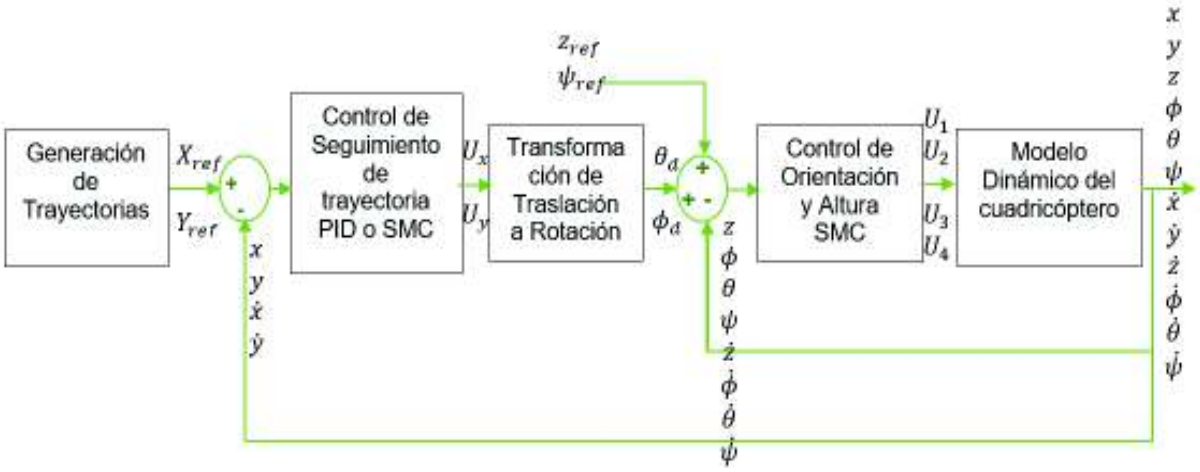


Figura 2.4. Esquema de Control del Sistema

Del esquema mostrado en la Figura 2.4, se observa que en el lazo interno se realiza un control para la altura (z) y orientación del cuadricóptero (ϕ, θ, ψ), mediante SMC y en el segundo lazo o lazo externo se controla la trayectoria en (x, y), donde las referencias de los ángulos de roll y pitch (ϕ_d, θ_d), son determinadas automáticamente por los desplazamientos en (x, y), convirtiéndose en un control en cascada.

Control de Orientación y Altitud por Modos Deslizantes (SMC)

Dado que el SMC o Control por Modos Deslizantes presenta una gran robustez al trabajar con sistemas no lineales, se utiliza esta técnica para diseñar el control para los ángulos de roll, pitch y yaw, que determina la orientación del cuadricóptero, y el control en z que describe la altura deseada del cuadricóptero. El diseño inicia con el planteamiento de la superficie deslizante, la obtención de la ley de control y posteriormente el análisis de estabilidad de Lyapunov.

Control de Altitud

Se plantea la superficie deslizante en función del error y la variación del error:

$$s(t) = \left(\frac{d}{dt} + \lambda\right)^n e(t)dt$$

Ecuación 2.48. Superficie deslizante de orden n

Se escoge una superficie de primer orden y en función de z, se reescribe:

$$s_z = \dot{e}_z + \lambda e_z$$

Ecuación 2.49. Superficie deslizante de orden 1

El error en z se define como la resta entre la altitud deseada y el valor real del sistema:

$$e_z = z_d - z$$

Ecuación 2.50. Función de error en Z

Derivando la Ecuación 2.50, se obtiene:

$$\dot{e_z} = \dot{z}_d - \dot{z}$$

Ecuación 2.51. Derivada del error en Z

Sustituyendo la Ecuación 2.51 y la Ecuación 2.50 en la Ecuación 2.49, se obtiene la superficie deslizante para la altura:

$$s_z = (\dot{z}_d - \dot{z}) + \lambda(z_d - z)$$

Ecuación 2.52. Superficie deslizante para la altitud

Derivando la Ecuación 2.52, se tiene:

$$\dot{s}_z = 0 = (\ddot{z}_d - \ddot{z}) + \lambda(\dot{z}_d - \dot{z})$$

Ecuación 2.53. Derivada de la superficie deslizante para la altitud

La dinámica que define la altitud del sistema está dada por:

$$\ddot{z} = -g + \frac{\cos\theta\cos\phi}{m} U_1$$

Ecuación 2.54. Dinámica que describe la aceleración de la altitud en Z

Sustituyendo la Ecuación 2.54 en la Ecuación 2.53, se tiene:

$$\dot{s}_z = \left[\ddot{z}_d - \left(-g + \frac{\cos\theta\cos\phi}{m} U_1\right)\right] + \lambda(\dot{z}_d - \dot{z})$$

Ecuación 2.55. Derivada de la superficie deslizante en función de la dinámica de Z

Igualando a 0 la derivada de la superficie, para la parte continua de z, se tiene:

$$\dot{s}_z = 0 = \left[\ddot{z}_d - \left(-g + \frac{\cos\theta\cos\phi}{m} U_1\right)\right] + \lambda(\dot{z}_d - \dot{z})$$

Resolviendo, la parte continua de la ley de control para la altitud queda definida por:

$$U_c(t) = [\ddot{z}_d + \lambda(\dot{z}_d - \dot{z}) + g] \frac{m}{\cos\theta\cos\phi}$$

Ecuación 2.56. Función continua de la ley de control para la altitud

El sistema seguirá la superficie definida y se mantendrá en ella, siempre que se cumpla la condición de deslizamiento:

$$\mu_D(t) = K_{Dz}\frac{S_z}{|S_z| + \delta}$$

Ecuación 2.57. Función discontinua de la ley de control para la altitud

Finalmente, la ley de control para la altitud se describe como:

$$U_1 = [\ddot{z}_d + \lambda(\dot{z}_d - \dot{z}) + g]\ \frac{m}{\cos\theta\cos\phi} + K_{Dz}\frac{S_z}{|S_z| + \delta}$$

Ecuación 2.58. Ley de control SMC para la altitud del cuadricóptero

Para verificar la estabilidad del sistema con la nueva ley de control, se realiza el análisis de estabilidad de Lyapunov, donde se debe cumplir que: la función de Lyapunov evaluada en 0, debe ser igual a cero, la función debe ser positiva y la derivada de la función de Lyapunov debe ser menor a 0. Se plantea como función candidata de Lyapunov a una superficie cuadrática, para garantizar que la función sea positiva:

$$V_z = \frac{1}{2}s_z^2$$

Ecuación 2.59. Candidata de Lyapunov en función del error cuadrático de Z

La candidata de Lyapunov cumple con la condición de $V_z(z_0) = 0$, sabiendo que el punto de equilibrio es cero, se tiene:

$$V_z = \frac{1}{2}s_z^2\ ;\ \ s_{z(0)} = 0\ \rightarrow V_z\big(\ s_{z(0)}\big) = 0$$

La segunda condición debe definir a la función de Lyapunov como positiva:

$$V_z = \frac{1}{2}s_z^2\ > 0; \qquad \forall s_z \neq\ s_{z(0)}$$

Derivando la Ecuación 2.59, se tiene:

$$\dot{V}_z =\ s_z\dot{s}_z\ < 0$$

Considerando la Ecuación 2.55, se tiene:

$$\dot{V}_z =\ s_z\ \left[\left[\ddot{z}_d - \left(-g + \frac{\cos\theta\cos\phi}{m}\ U_1\right)\right] +\ \lambda(\dot{z}_d - \dot{z})\right] < 0$$

Reemplazando la ley de control U_1 de la Ecuación 2.58 y eliminando términos semejantes:

$$\dot{V}_z = -s_z\ \left[K_{Dz}\ \frac{s_z}{|S_z| + \delta}\ \frac{\cos\theta\cos\phi}{m}\right] < 0$$

Reescribiendo se tiene:

$$\dot{V}_z = -s_z^2\ \frac{K_{Dz}}{m}\ \frac{\cos\theta\cos\phi}{|S_z| + \delta}\ < 0$$

Teniendo en cuenta que $m,\ \delta,\ K_{Dz}$ son constantes positivas, que la función $\cos$ es par y que los $\cos\theta$ y $\cos\phi$ son positivos debido a que se consideran ángulos menores a $\frac{\pi}{2}$, entonces, se cumplen las condiciones que garantizan la estabilidad del sistema.

Control para el Sistema de Orientación

Siguiendo el proceso descrito en la sección anterior, se muestran las leyes de control para roll, pitch y yaw, por lo que se definen las siguientes ecuaciones para el control de orientación:

$$U_2 = \left[\ddot{\phi}_d - \dot{\psi}\dot{\theta}\left(\frac{I_y - I_z}{I_x}\right) - \dot{\psi}\dot{\theta} + \frac{J_r\,\dot{\theta}\Omega_r}{I_x} + \lambda\left(\dot{\phi}_d - \dot{\phi}\right)\right] I_x + K_{D\phi}\frac{S_\phi}{|S_\phi| + \delta}$$

Ecuación 2.60. Ley de control SMC para la orientación en roll del cuadricóptero

$$U_3 = \left[\ddot{\theta}_d - \dot{\psi}\dot{\phi}\left(\frac{I_z - I_x}{I_y}\right) + \dot{\psi}\dot{\phi} - \frac{J_r\,\dot{\phi}\Omega_r}{I_y} + \lambda\left(\dot{\theta}_d - \dot{\theta}\right)\right] I_y \;\; + K_{D\theta}\frac{S_\theta}{|S_\theta| + \delta}$$

Ecuación 2.61. Ley de control SMC para la orientación en pitch del cuadricóptero

$$U_4 = \left[\ddot{\psi}_d - \dot{\phi}\dot{\theta}\left(\frac{I_x - I_y}{I_z}\right) - \dot{\phi}\dot{\theta} + \lambda\left(\dot{\psi}_d - \dot{\psi}\right)\right] I_z + K_{D\psi}\frac{S_\psi}{|S_\psi| + \delta}$$

Ecuación 2. 62. Ley de control SMC para la orientación en yaw del cuadricóptero

Control para el Seguimiento de Trayectorias

El seguimiento de trayectorias del cuadricóptero se establece en el sistema de referencia fijo a tierra (I), por lo que, se determinan los valores deseados en x y y. El control realizado mediante SMC para z, roll, pitch y yaw, garantiza que el cuadricóptero esté de forma horizontal en el plano XY, para alcanzar las referencias deseadas en x y y de la trayectoria definida.

Control para el Seguimiento de Trayectorias mediante PID

La Figura 2.5 muestra el esquema de Control PID aplicado al seguimiento de trayectorias de un cuadricóptero, el cual consta de tres acciones: la acción proporcional K_p, que depende del error actual, la acción integral K_i que depende de los errores pasados y la acción derivativa K_d, que depende de los errores futuros.[26] Las leyes de control correspondientes a X y Y, están determinadas por:

$$U_x = K_p(x_d - x) + \; K_i \int (x_d - x)\,dt + K_d\,\frac{d(x_d - x)}{dt}$$

Ecuación 2.63. Ley de control PID para el seguimiento de trayectoria en X del cuadricóptero

$$U_y = K_p(y_d - y) + \; K_i \int (y_d - y)\,dt + K_d\,\frac{d(y_d - y)}{dt}$$

Ecuación 2.64. Ley de control PID para el seguimiento de trayectoria en Y del cuadricóptero

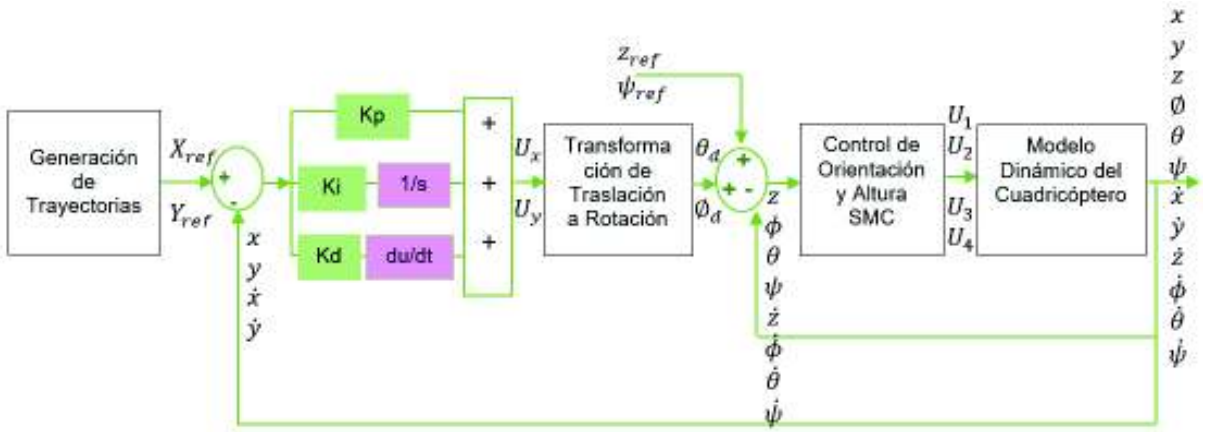


Figura 2.5. Esquema del Control de Seguimiento de Trayectoria de un Cuadricóptero mediante PID

Control para el Seguimiento de Trayectorias mediante SMC

Para definir las leyes de control para el seguimiento de trayectoria, se sigue el mismo proceso empleado para el control de orientación y altitud. A continuación, se muestra la dinámica traslacional para X y Y, respectivamente.

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} \dfrac{\sin\psi \sin\phi + \cos\psi \sin\theta \cos\phi}{m} U_1 \\ \dfrac{\cos\phi \sin\psi \sin\theta - \sin\phi \cos\psi}{m} U_1 \end{bmatrix}$$

Ecuación 2.65. Dinámica traslacional para X y Y

Para el desarrollo del control traslacional se considera $U_x = \sin\psi \sin\phi + \cos\psi \sin\theta \cos\phi$ y $U_y = \cos\phi \sin\psi \sin\theta - \sin\phi \cos\psi$, [28],[34],[35], por lo que reescribiendo la dinámica traslacional, se tiene:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} U_x \dfrac{U_1}{m} \\ U_y \dfrac{U_1}{m} \end{bmatrix}$$

Ecuación 2.66. Dinámica traslacional en función de U_x y U_y

Teniendo en cuenta estas consideraciones se presentan las superficies deslizantes para x y y, respectivamente.

$$s_x = (\dot{x}_d - \dot{x}) + \lambda(x_d - x)$$

Ecuación 2.67. Superficie deslizante para la trayectoria en X

$$s_y = (\dot{y}_d - \dot{y}) + \lambda(y_d - y)$$

Ecuación 2.68. Superficie deslizante para la trayectoria en Y

Las condiciones de deslizamiento que hacen que el sistema permanezca en las superficies deslizantes, son:

$$U_{D_x} = K_{Dx} \frac{S_x}{|S_x| + \delta}$$

Ecuación 2.69. Función discontinua para la ley de control SMC de X

$$U_{D_y} = K_{Dy} \frac{S_y}{|S_y| + \delta}$$

Ecuación 2.70. Función discontinua para la ley de control SMC de Y

Finalmente, se presentan las leyes de control para el seguimiento de trayectoria del UAV:

$$U_x = [\ddot{x}_d + \lambda(\dot{x_d} - \dot{x})] \frac{m}{U_1} + K_{Dx} \frac{S_x}{|S_x| + \delta}$$

Ecuación 2.71. Ley de control SMC para la trayectoria en X del cuadricóptero

$$U_y = [\ddot{y}_d + \lambda(\dot{y_d} - \dot{y})] \frac{m}{U_1} + K_{Dy} \frac{S_y}{|S_y| + \delta}$$

Ecuación 2.72. Ley de control SMC para la trayectoria en Y del cuadricóptero

Transformación del Sistema de Traslación a Rotación

El sistema de seguimiento de trayectoria del cuadricóptero recibe cuatro variables de entrada(x, y, z, ψ) y entrega 12 estados o variables de salidas $(x, y, z, \dot{x}, \dot{y}, \dot{z}, \phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi})$, por lo que solo nos es posible controlar cuatro de los 12 estados (x, y, z, ψ), es decir, a cada una de estas variables se les puede asignar valores de referencia, mientras que los ángulos de θ, ϕ están implícitamente controlados por las referencias de x y y , respectivamente . [36]

Las señales de control obtenidas de la posición (U_x, U_y) se encuentran en el sistema de referencia en tierra (I), y para ser leídas por el cuadricóptero deben ser transformadas al sistema de referencia del centro de masa (B), como se muestra en la Figura 2.6, tal como se había detallado en el modelado del cuadricóptero para generar un movimiento en roll (eje X), se deben descompensar las fuerzas de los rotores 2 y 4, que se encuentran en el eje Y , por lo tanto la señal U_x controlará la referencia del ángulo de pitch θ, y la señal U_y controlará la referencia del ángulo de roll ϕ.[28]

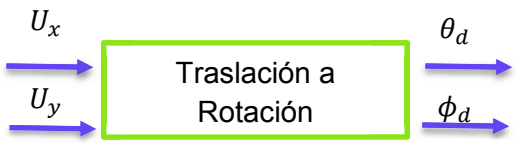


Figura 2.6. Esquema de transformación de Traslación a Rotación

Para realizar esta transformación, se utiliza la matriz de transformación de rotación de yaw respecto a z. [37]

$$R(z,\psi) = \begin{bmatrix} \cos(\psi) & -sin(\psi) \\ sin(\psi) & \cos(\psi) \end{bmatrix}$$

Ecuación 2.73. Matriz de rotación de yaw respecto a Z

La matriz de la Ecuación 2.73, sirve para transformar del sistema del centro de masa B, al sistema fijo en tierra I, por lo cual se realiza la transformación inversa:

$$R(\psi,z) = R(z,\psi)^{-1}$$

Resolviendo se tiene:

$$R(\psi,z) = \begin{bmatrix} \cos(\psi) & sin(\psi) \\ -sin(\psi) & \cos(\psi) \end{bmatrix}$$

Ecuación 2.74. Matriz de rotación de Z respecto a yaw

Finalmente, los ángulos de roll (ϕ) y pitch (θ) deseados, para generar los movimientos de x y y, en función de las señales U_x y U_y, están dados por:

$$\begin{bmatrix} \theta_d \\ \phi_d \end{bmatrix} = \begin{bmatrix} \cos(\psi) & sin(\psi) \\ -sin(\psi) & \cos(\psi) \end{bmatrix} \begin{bmatrix} U_x \\ U_y \end{bmatrix}$$

Ecuación 2.75. Ángulos deseados para pitch y roll en función de las leyes de control

2.3 Algoritmos Genéticos

La idea fundamental de los algoritmos genéticos consiste en encontrar una solución aceptable a un problema por medio del mejoramiento de un conjunto de individuos, cuya función de evaluación corresponde a una solución del problema. Esta optimización se realiza mediante procesos selectivos y de intercambio de información genética. [38] Se debe tener en cuenta que los Algoritmos Genéticos como herramienta de optimización, son independientes del problema y para entender el proceso de actuación de los AG, se revisará su aplicación a una planta de primer orden.

Sintonización de un PID para una planta de Primer Orden con AG

Para sintonizar controladores mediante Algoritmos Genéticos, se debe definir una función aptitud o función objetivo, mediante la cual se evalúen los individuos (parámetros de sintonización), el objetivo es encontrar mediante esta técnica de computación evolutiva parámetros que consigan que el error sea el mínimo posible, para ello en el presente trabajo

se opta por utilizar como función objetivo el ISE del sistema como se observa en la Figura 2.7, que es un índice de desempeño cuantitativo, al encontrar parámetros que optimicen el ISE, se está obteniendo un error de posición mínimo. De forma general en la sección 1.3, se presentó una estructura de algoritmo genético básico, es decir, para adaptar el algoritmo genético para la optimización del ISE y búsqueda de los parámetros óptimos de un controlador, se debe tener en cuenta en el caso del PID y SMC, será un AG multivariable para conseguir optimizar K_p, K_i, K_d, y λ y K_D, respectivamente.[39]

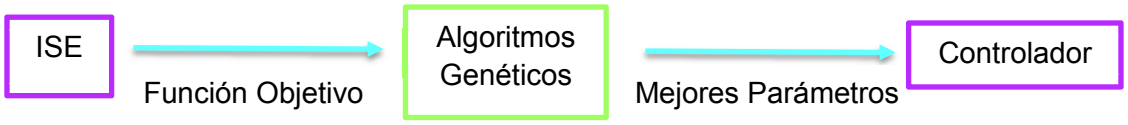


Figura 2.7. Esquema de Sintonización de un Controlador Mediante Algoritmos Genéticos

Se analiza una planta de primer orden con retardo, que representa la dinámica de un reactor, se escogió esta planta debido a la simplicidad que expone su función de transferencia, descrita por:

$$Gp(s) = \frac{0.78}{2.32s + 1}\, e^{-2.97s}$$

Ecuación 2.76. Función de transferencia de un reactor

Se diseñó un controlador PID mediante curva de reacción, y posteriormente mediante Algoritmos Genéticos se sintonizó el mismo controlador, obteniendo la siguiente gráfica comparativa de la salida del sistema.

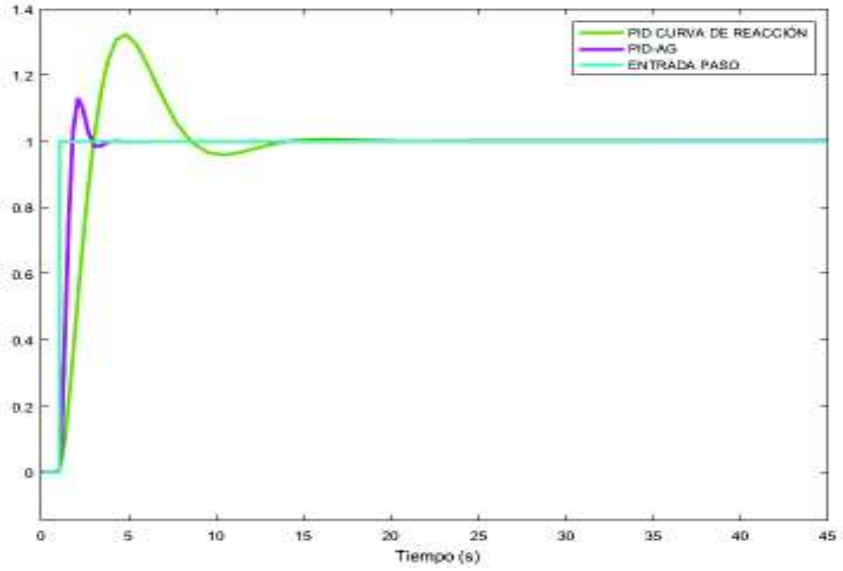


Figura 2.8. Gráfica Comparativa de un PID mediante Curva de Reacción y AG

En la Figura 2.8, se puede observar que el tiempo de establecimiento y sobre impulso, son menores en la salida del sistema sintonizado mediante AG, debido a que los Algoritmos Genéticos cumplen con la tarea de optimizar la función aptitud (ISE), sin restricción alguna, a continuación, se muestra los ISE respectivos.

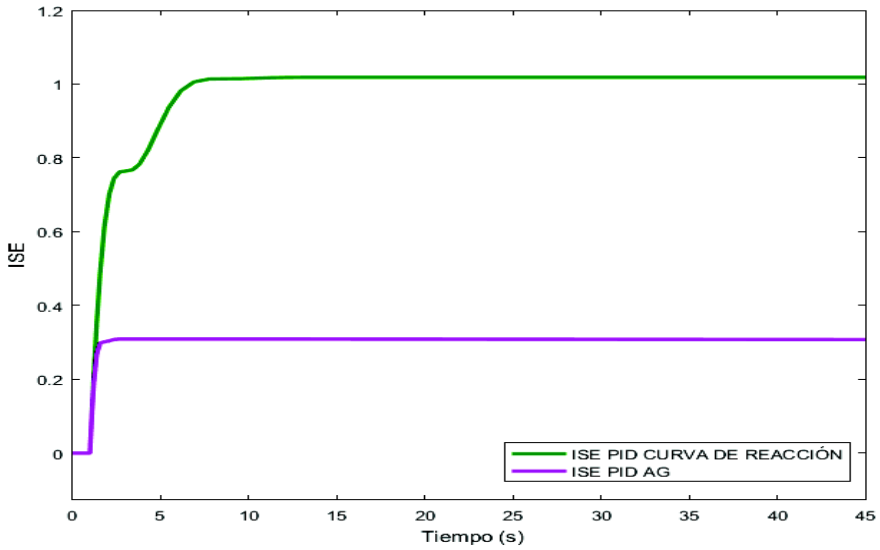


Figura 2.9. Gráfica Comparativa de los índices de desempeño generados.

En la Figura 2.9, se observa que el ISE ha sido optimizado de manera considerable al utilizar los Algoritmos Genéticos como herramienta de Sintonización, la respuesta obtenida por el programa de AG se muestra en la Figura 2.10.

```
y =

    0.3081

Optimization terminated: maximum number of generations exceeded.

x =

   79.9171    3.2777   19.8579

Elapsed time is 1461.037491 seconds.
```

Figura 2.10. Valores para Kp, Ki, Kd del PID – AG

Donde y es el costo de la función objetivo a optimizar (valor del ISE), los valores de x son los parámetros de K_p, K_i, K_d, y finalmente se muestra el costo computacional del programa, que en minutos son alrededor de 24 minutos, el costo computacional de los AG es elevado y depende del sistema a optimizar, el número de variables a calcular, las generaciones con las que se trabaje y con el computador que se lleve a cabo el proceso. Para este caso el tiempo

obtenido es para una planta de primer orden con retardo, para tres variables K_p, K_i, K_d, de un PID, se trabajó con 30 generaciones, con un tamaño de población de 50 individuos, selección por Torneo, porcentaje de cruce de 0.8, el apareamiento se lo realizó por cruce de un punto y con porcentaje de mutación de 0.01, finalmente todo esto se desarrolló en un computador cuyo procesador es de 2.7 GHz, 8 GB de RAM, Core i5 y Sistema de 64- bit.

A continuación, se muestran los pasos empleados para la realización de este programa:

Definir Parámetros

Se definen parámetros como número de individuos, número de variables, rangos de valores para la búsqueda del mejor individuo de cada variable, definir la función objetivo, número de generaciones, porcentaje de cruce entre 0 y 1, elegir el método de selección, el método de cruce o apareamiento y porcentaje de mutación, en la Figura 2.11, se muestra este proceso en forma de diagrama de flujo.

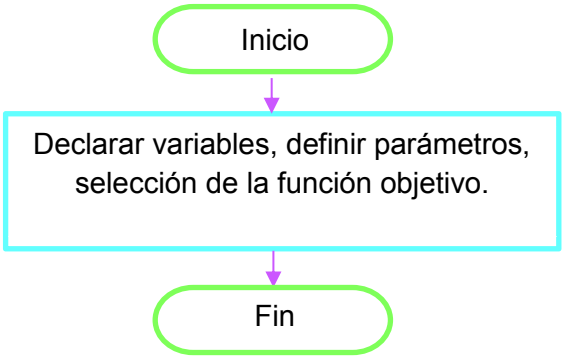


Figura 2.11. Diagrama de flujo para definir parámetros

Generar una población

Se crea una población de individuos aleatorios, por cada variable creada se generará una población, el diagrama de flujo para esta etapa se presenta en la Figura 2.12.

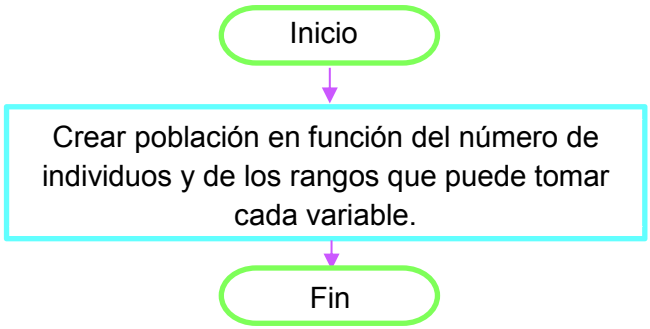


Figura 2.12 Diagrama de flujo para generar una población

Evaluación de la Función aptitud

Se evalúa los individuos en la función aptitud y se obtiene los valores de los costos, en esta etapa se da inicio al conteo de las generaciones previamente definidas, ver Figura 2.13.

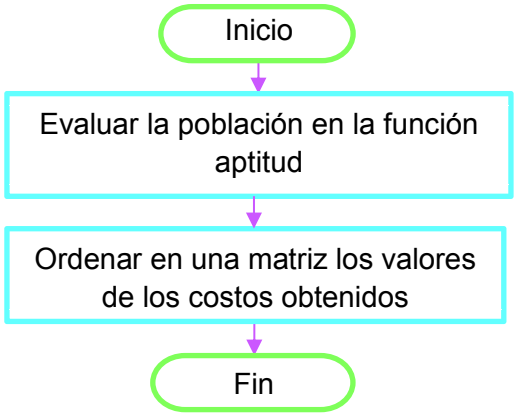


Figura 2.13 Diagrama de Flujo para Evaluar la Función Objetivo

Selección

Se selecciona los mejores individuos, que serán los que posteriormente se crucen para dar lugar a nuevos descendientes, se realiza la selección por torneo, este proceso se presenta en la Figura 2.14, en forma de diagrama de flujo.

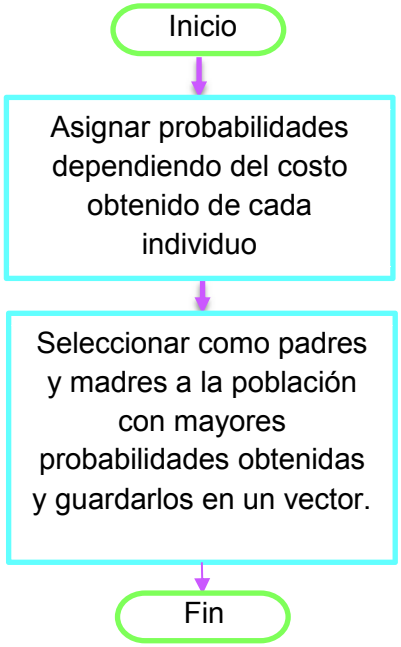


Figura 2.14. Diagrama de flujo para la selección de los individuos

Cruce o Apareamiento

En esta etapa se realiza una recombinación de los individuos seleccionados como padre y madre, se realiza el cruce basado en un punto, como se muestra en la Figura 2.15.

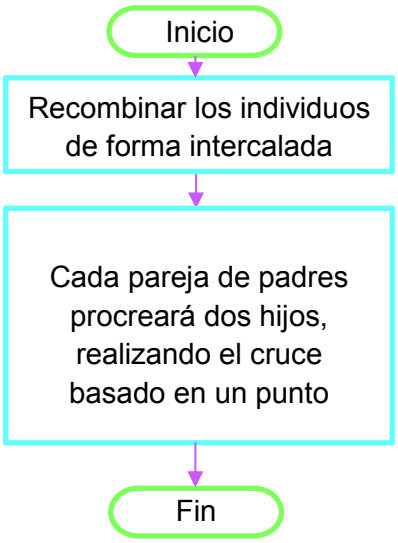


Figura 2.15. Diagrama de flujo para la etapa de cruce

Mutación

Se mutan los nuevos descendientes, comúnmente se elige un porcentaje de mutación bajo para que los individuos solo cambien un gen, ver Figura 2.16.

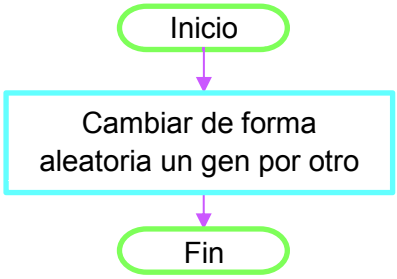


Figura 2.16 Diagrama de flujo para la mutación de los individuos

Para la aplicación de los AG en Matlab, se cuenta con la ayuda de la herramienta ***gaoptimset***, la cual permite trabajar con varias opciones para la selección de población, cruce y mutación, además permite elegir el criterio de parada por tolerancia, tiempo o número de generaciones, éste último criterio fue considerado para el desarrollo de este trabajo. Al utilizar la herramienta de Matlab no es necesaria la etapa de decodificación puesto que se trabaja directamente con poblaciones en números decimales. La ayuda proporcionada por Matlab se muestra en el ANEXO IV, y en [40].

Sintonización de controladores PID y SMC, para el control de altura y seguimiento de trayectorias de un Cuadricóptero mediante AG.

Se propone un esquema para la sintonización de controladores PID y SMC para el seguimiento de trayectorias, basado en algoritmos genéticos, utilizando como función objetivo el ISE del sistema, como se muestra en la Figura 2.18.

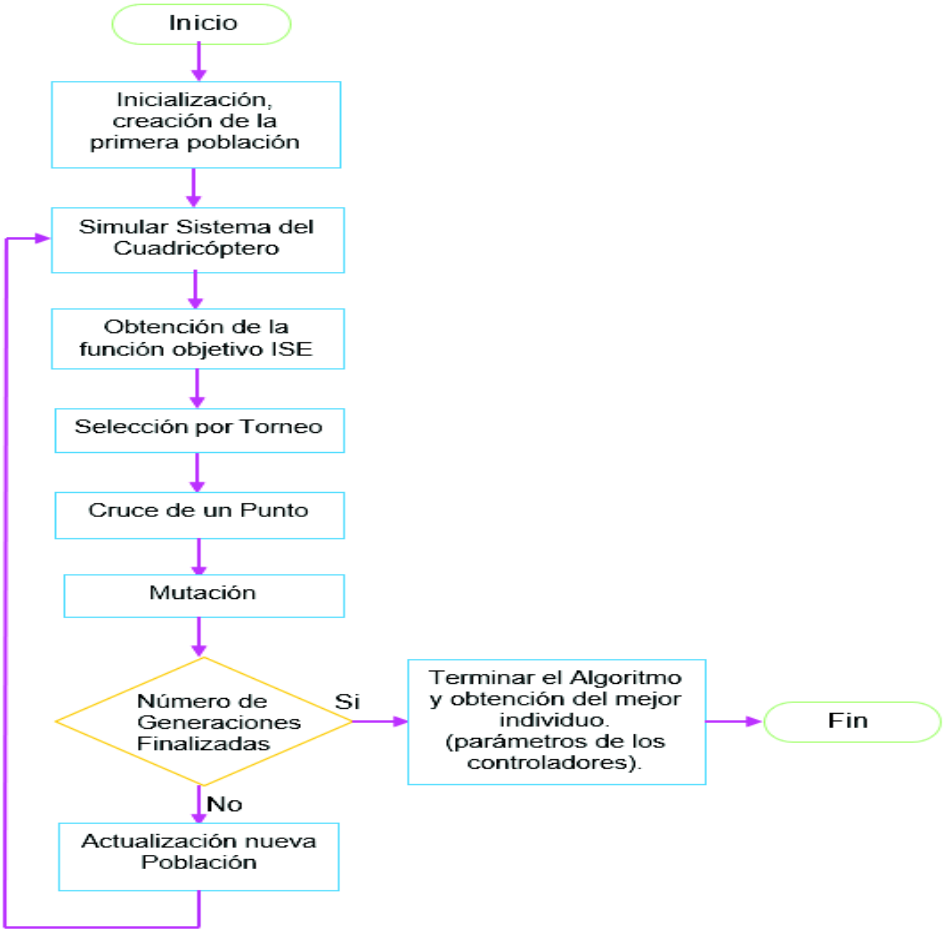


Figura 2.17. Diagrama de flujo basado en AG para la sintonización de controladores para un cuadricóptero.

Para el problema del seguimiento de trayectoria del cuadricóptero, se tienen los ISE de cada posición, (x, y, z) , en este caso, el ISE total correspondiente a la suma de los ISE en X y Y, constituirán la función objetivo o función aptitud, para el control de seguimiento de trayectoria, y el ISE en Z para el control de altura, en el diagrama de bloques de la Figura 2.18, se presenta un esquema total para la sintonización implementada en este proyecto.

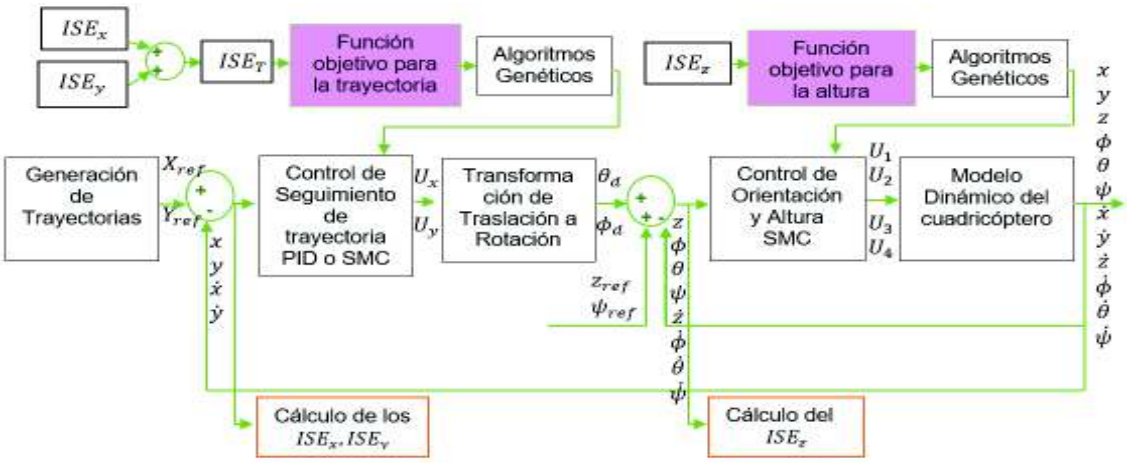


Figura 2.18. Sistema de sintonización para el control de altura y seguimiento de trayectoria

Al ser un sistema de control en cascada, dado que el control de altura se encuentra en el lazo interno, primero deberán ser sintonizados los parámetros para el control de altura que es un controlador SMC, y posteriormente se sintonizarán de forma simultánea los parámetros para los controladores de X y Y, que puede ser un PID o SMC.

Para el lazo interno se sintoniza mediante Algoritmos Genéticos los parámetros de λ_z y K_{D_z}, para el Controlador en Modo Deslizante para la altura, posteriormente son sintonizados K_p, K_i, K_d, para el Controlador PID, y finalmente se sintoniza $\lambda_x, \lambda_y, K_{D_x}, K_{D_y}$ para el SMC. En la Figura 2.19, se presenta el esquema de sintonización mediante AG para la altura del cuadricóptero con un control en modos deslizantes, el criterio de parada con el que se trabajó en este proyecto corresponde al número de generaciones deseadas.

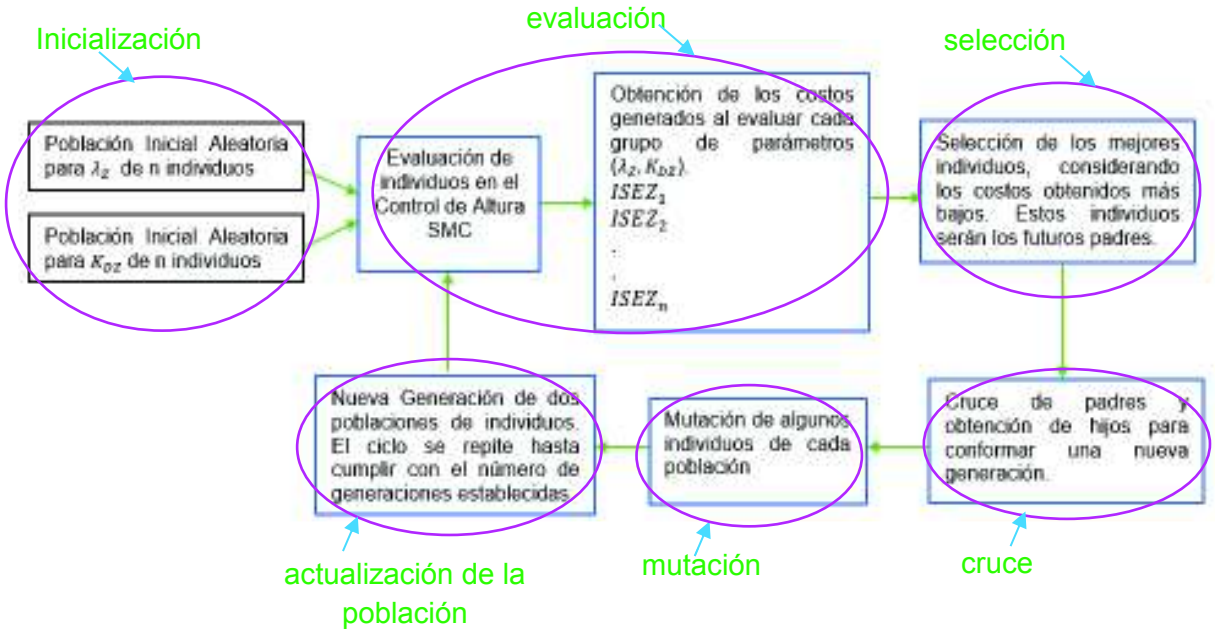


Figura 2.19 Sistema de sintonización implementado basado en Algoritmos Genéticos para el control de altura del cuadricóptero mediante SMC.

En las Figura 2.19, Figura 2.20 y Figura 2.21, se muestran los esquemas del proceso de sintonización implementado en este proyecto, como se puede observar, para el control de altura se trabajó con dos poblaciones para λ_z y K_{DZ}, respectivamente, para el control de seguimiento de trayectoria del cuadricóptero en la opción PID se trabajó con tres poblaciones para K_p, K_i, K_d, y finalmente para el seguimiento de trayectoria mediante SMC se consideraron cuatro poblaciones para $\lambda_x, \lambda_y, K_{Dx}, K_{Dy}$. Cabe mencionar que los parámetros K_p, K_i, K_d, son los mismos para el control en X y Y , a diferencia del SMC que se sintonizan parámetros para el control en X(λ_x,K_{Dx}) y parámetros para el control en Y (λ_y,K_{Dy}).

En la Figura 2.20 y la Figura 2.21, se presentan los sistemas implementados para la sintonización de controladores PID y SMC para el seguimiento de trayectoria del cuadricóptero.

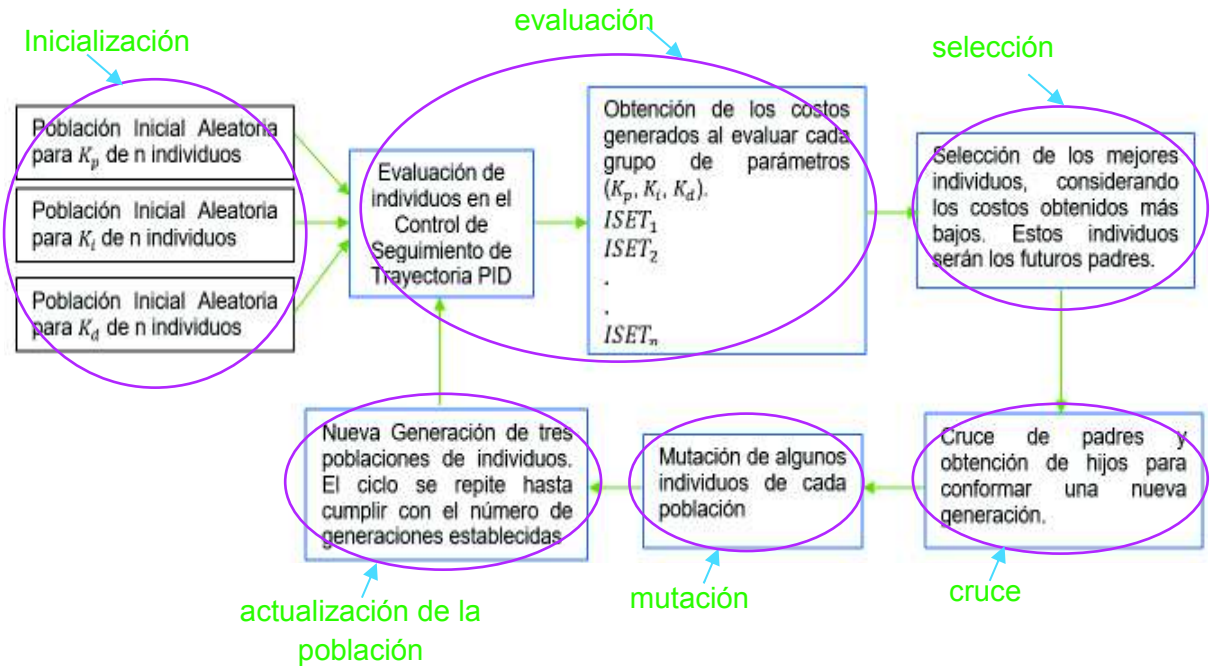


Figura 2.20 Sistema de sintonización implementado, basado en Algoritmos Genéticos para el control de seguimiento de trayectoria mediante PID.

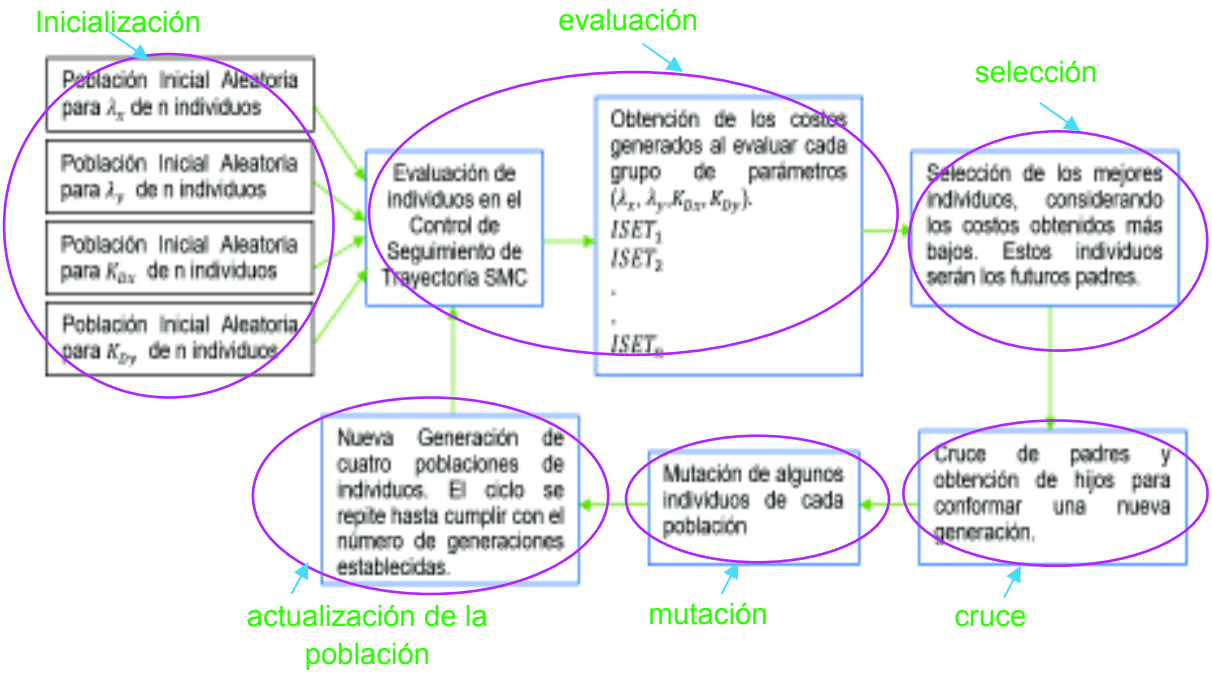


Figura 2.21 Sistema de sintonización implementado, basado en Algoritmos Genéticos para el control de seguimiento de trayectoria mediante SMC.

Costo Computacional de los Algoritmos Genéticos

El costo computacional de los AG depende del número de individuos con los que se trabaje, la cantidad de variables a optimizar, el número de generaciones y el tiempo de simulación del sistema al cual se van a aplicar los algoritmos genéticos. En el presente proyecto para todos los casos de estudio, se trabajó con una población de 50 individuos generados de forma aleatoria entre los rangos dados para cada variable y el criterio de parada considerado fueron 30 generaciones. En este proyecto se trabajó con un porcentaje de cruce de 0.8, lo que significa que el 80% de la población va a generar nuevos descendientes y el 20% de la población que presenta mejores características pasa directamente a la nueva población. Sabiendo que el algoritmo genético evalúa a todos los miembros de la población, se tiene como resultado 1210 evaluaciones del sistema aproximadamente. Dado que el programa desarrollado en el presente proyecto hace uso de Matlab - Simulink, el costo computacional fue calculado con los comandos TIC y TOC.

Tabla 2.1. Costo Computacional de los AG para el control de un cuadricóptero

Casos de Estudio		**Tiempo de Simulación**	**Numero de variables a optimizar**		**Costo Computacional (horas)**	
			PID	**SMC**	**PID**	**SMC**
Control de Altura	Control de Altura	10 (s)	---	2	---	10.5 h
Seguimiento de Trayectoria	Curva de Lissajous	16 (s)	3	4	22.8h	14.3h
	Senoidal	16 (s)	3	4	24.5 h	14.4 h
	Cuadrada	32 (s)	3	4	26.3h	18.8 h

Como se observa en la Tabla 2.1, los costos más elevados corresponden al seguimiento de trayectorias efectuadas por el controlador PID, esto se debe a que en la simulación el controlador PID presentaba un comportamiento bastante lento a comparación del SMC, aun cuando el número de variables a sintonizar en el PID es menor a las variables sintonizadas por el controlador SMC. Para ambos controladores la trayectoria con mayor costo computacional es la cuadrada debido a que el tiempo de simulación para completar la trayectoria es mayor a las demás trayectorias sintonizadas. El tiempo de simulación de cada caso de estudio corresponde al tiempo seteado para cada simulación, y no es el mismo que el tiempo que se demora en ejecutar dicha simulación. Estas pruebas fueron realizadas en un computador con 8 GB de RAM, procesador de 2.7 GHz, Core i5 y Sistema de 64- bit.

Herramienta Computacional para la sintonización de Controladores PID y SMC para el seguimiento de trayectoria de un cuadricóptero.

En el presente trabajo se elaboró una herramienta computacional que permite una manipulación sencilla de los parámetros de los Algoritmos Genéticos y del cuadricóptero, en este estudio técnico se trabaja con los datos del drone Draganflyer, mostrados en la Tabla 3.1, sin embargo, esta herramienta permite al usuario sintonizar los parámetros de los controladores PID y SMC para un cuadricóptero con diferentes características, como se observa en la Figura 2.22.

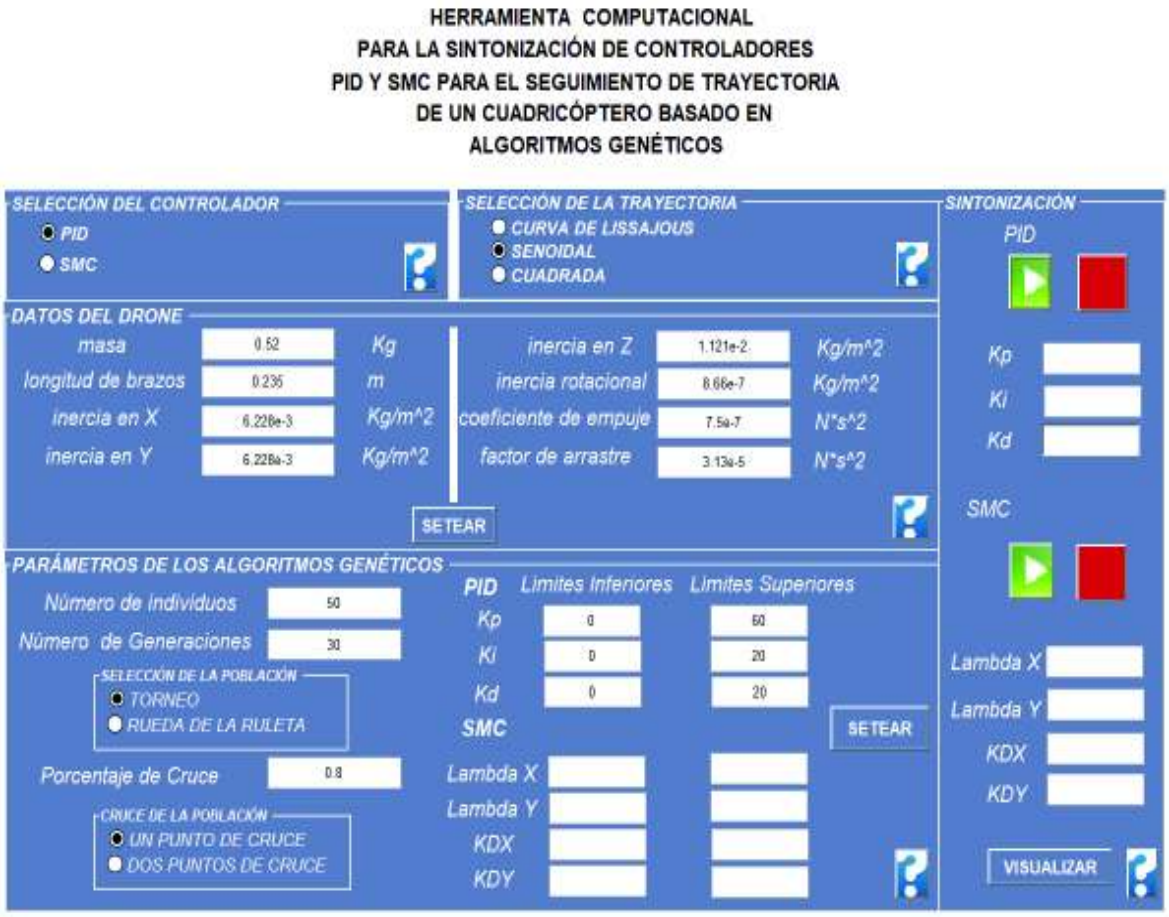


Figura 2.22. Herramienta computacional para la sintonización de controladores PID y SMC para el seguimiento de trayectorias de un cuadricóptero.

En cada sección, se muestran cuadros de ayuda que facilitan al usuario la utilización de esta herramienta. En esta interfaz se ha optado por considerar solo dos opciones para la selección de los individuos, debido a que la selección por torneo y la selección por rueda de la ruleta son las más adecuadas para problemas con tamaños de poblaciones considerables, sin embargo, para la sintonización de todos los controladores y trayectorias del presente

proyecto, se empleó la selección por torneo, dado que es la que mejor simula el comportamiento evolutivo de las especies. En cuanto al cruce de la población de igual forma se consideró dos opciones para que el usuario puede acceder a ellas, sin embargo, en el presente se trabajó con el cruce de un solo punto. Los límites inferiores y superiores establecen el rango en el cual se generarán los individuos de cada población, en la Figura 2.22, se muestra que para K_p se ha delimitado un rango entre 0 y 60, es decir, que los individuos que pertenecen a la población generada para K_p pueden tomar valores entre 0 y 60, mientras que, para K_i el rango dado es entre 0 y 20, indicando que sus individuos tomarán valores entre 0 y 20, al igual que la población de K_d. Estos rangos fueron determinados luego de sintonizar los controladores de forma heurística utilizando el ISE obtenido. Finalmente, la sección de Sintonización de la Figura 2.22, cuenta con la opción visualizar, la cual nos muestra el mejor individuo (parámetros), obtenidos por el Algoritmo Genético para la sintonización del controlador escogido. En la Figura 2.23, se presenta un diagrama de flujo, el cual permitirá una mejor comprensión al usuario para el uso de la herramienta.

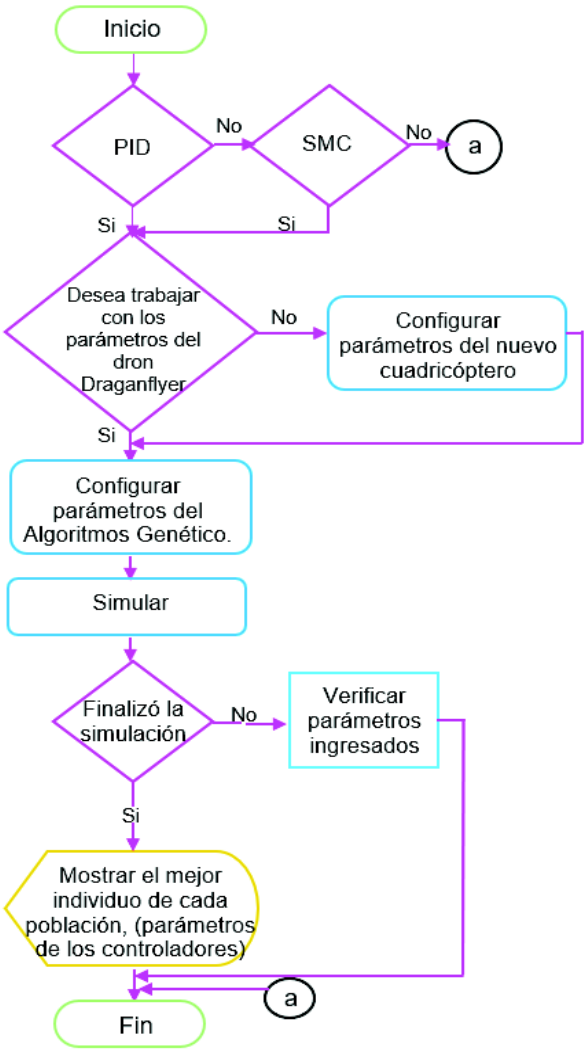
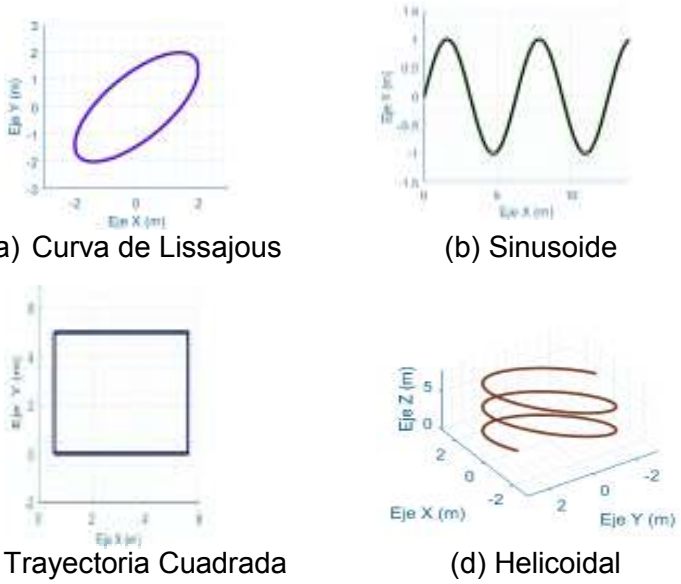


Figura 2.23 Diagrama de flujo de la herramienta computacional para sintonización de controladores PID y SMC.

2.4 Generación de Trayectorias

En el presente proyecto se plantea el seguimiento de cuatro trayectorias, 3 de ellas fijas en el plano Z a una altura de 2 metros y una variante en Z, como se muestra en la Figura 2.24.



(a) Curva de Lissajous (b) Sinusoide

(c) Trayectoria Cuadrada (d) Helicoidal

Figura 2.24. Trayectorias Generadas

En la Tabla 2.2, se muestran las ecuaciones que describen las trayectorias en (x, y, z), la trayectoria en forma de Curva de Lissajous y Sinusoide tienen un tiempo de simulación de 16 (s), la trayectoria cuadrada con 8 (m) por lado se ejecuta en 32 (s) y la Helicoidal se forma en base a una Curva de Lissajous con un tiempo de simulación de 70 (s).

Tabla 2.2. Configuración de las Trayectorias

TRAYECTORIA	**EJE X**	**EJE Y**	**EJE Z**
Curva de Lissajous	$x = 2\sin(0.4t)$	$y = 2\sin(0.4t + 0.8)$	$cte = 2$
Sinusoide	$x = t$	$\mathrm{y} = \sin(t)$	$cte = 2$
Cuadrada	$0.8t, \quad 0 \le t \le 8$ $2.8t, \quad 8 \le t \le 16$ $-0.8t + 7, \quad 16 \le t \le 24$ $0, \quad 24 \le t \le 32$	$0, \quad 0 \le t \le 8$ $0.8t - 2.8, \quad 8 \le t \le 16$ $7, \quad 16 \le t \le 24$ $-0.8t + 17, \quad 24 \le t \le 32$	$cte = 2$
Helicoidal	$x = 2\sin(0.4t)$	$y = 2\sin(0.4t + 0.8)$	$z = 0.2t$

2.5 Interfaz Gráfica de Usuario

La interfaz gráfica se crea con el fin de facilitar al usuario el análisis de las gráficas obtenidas, así como introducir de forma sencilla: parámetros de sintonización, escoger la trayectoria deseada, posibles cambios de referencia y elegir trabajar con perturbaciones. Además, cuenta con un panel de visualización de índices de desempeño (ISE, IAE), de cada una de las posiciones para su comparación y análisis respectivo, también cuenta con visualización de formas de onda de posiciones y errores obtenidos por los diferentes controladores y sintonizaciones. Las simulaciones son realizadas en Matlab - Simulink, por lo que resulta conveniente trabajar en el desarrollo de la Interfaz Gráfica de Usuario con una herramienta de este programa como GUIDE (Graphical User Interface Development). En la Figura 2.25, se presenta el esquema de Simulación de este proyecto.

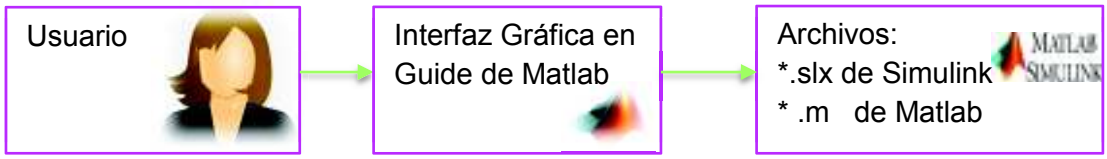


Figura 2.25. Esquema de simulación del sistema

Para comprender a detalle el proceso del esquema presentado, se desarrollan los diagramas de flujos correspondientes a cada etapa.

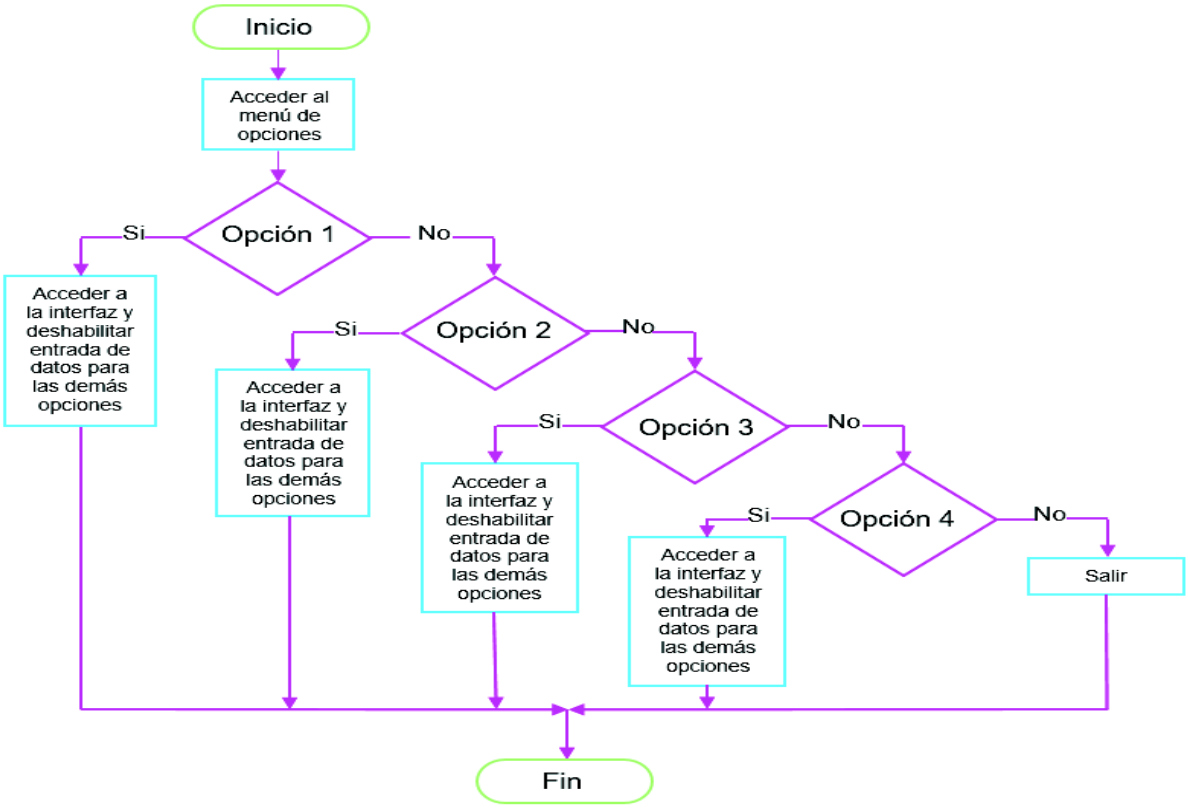


Figura 2.26. Diagrama de Flujo del Menú de Opciones de la Interfaz Gráfica de Usuario

En la Figura 2.26, se ilustra el diagrama de flujo del menú de opciones donde la opción 1 corresponde a la altura y orientación del cuadricóptero, la opción 2 presenta el seguimiento de trayectoria mediante los controladores PID y SMC, la opción 3 indica el seguimiento de trayectoria mediante PID y PID – AG y finalmente la opción 4 está asignada al seguimiento de trayectoria mediante SMC y SMC – AG.

En la Figura 2.27, se ilustra el diagrama de flujo para la ventana de orientación y altura del cuadricóptero, el primer paso es cargar el archivo altura.slx, el cual contiene la simulación del cuadricóptero con el control de orientación y altura, luego se cargan los parámetros como referencias de altura, habilitar o deshabilitar perturbaciones, finalmente, luego de que la simulación termine exitosamente, se visualizan las gráficas y resultados obtenidos.

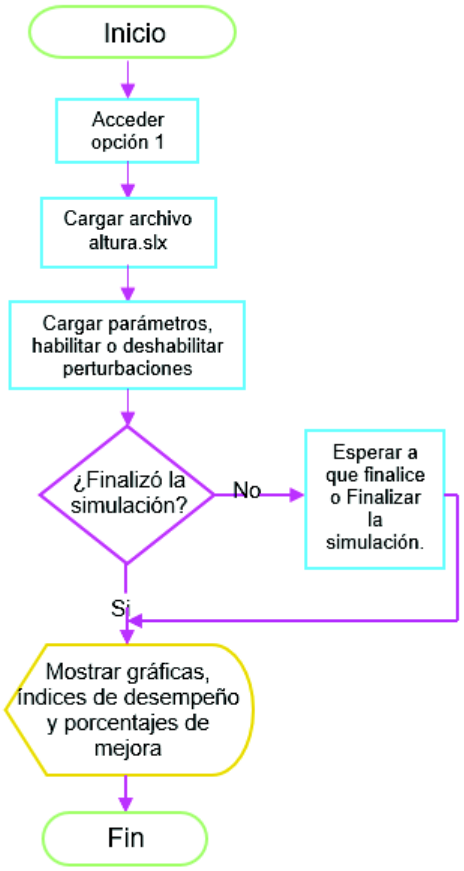


Figura 2.27. Diagrama de flujo para la opción 1 de la interfaz gráfica de usuario

En la Figura 2.28, se muestra el diagrama de flujo para el seguimiento de trayectoria mediante PID y SMC, de forma parecida que en la opción anterior se carga el archivo pidysmc.slx y los parámetros requeridos, la diferencia es que en esta ventana se tienen cuatro opciones de trayectorias y de acuerdo a la trayectoria escogida se setean los parámetros de sintonización de la misma, se pueden habilitar o deshabilitar perturbaciones, finalmente, luego de que la simulación haya finalizado, se muestran las gráficas y los resultados obtenidos, esta

metodología se sigue para las ventanas restantes, cargando los archivos pidypidag.slx y smcysmcag.slx, mostrando los diagramas de flujo en las Figura 2.29 y Figura 2.30 respectivamente.

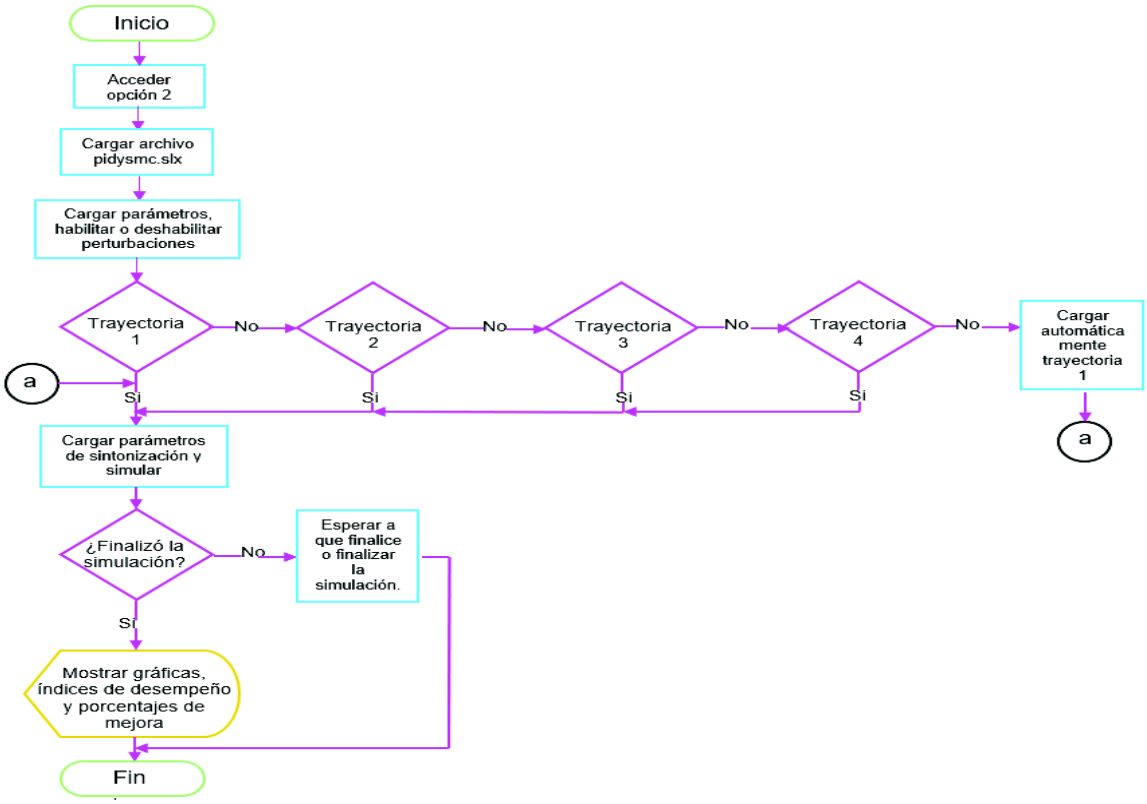


Figura 2.28. Diagrama de flujo para la opción 2 de la interfaz gráfica de usuario

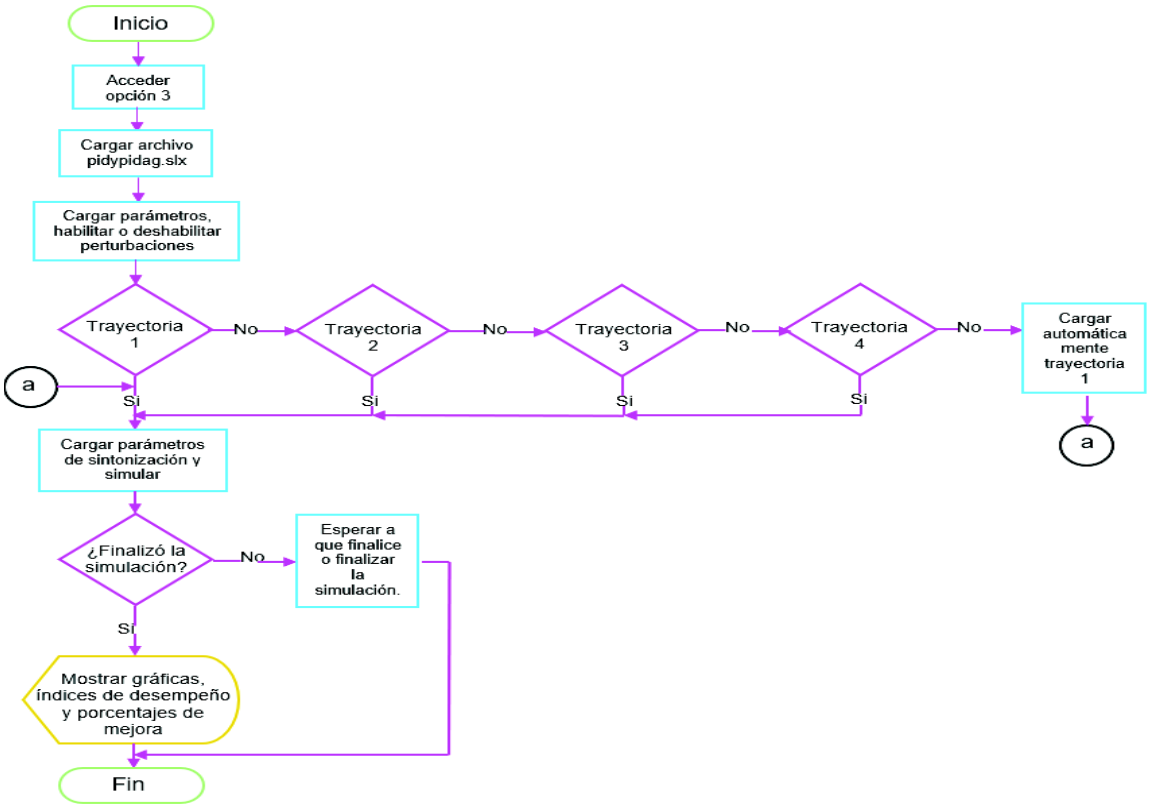


Figura 2.29. Diagrama de flujo para la opción 3 de la interfaz gráfica de usuario

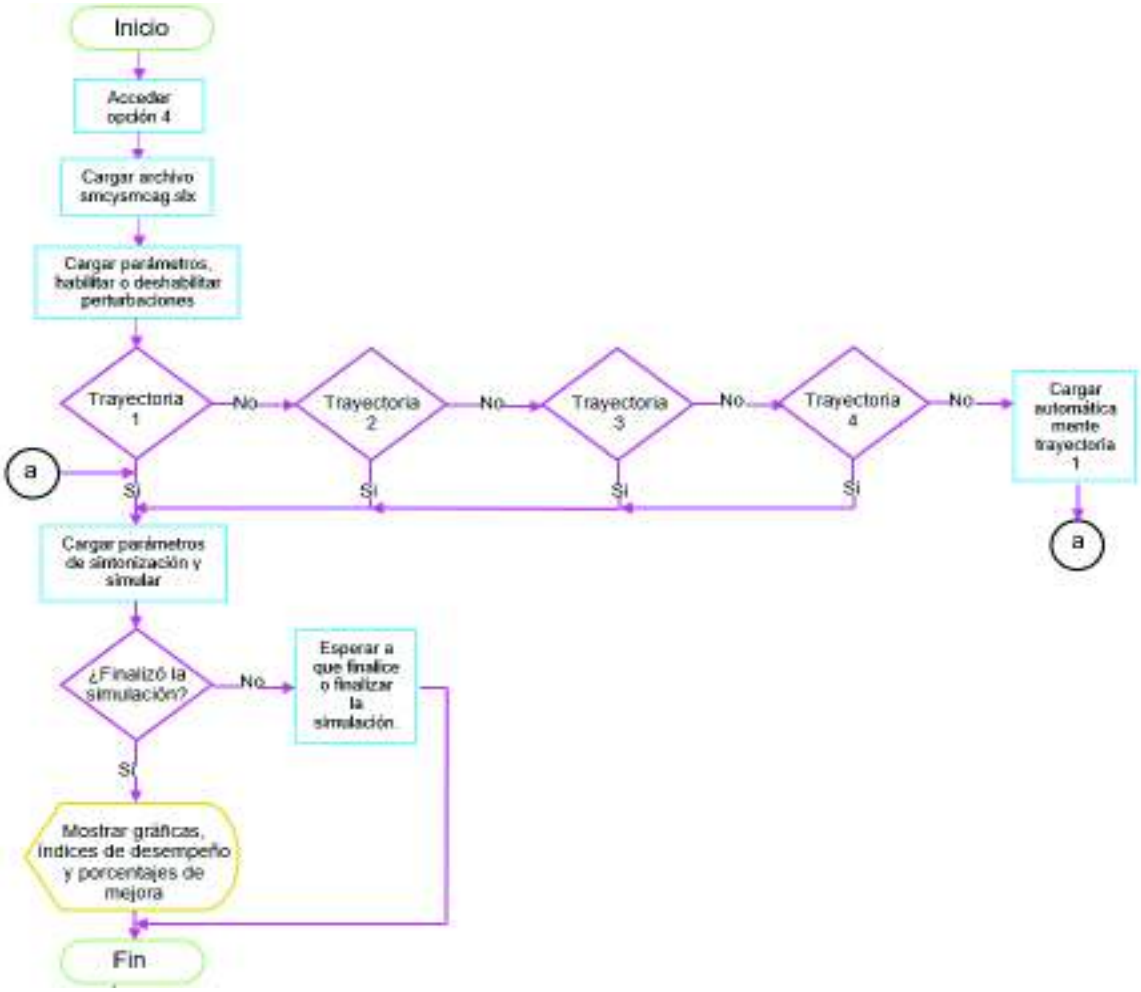


Figura 2.30. Diagrama de Flujo para la opción 4 de la interfaz gráfica de usuario

En el Anexo I, se adjunta el manual de usuario de la interfaz gráfica con la información necesaria para su funcionamiento.

3. RESULTADOS Y DISCUSIÓN

En esta sección se presentan los resultados de las simulaciones realizadas para el seguimiento de trayectoria de un cuadricóptero mediante PID y SMC, y posteriormente las sintonizaciones de dichos controladores mediante Algoritmos Genéticos.

Para comparar el desempeño de los controladores implementados en el seguimiento de trayectoria se ha utilizado los valores de los índices de desempeño: ISE (Integral del Error Cuadrático) e IAE (Integral del Error Absoluto).

$$IAE = \int_0^\infty |e(t)|dt$$

Ecuación 3.1. Integral del error absoluto

$$ISE = \int_0^\infty e(t)^2 dt$$

Ecuación 3.2. Integral del error cuadrático

Para cada trayectoria se presenta el análisis de los índices de desempeño indicados, así como las gráficas de las trayectorias en X, Y y Z, errores obtenidos y porcentajes de mejora de los controladores PID y SMC al ser sintonizados de forma heurística y mediante Algoritmos Genéticos.

3.1 Parámetros del Cuadricóptero

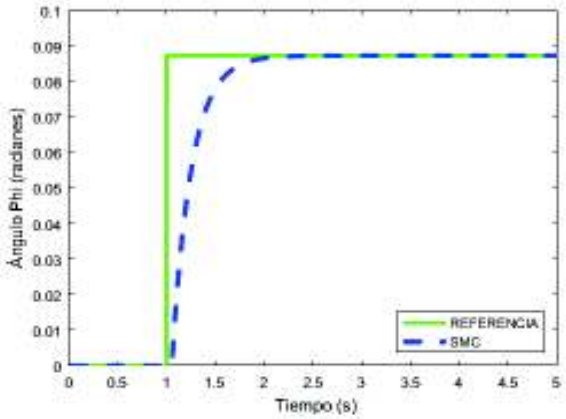
En el presente proyecto se trabaja con los parámetros del Drone Draganflyer, sin despreciar los efectos giroscópicos, dichos parámetros son detallados en la Tabla 3.1

Tabla 3.1. Parámetros del Drone Draganflyer [6],[36]

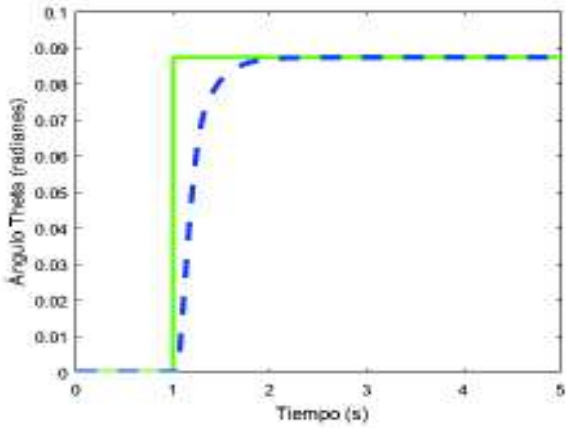
Parámetro	Descripción	Valor	Unidad
m	Masa del drone	0.52	$[\mathrm{Kg}]$
l	Longitud del brazo del drone	0.235	$[\mathrm{m}]$
g	Gravedad	9.8	$[\mathrm{m}/\mathrm{seg}^2]$
I_x	Inercia en X	6.228e-3	$[\mathrm{kg}/\mathrm{m}^2]$
I_y	Inercia en Y	6.228e-3	$[\mathrm{kg}/\mathrm{m}^2]$
I_z	Inercia en Z	1.121e-2	$[\mathrm{kg}/\mathrm{m}^2]$
J_r	Inercia del rotor (motor + reductora +hélice)	8.66e7	$[\mathrm{kg}/\mathrm{m}^2]$
b	Factor de arrastre	3.13e-5	$[\mathrm{N}*\mathrm{m}/\mathrm{seg}^2]$
d	Coeficiente de empuje	7.5e-7	$[\mathrm{N}*\mathrm{m}/\mathrm{seg}^2]$

3.2 Control de Orientación mediante SMC

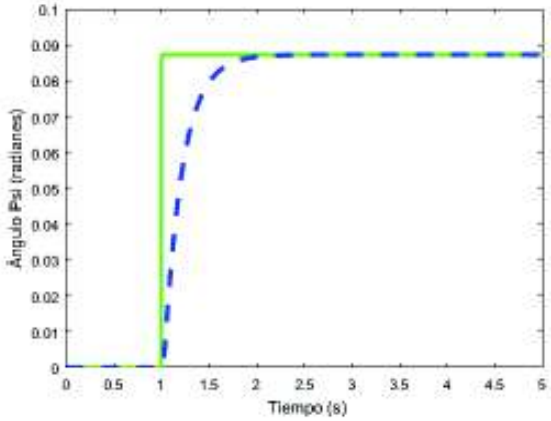
El control de orientación para el cuadricóptero fue realizado mediante un controlador en Modo Deslizante. En la Figura 3.1, se observa una entrada de $\pi/36$ equivalente a 0.087 radianes o 5 grados, la calibración fue realizada de forma heurística observando el comportamiento de la salida de la planta. Finalmente, los valores de las ganancias del controlador SMC para la orientación son mostrados en la Tabla 3.2.



(a) Control de Roll (ϕ)



(b) Control de Pitch (θ)



(c) Control de Yaw (ψ)

Figura 3.1. Control de Orientación mediante SMC

Tabla 3.2. Parámetros para el Control de Orientación

Controlador	**SMC**		
Variables	$\boldsymbol{\lambda}$	$\boldsymbol{K_D}$	$\boldsymbol{\delta}$
ϕ	5	12	0.3
θ	5	12	0.3
ψ	5	12	0.3

En la Figura 3.2 ,se presentan las acciones de control generadas por el cuadricóptero para las leyes de U_2, U_3, U_4, correspondientes al control de orientación para roll, pitch y yaw, respectivamente. En las gráficas mostradas se observa que al 1 (s), el UAV realiza un esfuerzo de control de 20 Newtons, esto se debe a que en ese instante se inicia la referencia de la señal paso y luego se estabiliza alrededor de 0 Newtons, dado que la referencia de ángulos fue asignada a 5 grados. Los esfuerzos de control presentados son obtenidos mediante la implementación del controlador SMC con los valores de sintonización presentados en la Tabla 3.2.

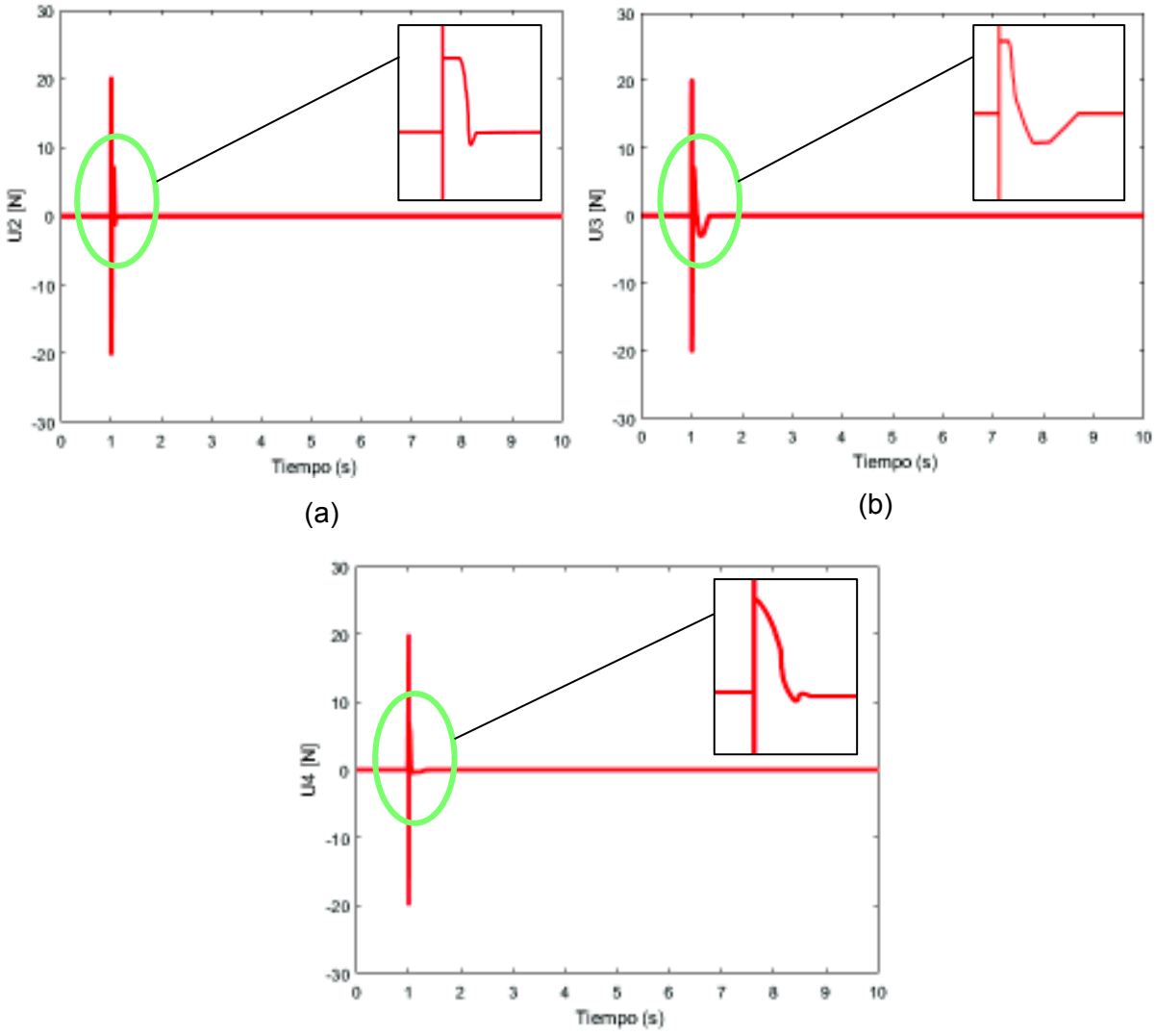


Figura 3.2. Señales de control para la orientación del cuadricóptero

3.3 Control de Altura para un Cuadricóptero SMC Y SMC- AG

El control de altura para el cuadricóptero fue realizado mediante SMC, sintonizado heurísticamente y más adelante con Algoritmos Genéticos, a través de los AG se obtuvieron los parámetros de λ y K_D, mientras que delta se mantuvo fijo en su sintonización inicial, en la Figura 3.3, se presenta el proceso de sintonización mediante AG para la altura del cuadricóptero.

Figura 3.3. Sintonización para el control de altura mediante algoritmos genéticos

En la Figura 3.4, se observa que existe una mejora en las respuestas de altura del sistema en cuanto a tiempo de establecimiento, la referencia de altura fue dada en 2 metros, debido a que la aplicación del drone será para interiores.

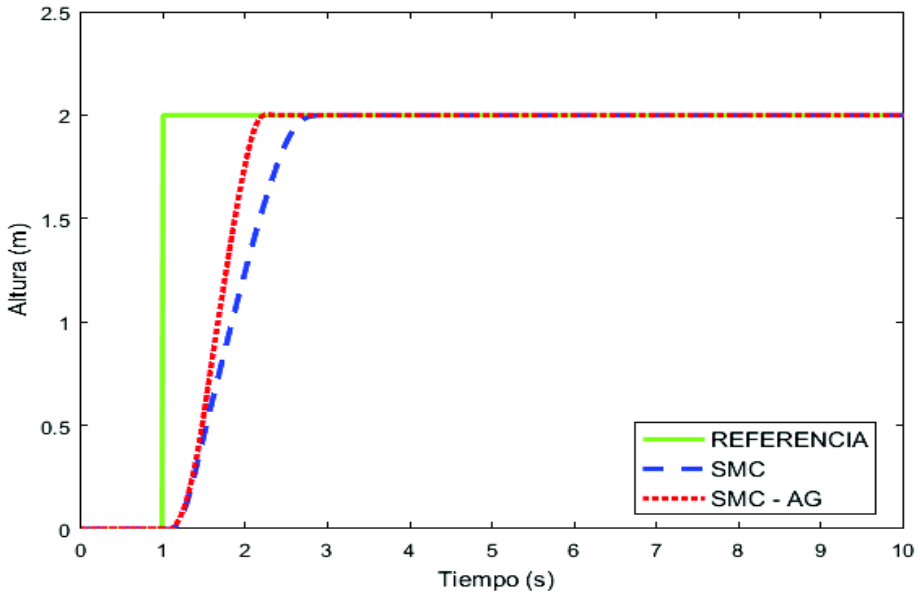
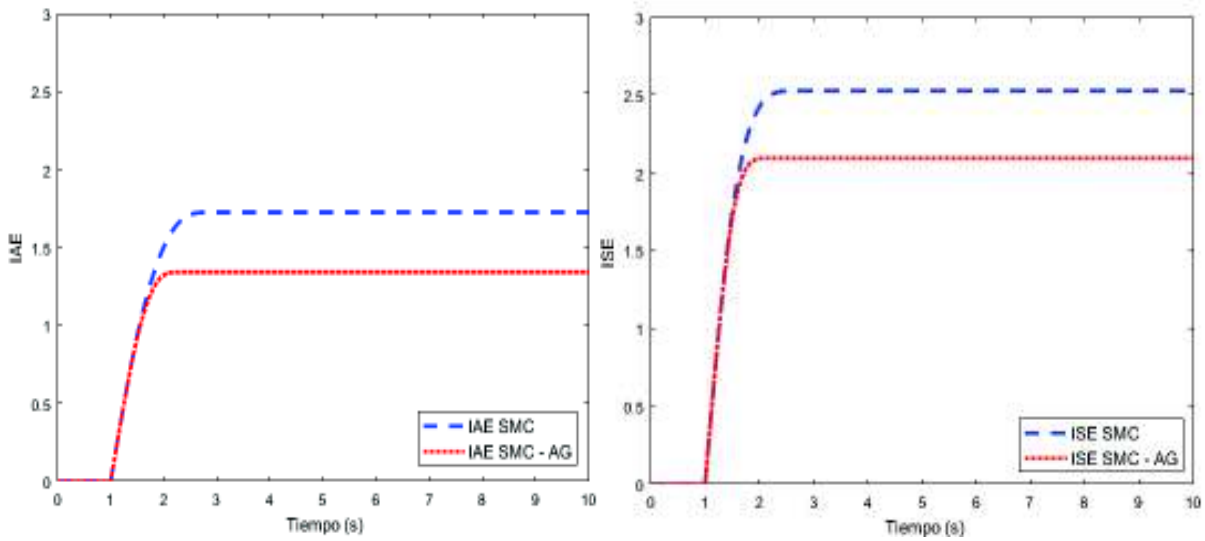


Figura 3.4. Control de Altura mediante SMC Y SMC – AG

En la Figura 3.5, se muestran los índices de desempeño de altura del sistema (IAE, ISE), se observa que el Índice de la Integral del Error Absoluto (IAE), es más pequeño que el Índice de la Integral del Error Cuadrático (ISE), esto se debe a que el ISE evalúa con mayor peso los errores grandes que ocurren al inicio de la simulación. Además, se observa una mejoría en ambos índices de desempeño luego de ser sintonizados mediante AG.



(a) IAE (b) ISE

Figura 3.5. Índices de Desempeño de la Altura mediante SMC y SMC -AG

En la Tabla 3.3 se aprecia los valores de los parámetros de los controlador SMC y SMC -AG para la altura.

Tabla 3.3. Parámetros para el Control de Altura mediante SMC Y SMC -AG

Controlador	**SMC**			**SMC -AG**		
Variables	$\boldsymbol{\lambda}$	$\boldsymbol{K_D}$	$\boldsymbol{\delta}$	$\boldsymbol{\lambda}$	$\boldsymbol{K_D}$	$\boldsymbol{\delta}$
z	30	29	0.6	31.1932	51.6124	0.6

En la Figura 3.6, se presenta los valores para λ y K_D, obtenidos mediante Algoritmos Genéticos, luego de haber excedido el número de generaciones, en este caso se trabajó con 30 generaciones. En esta sintonización se trabajó con la selección por Torneo, cruce de un punto, porcentaje de cruce de 0.8, porcentaje de mutación de 0.01, y una población de 50 individuos, el valor de fval mostrado en la imagen corresponde al costo del ISE, dado que la función aptitud o función objetivo fue el ISE en Z, y los valores de x corresponden a los parámetros λ y K_D, para la altitud.

```
Optimization terminated: maximum number of generations exceeded.

x =

   33.1932   51.6124


fval =

    2.1253
```

Figura 3.6. Parámetros para el SMC obtenidos mediante Algoritmos Genéticos

En la Figura 3.7, se muestran las señales obtenidas por los controladores SMC y SMC-AG, para la altitud del cuadricóptero, al 1 (s) se observa que ambas señales se produce un pico de esfuerzo de control, esto se debe a que en ese instante inicia la referencia de altura, luego de estabilizarse el esfuerzo de control permanece constante alrededor de 5.1 Newtons, que es la fuerza necesaria para que el cuadricóptero se mantenga en el vuelo. La señal de acción de control obtenida por el SMC-AG se estabiliza primero que la obtenida por el SMC.

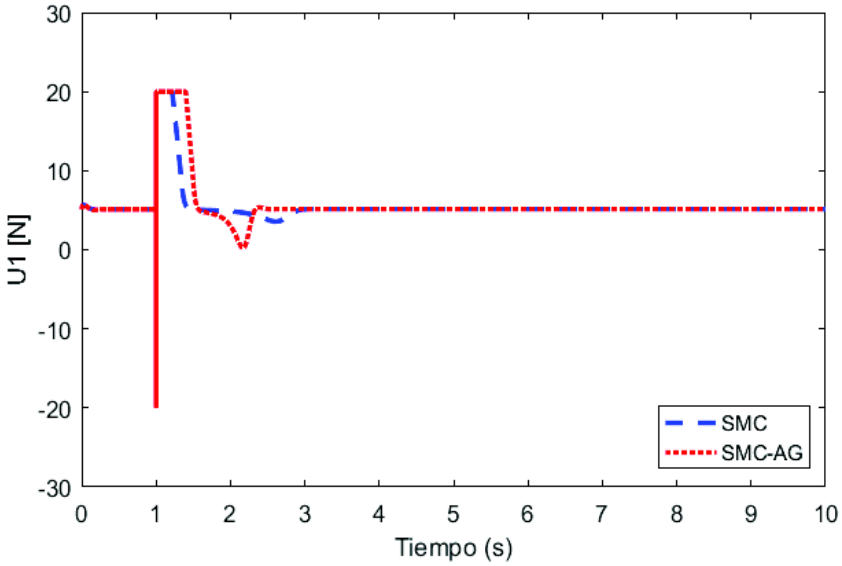


Figura 3.7. Salidas de Control para la Altitud

3.4 Control de Seguimiento de Trayectoria de un Cuadricóptero

Para el Seguimiento de Trayectoria de un Cuadricóptero, se implementaron dos algoritmos de control: PID y SMC, ambos sintonizados de forma heurística luego de observar el comportamiento de la planta en un proceso de prueba y error del ISE obtenido, y posteriormente sintonizados mediante Algoritmos Genéticos.

Para la sintonización con Algoritmos Genéticos se trabajó con 30 generaciones siendo suficientes para este problema puesto que a partir de un cierto número de iteraciones los costos de los individuos eran los mismos, un porcentaje de cruce de 0.8, porcentaje de mutación de 0.01, una población de 50 individuos, selección por torneo, cruce de un punto, los límites superiores e inferiores para establecer el conjunto poblacional fueron determinados de acuerdo a la sintonización previa de cada trayectoria y como función objetivo se utilizó la sumatoria de los índices de desempeño ISE, correspondiente a las posiciones en X y Y.

En el presente trabajo se sintonizaron mediante AG los parámetros para los controladores PID y SMC para las trayectorias: Curva de Lissajous, Sinusoide y Cuadrada, mientras que la trayectoria Helicoidal tiene como base una Curva de Lissajous por lo que se utilizaron los mismos parámetros que para la trayectoria mencionada.

Para la sintonización realizada de forma heurística de los controladores PID y SMC, se trabaja con el mismo controlador para X y Y, es decir los parámetros o ganancias de ambos controladores son las mismas tanto en la Ley de Control en X como en Y, lo mismo ocurre en la sintonización realizada mediante Algoritmos Genéticos en el Controlador PID. Se sintonizan tres variables (K_p, K_i, K_d), siendo utilizados en el control de X y de Y, como se ilustra en la Figura 3.8.

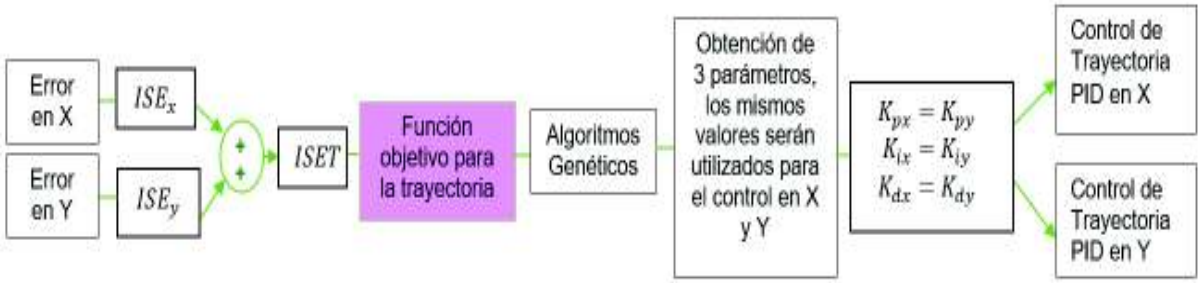


Figura 3.8. Sintonización para el controlador PID del seguimiento de trayectoria mediante algoritmos genéticos.

A diferencia de la sintonización mediante AG para PID, en el presente proyecto en el Controlador SMC se sintonizan cuatro variables (λ_x, K_{Dx}), para el Control en X y (λ_y, K_{Dy}), para el Control en Y como se ilustra en la Figura 3.9. Esto debido a que se deseaba probar la sintonización de los AG con cuatro variables.

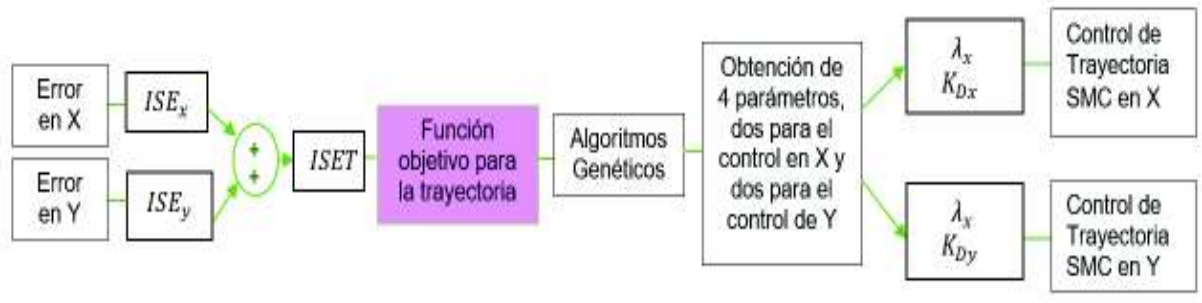


Figura 3.9. Sintonización para el controlador SMC del seguimiento de trayectoria mediante algoritmos genéticos.

Seguimiento de Trayectoria para una Curva de Lissajous.

La primera prueba realizada fue el seguimiento de una curva de Lissajous con un tiempo de simulación de 16 (s), se parten de los puntos iniciales $X_o = 0;\ Y_o = 0\ ;\ Z_0 = 2$.

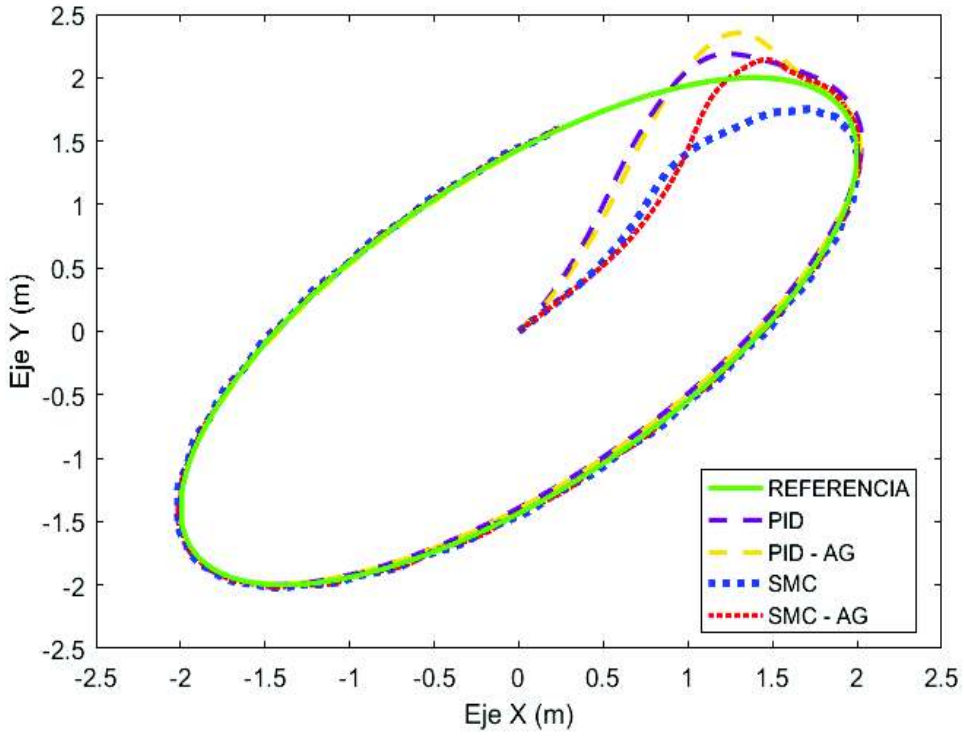


Figura 3.10. Seguimiento de Trayectoria de una Curva de Lissajous, mediante PID, PID – AG, SMC y SMC – AG.

En la Figura 3.10, se muestra el seguimiento de Trayectoria de una Curva de Lissajous, donde se aprecia que el controlador SMC presenta menor sobre impulso al inicio de la trayectoria tanto en su sintonización de forma heurística como mediante AG en comparación al PID, de igual forma se observa que los controladores PID-AG y SMC-AG presentan mayor sobre impulso, pero un error de posición y tiempo de establecimiento menor que los controladores sintonizados de forma heurística. Finalmente se puede apreciar que el seguimiento mediante PID y SMC sintonizados con AG alcanzan más rápido la trayectoria.

En la Figura 3.11, se muestran las trayectorias en X(m), obtenidas por los controladores PID y SMC y sus respectivas sintonizaciones, en (a) se observa que el controlador PID presenta un error de posición considerable respecto al SMC, posteriormente en (b) se visualiza como dicho error de posición del PID disminuye al ser sintonizado mediante Algoritmos Genéticos, en (c) no se observa diferencia a simple vista dado que la mejora en el ISE a lo largo de toda la trayectoria es de 33% como se muestra en la Tabla 3.4, sin embargo, el error de posición y tiempo de establecimiento han disminuido.

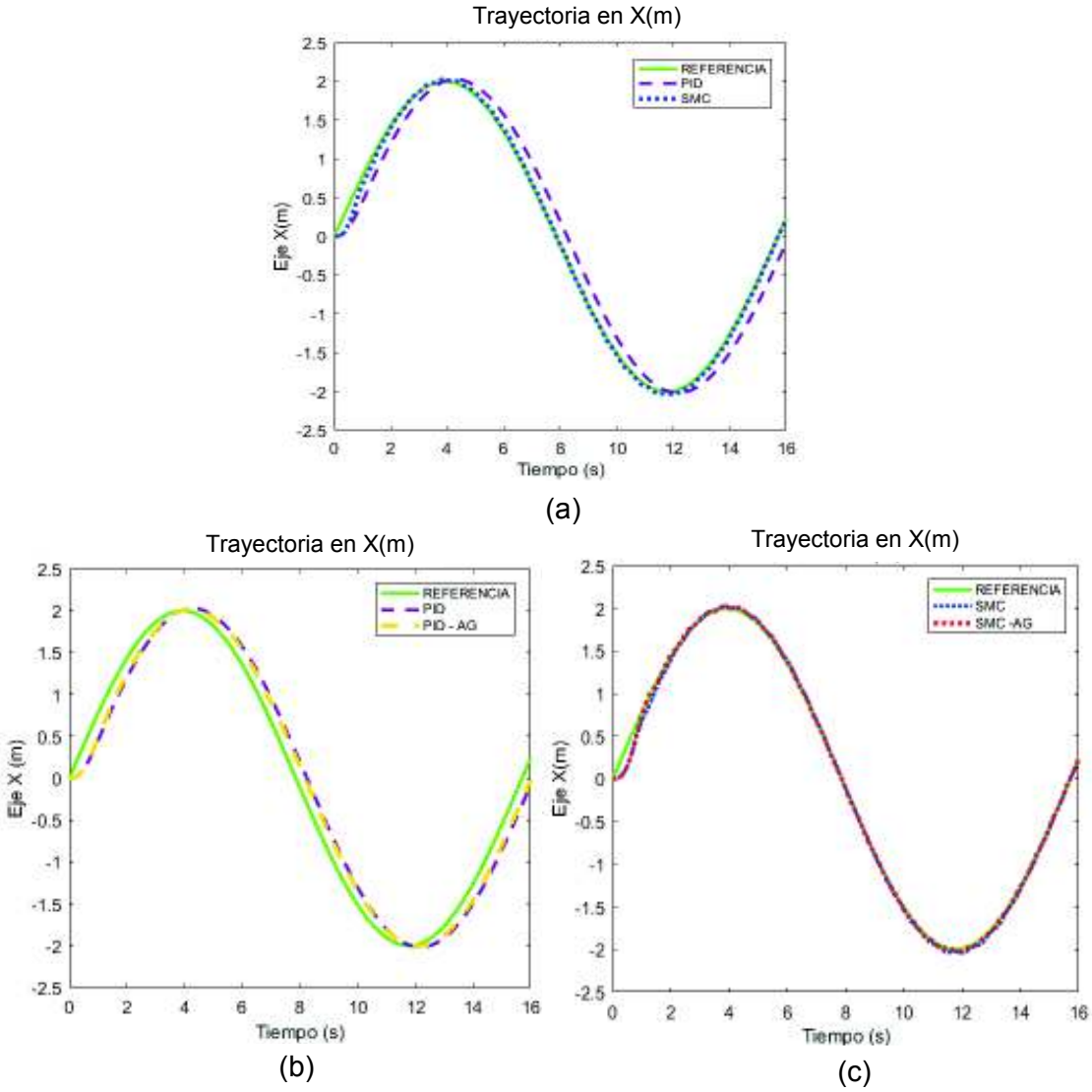


Figura 3.11. Trayectorias en X(m) respecto al Tiempo (s) de una Curva de Lissajous, (a) Trayectoria en X (m) Mediante PID y SMC, (b) Trayectoria en X (m) Mediante PID y PID - AG, (c) Trayectoria en X (m) Mediante SMC y SMC – AG

En la Figura 3.12, se muestran las gráficas de las trayectorias en Y(m), en (a) se observa que el PID presenta un sobre impulso al inicio de la trayectoria mientras que el controlador SMC no presenta sobre impulso y logra estabilizarse a los 4 (s), en (b) al sintonizar el PID mediante AG se observa que presenta mayor sobre impulso pero el error de posición disminuye, en (c) al sintonizar el SMC mediante AG presenta un pequeño sobre impulso pero el tiempo de establecimiento es menor.

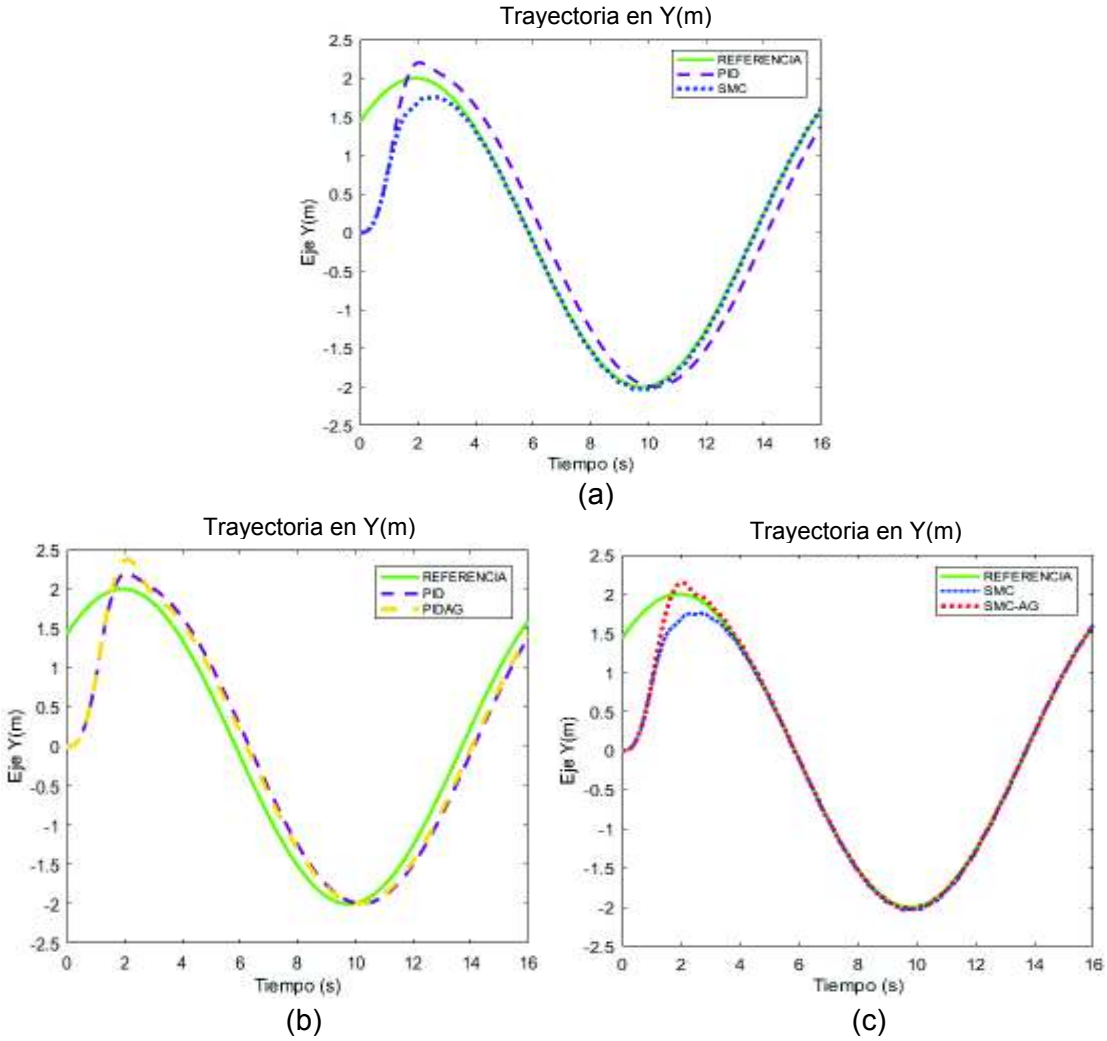


Figura 3.12. Trayectorias en Y(m) con respecto al Tiempo (s) para una curva de Lissajous, (a) Mediante PID y SMC, (b) Mediante PID y PID-AG, (c) mediante SMC y SMC-AG

En la Figura 3.13, se presentan las gráficas de los errores en X (m), correspondientes a los controladores PID y SMC con sus respectivas sintonizaciones. En (a) se observa que el SMC logra estabilizarse a los 3 (s) y permanece alrededor de 0, mientras que el PID presenta error de posición de 0.3 (m) equivalente a 30 cm, en (b) se observa que el PID al ser sintonizado mediante AG disminuye su error de posición a 25 cm aproximadamente y el sobre impulso también es menor y finalmente en (c) se visualiza que el sobre impulso luego de haber sintonizado el SMC mediante AG se mantiene pero el tiempo de establecimiento y error de posición es menor. En los errores obtenidos por el SMC, se observa un pequeño chattering que es menor a 0.05 m, lo que no se considera un problema para el desempeño del sistema.

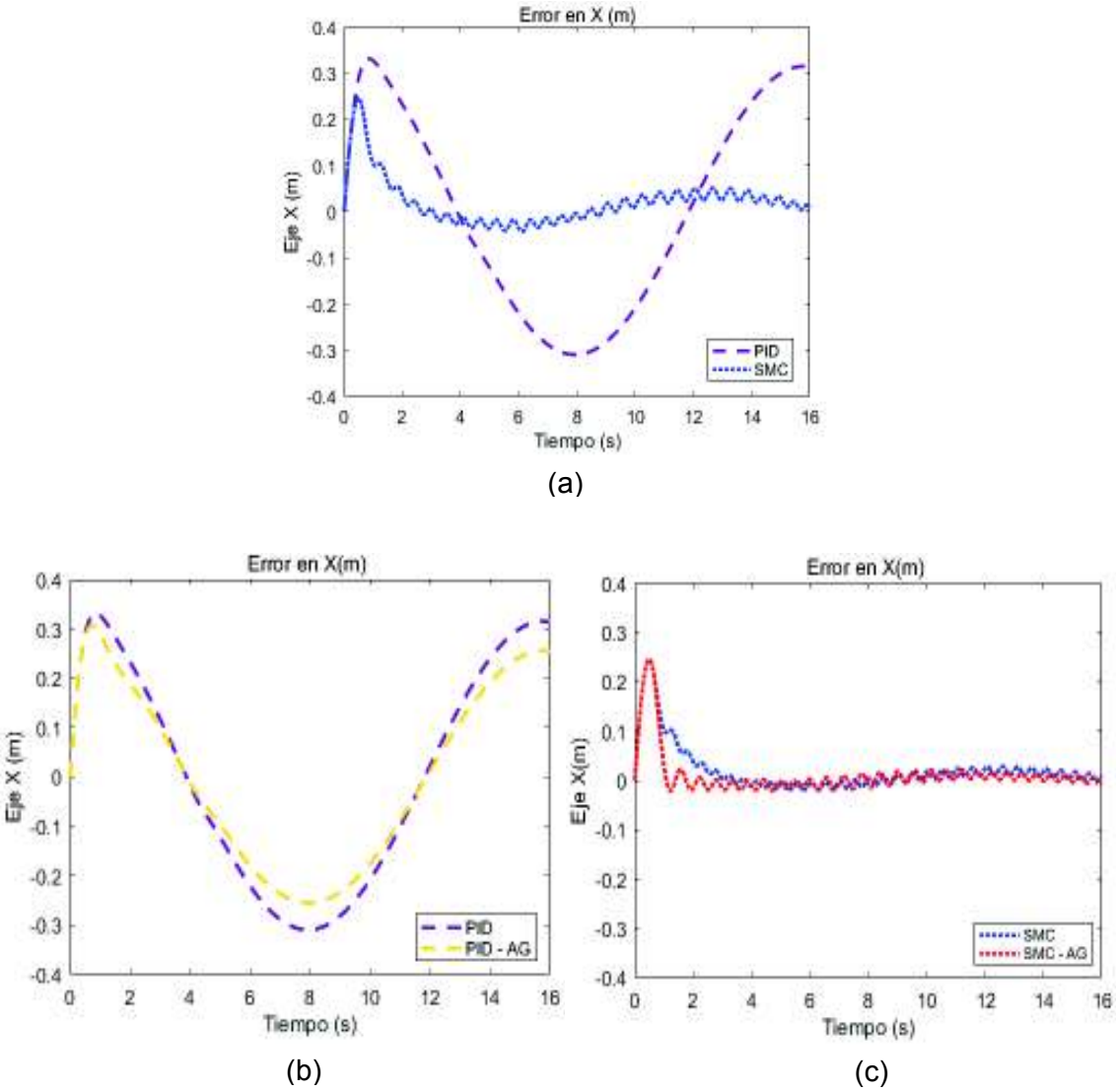


Figura 3.13. Errores en X (m) respecto al Tiempo (s) para una Curva de Lissajous, (a) Mediante PID Y SMC, (b) Mediante PID y PID - AG, (c) Mediante SMC y SMC – AG

En la Figura 3.14, se muestran los errores en Y (m) respecto al Tiempo (s), para una curva de Lissajous. En (a) se puede observar que el error es bastante grande al inicio de la trayectoria tanto en el PID como en el SMC, esto se debe a que se parte de la condición inicial $Y_0 = 0$, a los 4(s) el SMC ya se ha estabilizado y se mantiene alrededor de 0, mientras que el PID presenta una etapa transitoria a los 2 (s) y un error de posición de 40 cm, en (b) al sintonizar el PID mediante AG se observa que el error de posición ha disminuido a 30 cm aproximadamente, y en (c) se analiza el error en Y(m) mediante SMC y SMC -AG, donde este último se estabiliza en menor tiempo.

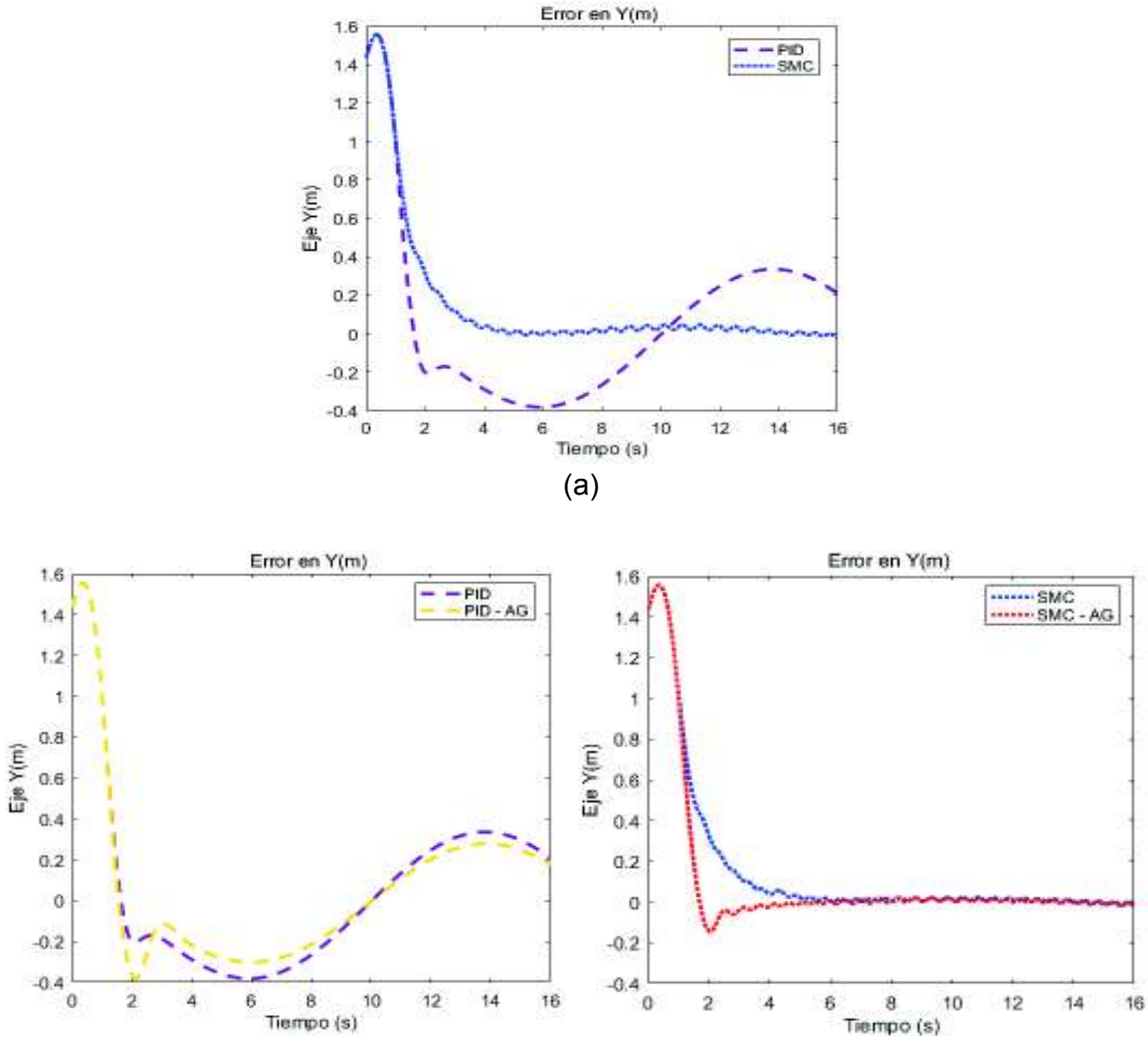


Figura 3.14. Error en Y(m) respecto al Tiempo (s) para una Curva de Lissajous, (a) Mediante PID y SMC, (b) Mediante PID y PID-AG, (c) Mediante SMC y SMC-AG

En la Figura 3.15, se muestran los valores de los índices de desempeño ISE y IAE obtenidos para cada trayectoria. Se observa que los valores del ISE son menores en comparación al IAE tanto para el PID como para el SMC, esto es debido a que el ISE toma mayor peso a los errores grandes que ocurren al inicio de la simulación, mientras que el IAE toma en cuenta todos los errores a lo largo de la simulación[25]. De igual forma se puede decir que el SMC sintonizado con algoritmos genéticos presenta menores valores de índices de desempeño en todas las trayectorias. En el caso de la trayectoria en Y se observa que es la que presenta mayores valores para el ISE y IAE, esto se debe a que es donde se presentan los mayores errores al inicio de la simulación.

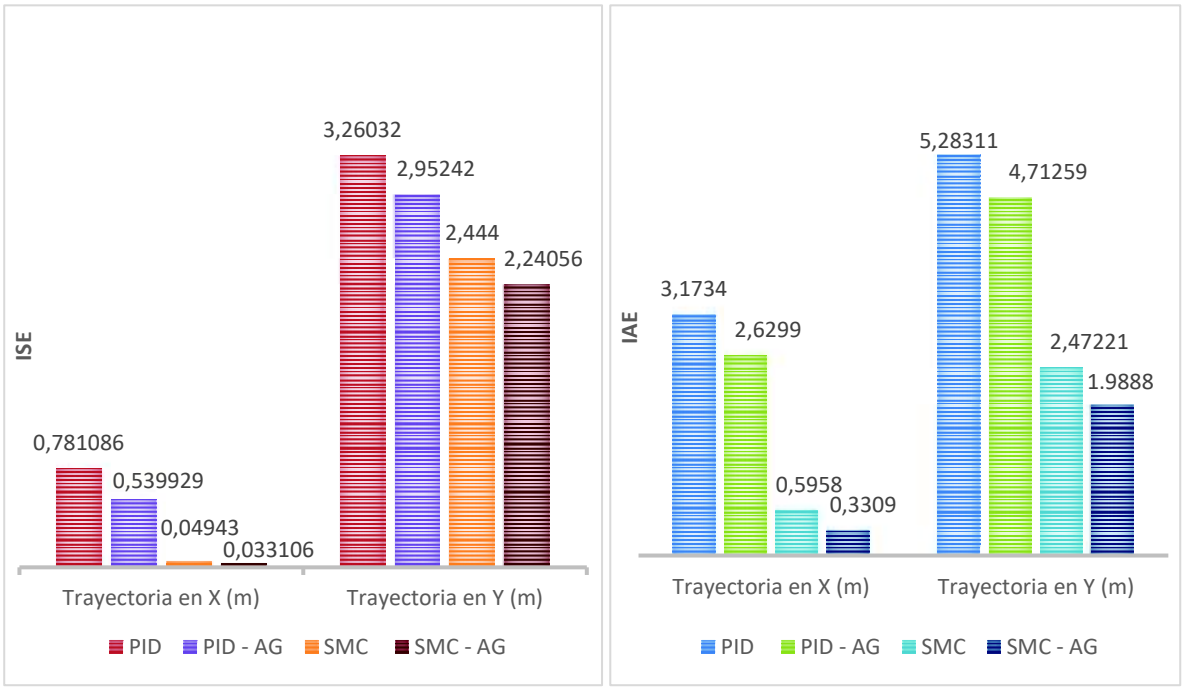


Figura 3.15. Comparación de índices de Desempeño entre PID y SMC sintonizados de forma heurística y mediante Algoritmos Genéticos, para una curva de Lissajous.

En la Tabla 3.4, se visualiza el porcentaje de mejora tanto en el ISE como IAE al comparar los diferentes controladores entre sí, dando como resultado al controlador SMC-AG (SMC sintonizado con Algoritmos Genéticos) con mejor desempeño y con menor desempeño al Controlador PID sintonizado heurísticamente.

Tabla 3.4. Porcentaje de Mejora en las Trayectorias X y Y, para una curva de Lissajous

	Trayectorias	**Porcentaje de Mejora %**		
		SMC Y PID	**PID-AG Y PID**	**SMC-AG Y SMC**
ISE	Trayectoria en X	93.6%	30.87%	33.02%
	Trayectoria en Y	25.16%	9.44%	8.19%
IAE	Trayectoria en X	81.22%	29.57%	44.46%
	Trayectoria en Y	53.20%	10.79%	19.55%

Finalmente, en la Tabla 3.5, se muestran los valores de los parámetros de los controladores PID y SMC en sus dos formas de sintonización.

Tabla 3.5. Parámetros para los controladores PID, SMC, PID - AG y SMC - AG, para el control en X y Y, para una Curva de Lissajous.

	Control en X(m)			Control en Y (m)		
Parámetros	K_P	K_i	K_d	K_P	K_i	K_d
PID	17.076	0.851	7.0952	17.076	0.851	7.0952
PID - AG	16.15338	0.3440	5.5020	16.15338	0.3440	5.5020
	λ_x	K_{Dx}	δ	λ_y	K_{Dy}	δ
SMC	0.858	2	0.03	0.858	2	0.03
SMC - AG	1.7650	3.8807	0.03	1.3151	3.4479	0.03

Seguimiento de Trayectoria para una Sinusoide.

Esta prueba fue realizada utilizando una onda senoidal con una amplitud de 1(m) y frecuencia de 0.14 Hz, con un tiempo de simulación de 16 (s) en los que se simulan más de dos periodos de onda, se parten de los puntos iniciales $X_o = 0$; $Y_o = 0$; $Z_0 = 2$.

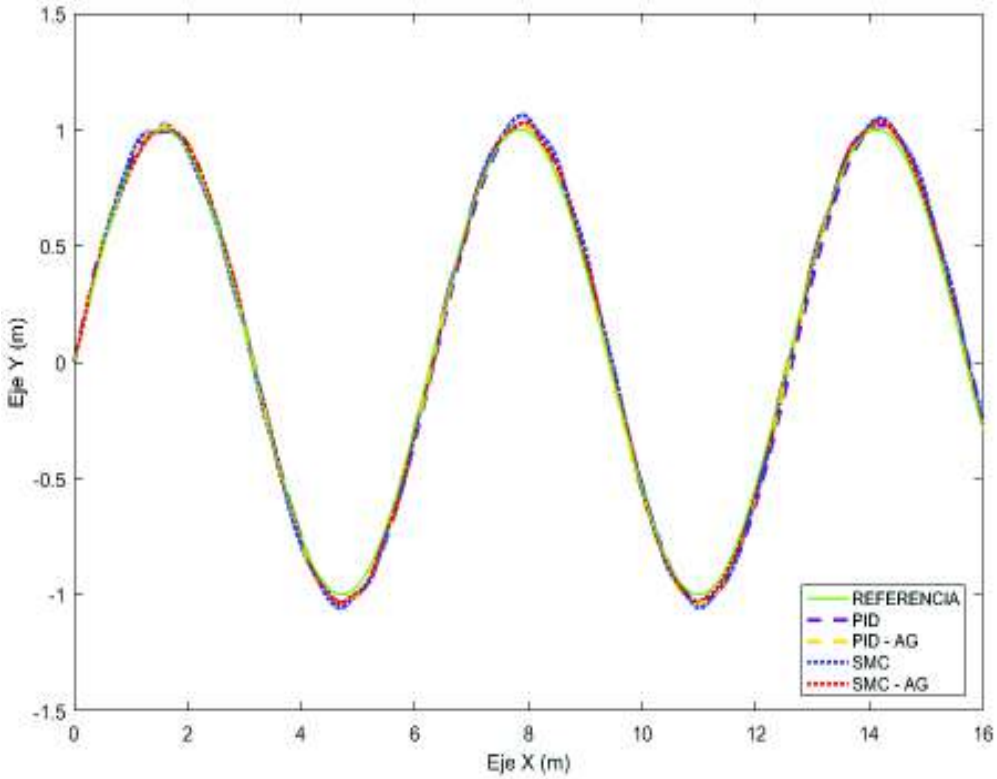


Figura 3.16. Seguimiento de Trayectoria mediante PID, PID-AG, SMC y SMC-AG para una trayectoria Senoidal

En la Figura 3.16, se puede observar que todos los controladores implementados permiten el seguimiento de la trayectoria en forma senoidal, sin embargo, los errores de posición que no son visibles a simple vista serán detallados en las Figura 3.19 y Figura 3.20.

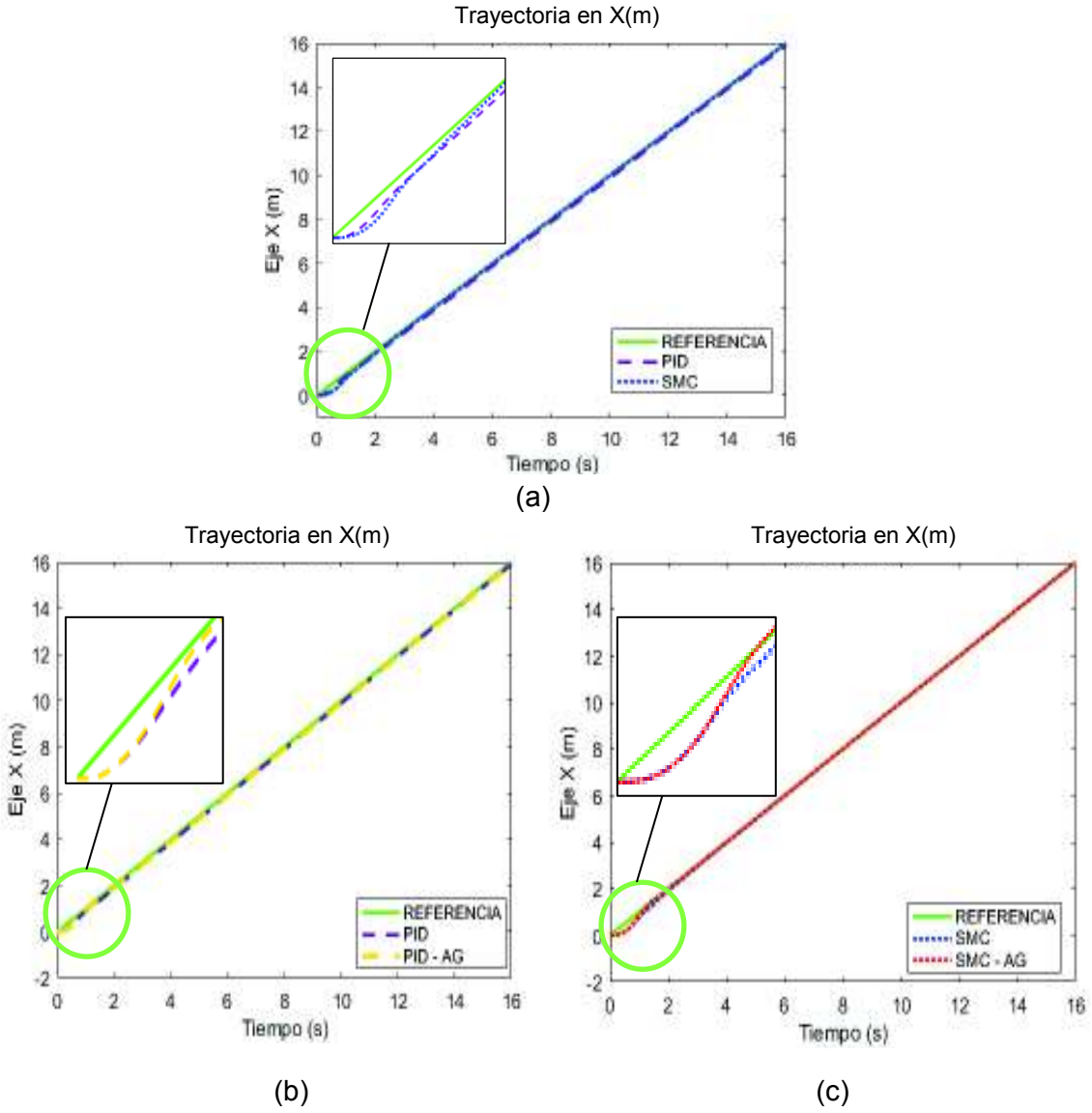


Figura 3.17. Trayectorias en X (m) respecto al Tiempo (s), para una Sinusoide. (a) Mediante PID y SMC, (b) Mediante PID y PID – AG, (c) Mediante SMC y SMC – AG

En la Figura 3.17 ,se observan gráficas lineales las cuales representan el tiempo de duración de la trayectoria que son 16 (s), correspondientes a dos periodos y medio de onda senoidal, en (a) se observa que el PID presenta error de posición puesto que no se encuentra sobrepuesto sobre la línea de referencia a diferencia del SMC que se encuentra sobrepuesto, en (b) no es muy notorio el sobre impulso al inicio de la trayectoria a diferencia de (a), sin embargo, ninguno de las dos sintonizaciones implementadas permite al PID sobreponerse sobre la línea de referencia, en (c) al comparar el SMC sintonizado de forma heurística y

mediante AG se observa que el sobre impulso de este último es mayor alcanzando la trayectoria en menor tiempo. En la Figura 3.18, se presentan las gráficas de las trayectorias en Y, en estas ilustraciones es más fácil observar las diferencias existentes. En (a) se muestra que el PID presenta un error de posición a lo largo de la trayectoria en comparación al SMC que se muestra casi sobre la línea de referencia, sin embargo, también se visualiza que el SMC no alcanza a seguir a la trayectoria en los picos de ésta, generando una etapa transitoria y cuando pasa el pico alcanza nuevamente la trayectoria, en (b) se observa que al sintonizar el PID con Algoritmos Genéticos el error disminuye y se mantiene constante a lo largo de la trayectoria, y en (c) se observa que el SMC–AG en los picos se acerca más a la trayectoria y la alcanza en menor tiempo.

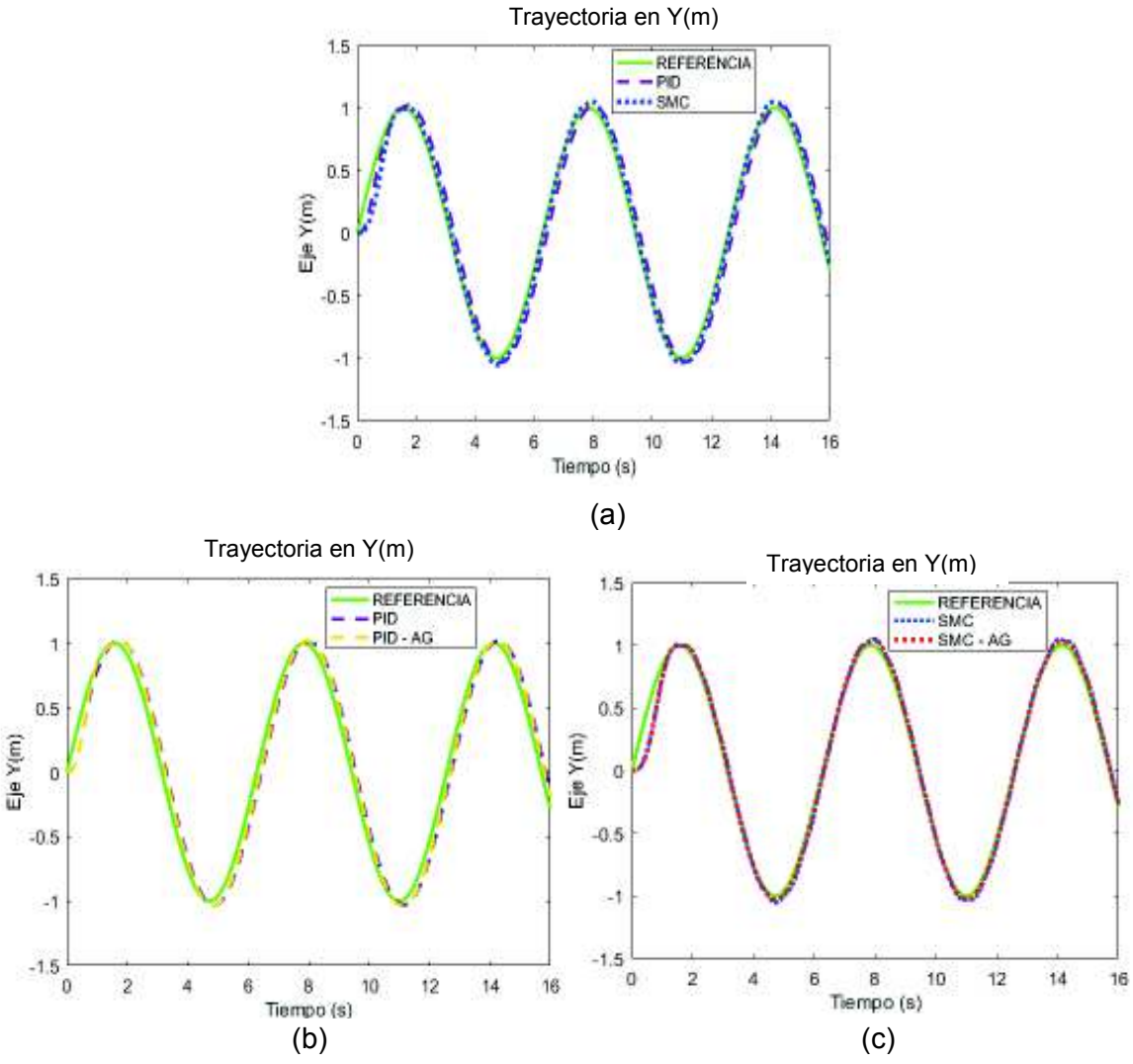


Figura 3.18. Trayectorias en Y (m) respecto al Tiempo (s) para una Sinusoide, (a) Mediante PID y SMC, (b) Mediante PID y PID – AG, (c) Mediante SMC y SMC – AG

En la Figura 3.19, se presenta los errores en X (m) generados por el seguimiento de trayectoria de una Sinusoide. En (a) se visualiza que el error del PID está alrededor de 0.1 metro, equivalente a 10 cm de desplazamiento de la referencia, el SMC presenta mayor error al inicio debido al sobre impulso generado, sin embargo, el error permanece cercano a 0, se observa que existen pequeñas oscilaciones conocidas como chattering que se encuentran por debajo de los 0.05 (m), en (b) se observa que el PID sintonizado con AG presenta una etapa transitoria a los 2 (s), pero el error disminuye, y en (c) el SMC–AG llega en menos tiempo a estabilizarse, manteniendo su error cerca de 0.

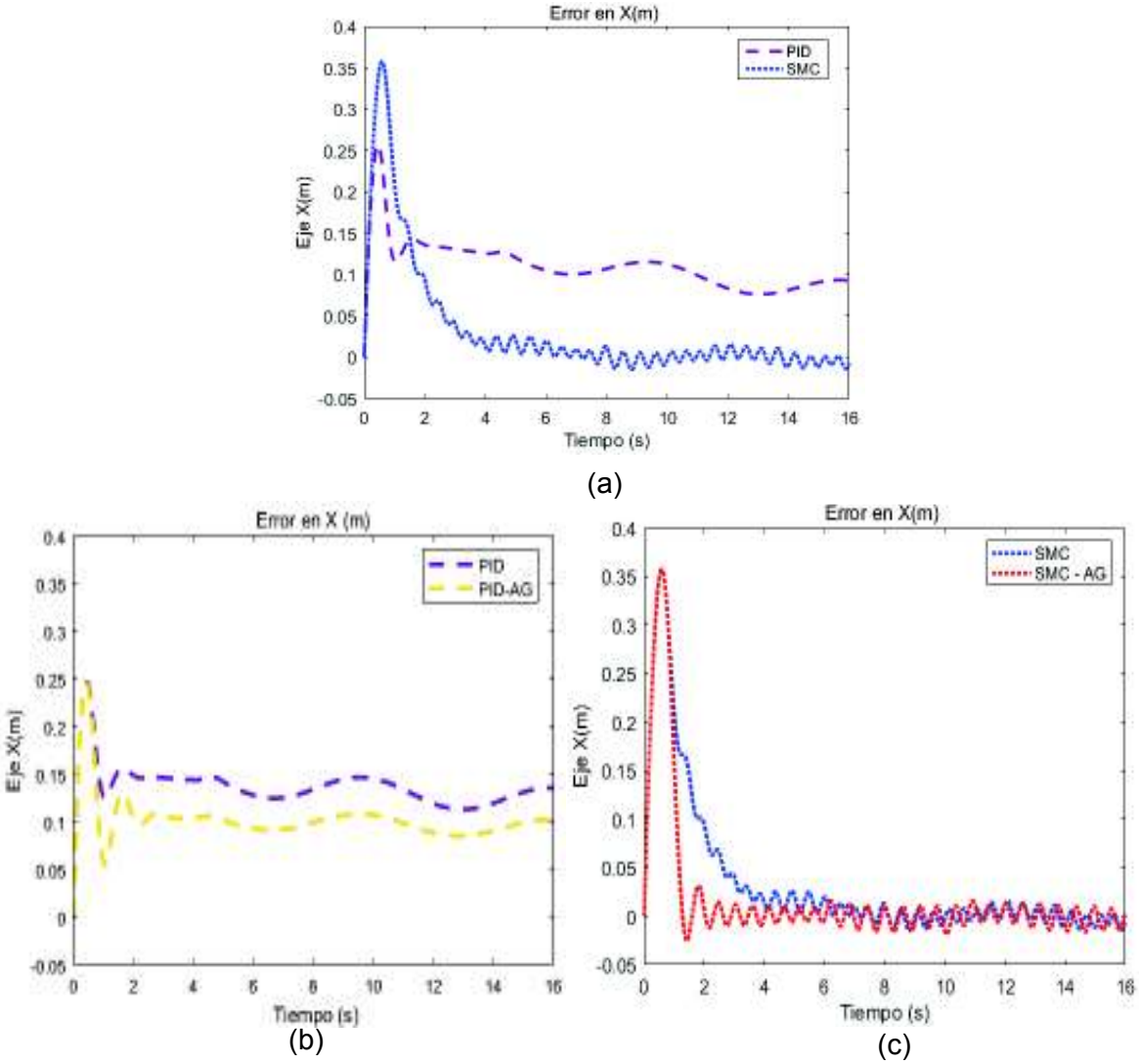


Figura 3.19. Errores en X (m) respecto al Tiempo (s) para una Sinusoide, (a) Mediante PIDY SMC, (b) Mediante PID y PID – AG, (c) Mediante SMC y SMC – AG.

En la Figura 3.20, se presentan los errores de posiciones en Y(m) para una Sinusoide, se puede afirmar que, al sintonizar ambos controladores mediante AG, se mejora el tiempo de establecimiento y error de posición, el chattering que se muestra es característico del

controlador SMC, y no se considera un problema para el sistema dado que se encuentra por debajo de 0.05(m).

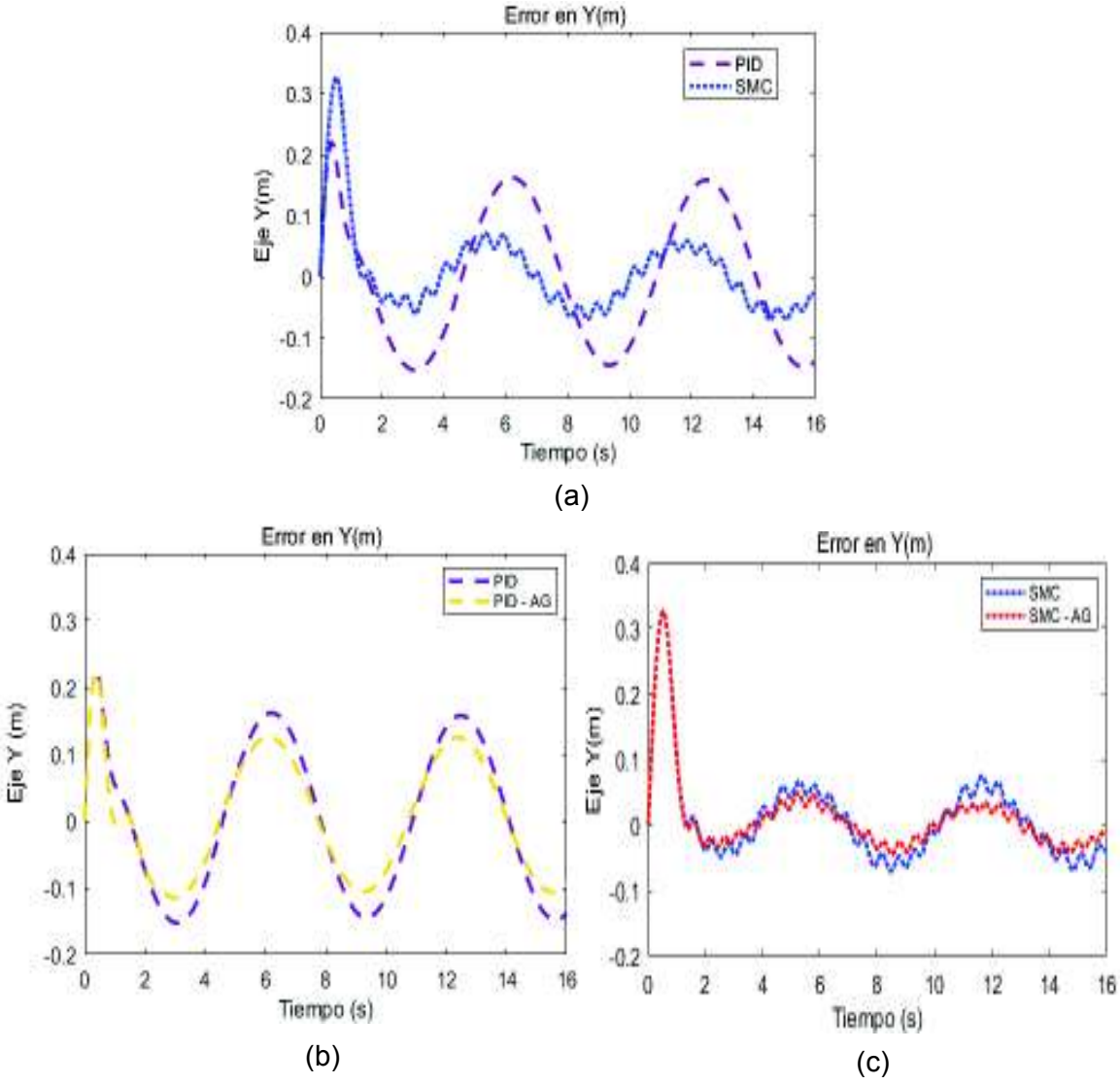


Figura 3.20. Errores en Y (m) respecto al Tiempo (s) para una Sinusoide, (a) Mediante PID y SMC, (b) Mediante PID y PID – AG, (c) Mediante SMC y SMC – AG

En la Figura 3.21 se ha elaborado un esquema de columnas representativas de los índices de desempeño obtenidos, donde se aprecia que el controlador que presenta mejor desempeño es el SMC–AG al tener el menor valor de ISE e IAE tanto en X como en Y, y el controlador que presenta los valores más elevados de ISE e IAE es el PID sintonizado de forma heurística, convirtiéndolo en el controlador con menor desempeño en este proyecto.

En la Tabla 3.6 se muestran los porcentajes de mejora para los controladores PID y SMC, al ser sintonizados de forma heurística y mediante AG, donde se aprecia que el control PID mejora hasta en un 38.24% y el SMC alcanza una mejoría de 31.72%

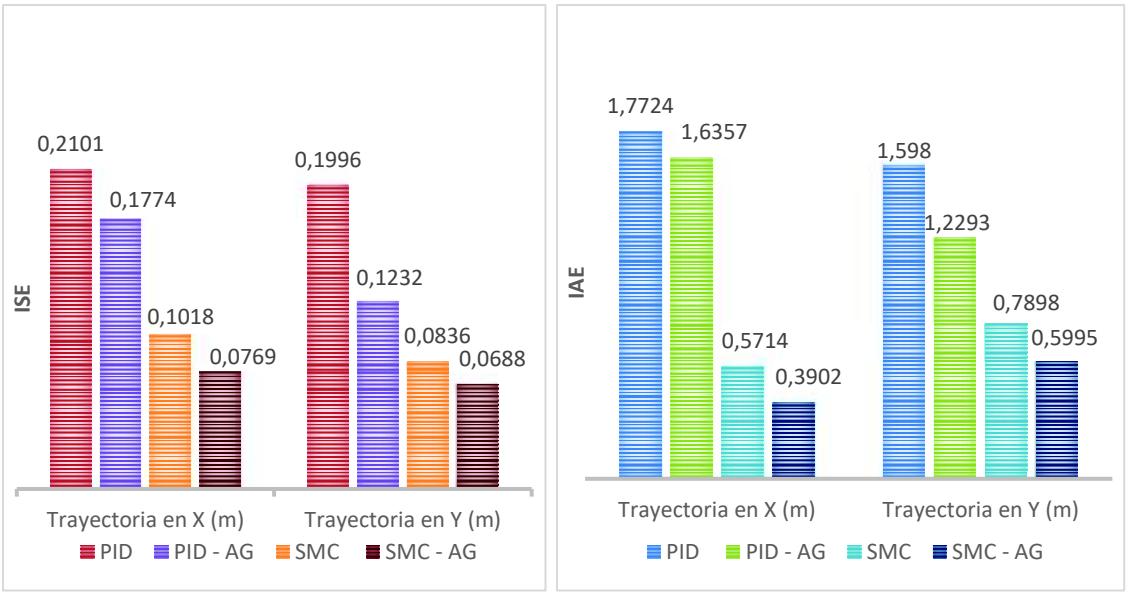


Figura 3.21. Comparación de índices de Desempeño entre PID y SMC sintonizados de forma heurística y mediante Algoritmos Genéticos, para una Sinusoide.

Tabla 3.6. Porcentaje de Mejora en las Trayectorias X y Y, para una Sinusoide.

	Trayectorias	**Porcentaje de Mejora %**		
		SMC Y PID	**PID-AG Y PID**	**SMC-AG Y SMC**
ISE	Trayectoria en X	51.54%	15.56%	24.39%
	Trayectoria en Y	58.12%	38.24%	17.69%
IAE	Trayectoria en X	67.76%	7.72%	31.72%
	Trayectoria en Y	50.56%	23.08%	24.09%

Finalmente, en la Tabla 3.7, se presentan los valores de los parámetros de los controladores PID y SMC en sus dos formas de sintonización.

Tabla 3.7. Parámetros para los controladores PID, SMC, PID - AG y SMC - AG, para el control en X y Y, para una Sinusoide.

	Control en X(m)			**Control en Y (m)**		
Parámetros	K_p	K_i	K_d	K_p	K_i	K_d
PID	40.011	0.56	6.085	40.011	0.56	6.085
PID - AG	39.7817	0.3897	4.3217	39.7817	0.3897	4.3217
	λ_x	K_{Dx}	δ	λ_Y	K_{Dy}	δ
SMC	0.858	2	0.03	0.858	2	0.03
SMC - AG	1.8505	8.2407	0.03	0.9362	6.9879	0.03

Seguimiento de Trayectoria para una Trayectoria Cuadrada.

Esta prueba fue realizada con una trayectoria cuadrada de 8 (m) de lado y con una duración de 32 (s), se parte de las condiciones iniciales $Xo = 0,\ Yo = 0,\ Zo = 2$. En la Figura 3.22, se muestra el seguimiento de Trayectoria mediante los controladores PID y SMC, sintonizados heurísticamente y con Algoritmos Genéticos, posteriormente al igual que en las trayectorias anteriores se muestran gráficas de posiciones y errores en X y Y.

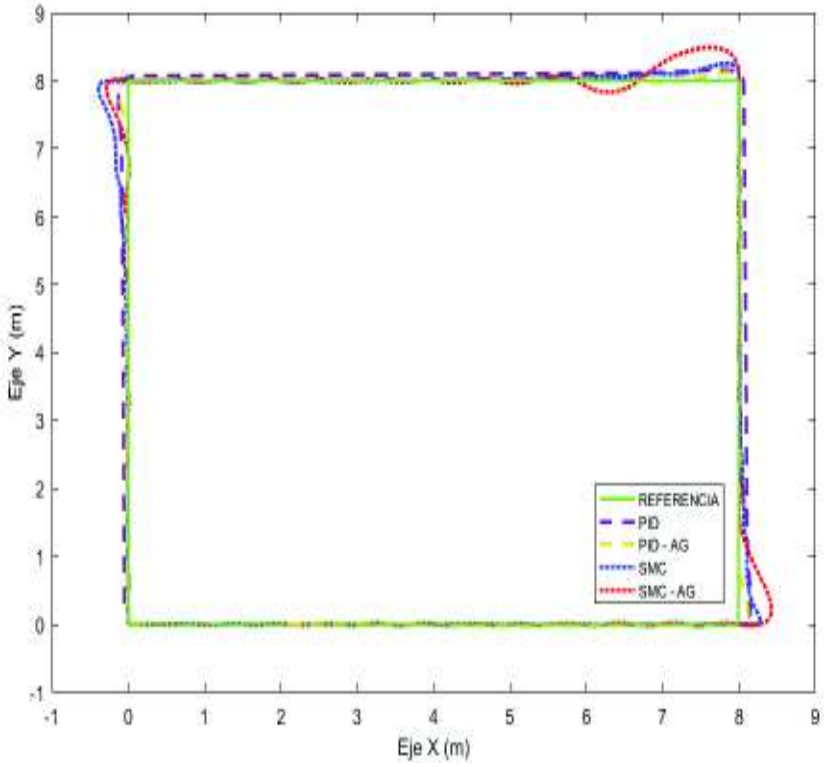


Figura 3.22. Seguimiento de Trayectoria mediante PID, SMC, PID–AG y SMC-AG para una trayectoria cuadrada

En la Figura 3.22, se muestra el seguimiento de una trayectoria cuadrada, a simple vista se observa que el SMC y el SMC–AG, presenta mayor sobre impulso en las esquinas de la trayectoria a diferencia del PID y PID-AG, sin embargo, el SMC sintonizado de ambas formas presenta menor error de posición, cuyo análisis se realizará en las gráficas posteriores.

En la Figura 3.23, se ilustran las gráficas de las trayectorias en X(m), en (a) se puede observar que el SMC presenta mayor sobre impulso, sin embargo, el tiempo de establecimiento y error de posición son menores, en (b) se presentan las gráficas obtenidas mediante PID sintonizado de forma heurística y con AG, este última presenta menor error de posición, y en (c) se logra visualizar que el sobre impulso en SMC–AG es mayor al SMC, alcanzando la referencia en menor tiempo y presentando menor error de posición.

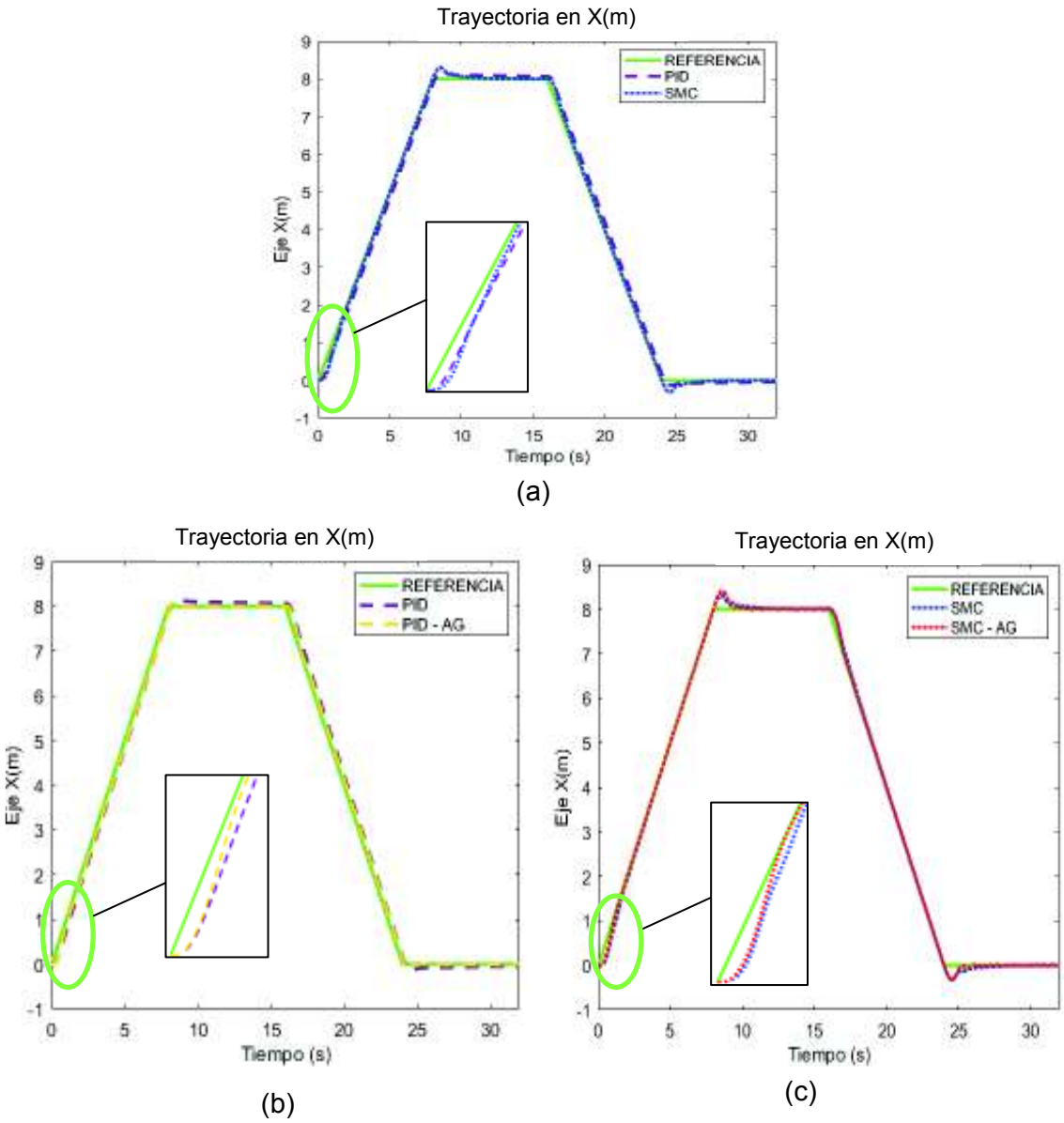


Figura 3.23. Trayectorias en X(m) respecto al Tiempo (s) para una Trayectoria Cuadrada, (a) Mediante PID y SMC, (b) Mediante PID y PID-AG, (c) Mediante SMC y SMC–AG

En la Figura 3.24, se ilustran las gráficas de las trayectorias en Y (m), obtenidas para una trayectoria cuadrada, en (a) el SMC presenta sobre impulso en las esquinas de la trayectoria, sin embargo, la misma se encuentra casi sobre la referencia a diferencia del PID que presenta un error de posición constante, en (b) al comparar las dos formas de sintonización del PID, se observa que la sintonización mediante AG genera menor error de posición, en (c) el SMC con AG alcanza la referencia en menor tiempo.

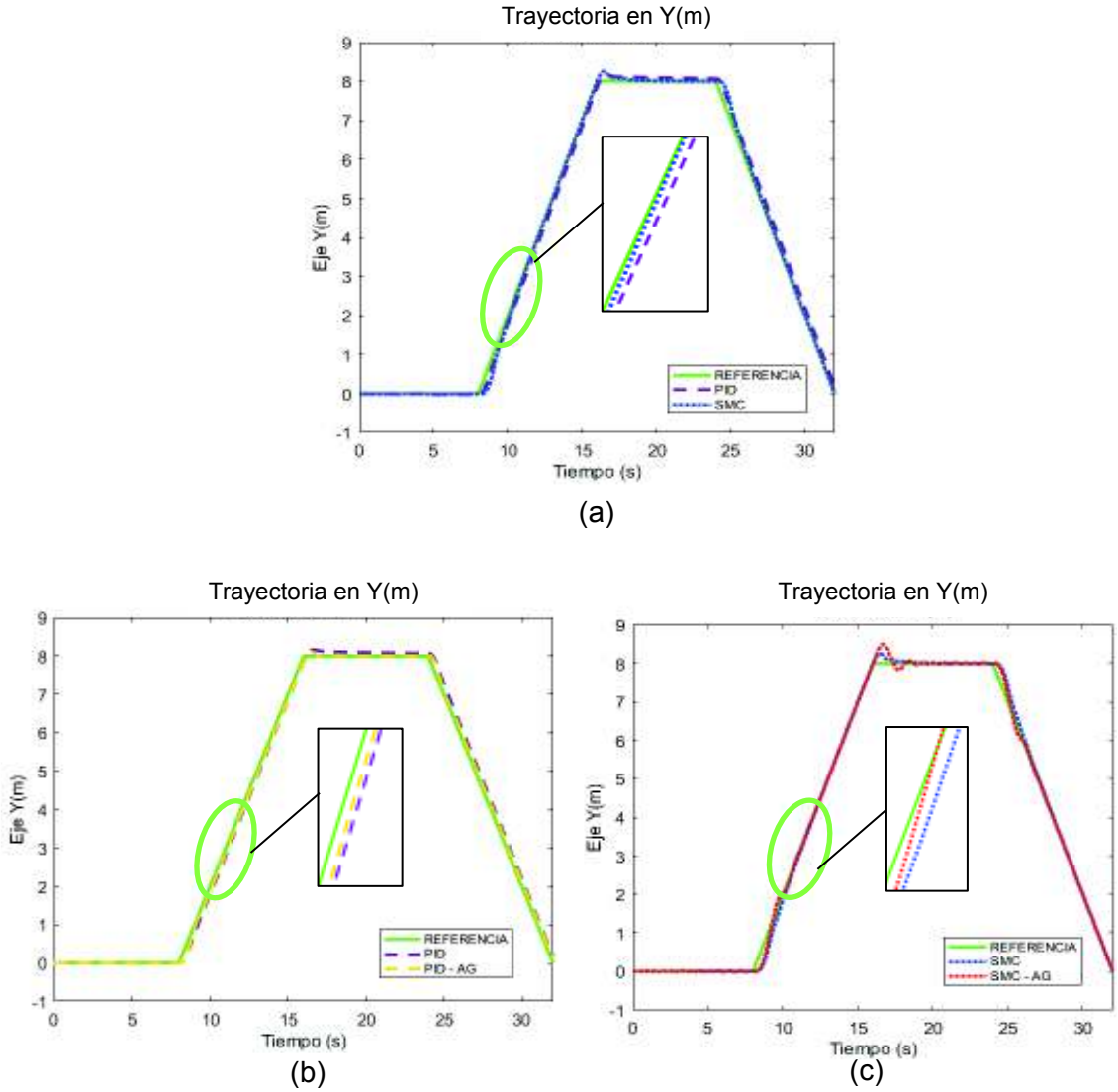


Figura 3.24. Trayectorias en Y(m) con respecto al Tiempo (s) para una Trayectoria Cuadrada, (a) Mediante PID y SMC, (b) Mediante PID y PID–AG, (c) mediante SMC y SMC-AG.

En la Figura 3.25, se muestran los errores en X(m) para la Trayectoria Cuadrada, en (a) se ilustra que el sobre impulso generado por el SMC llega a 0.4 (m), sin embargo, el error tiende a cero, a diferencia del PID, que tiene menor sobre impulso pero el error de posición se mantiene constante alrededor de 0.15 (m), en (b) al comparar el controlador PID sintonizado de forma heurística y mediante AG, se muestra que en la sintonización con AG el sobre impulso y el error de posición han disminuido, sin embargo, se presenta una etapa transitoria luego de cada esquina y posteriormente se estabiliza manteniendo un error constante, en (c) al analizar el SMC-AG se aprecia que en dos esquinas el sobre impulso llega de igual forma

a 0.4 (m), pero el tiempo de establecimiento disminuye y el error de posición se mantiene más cerca de 0.

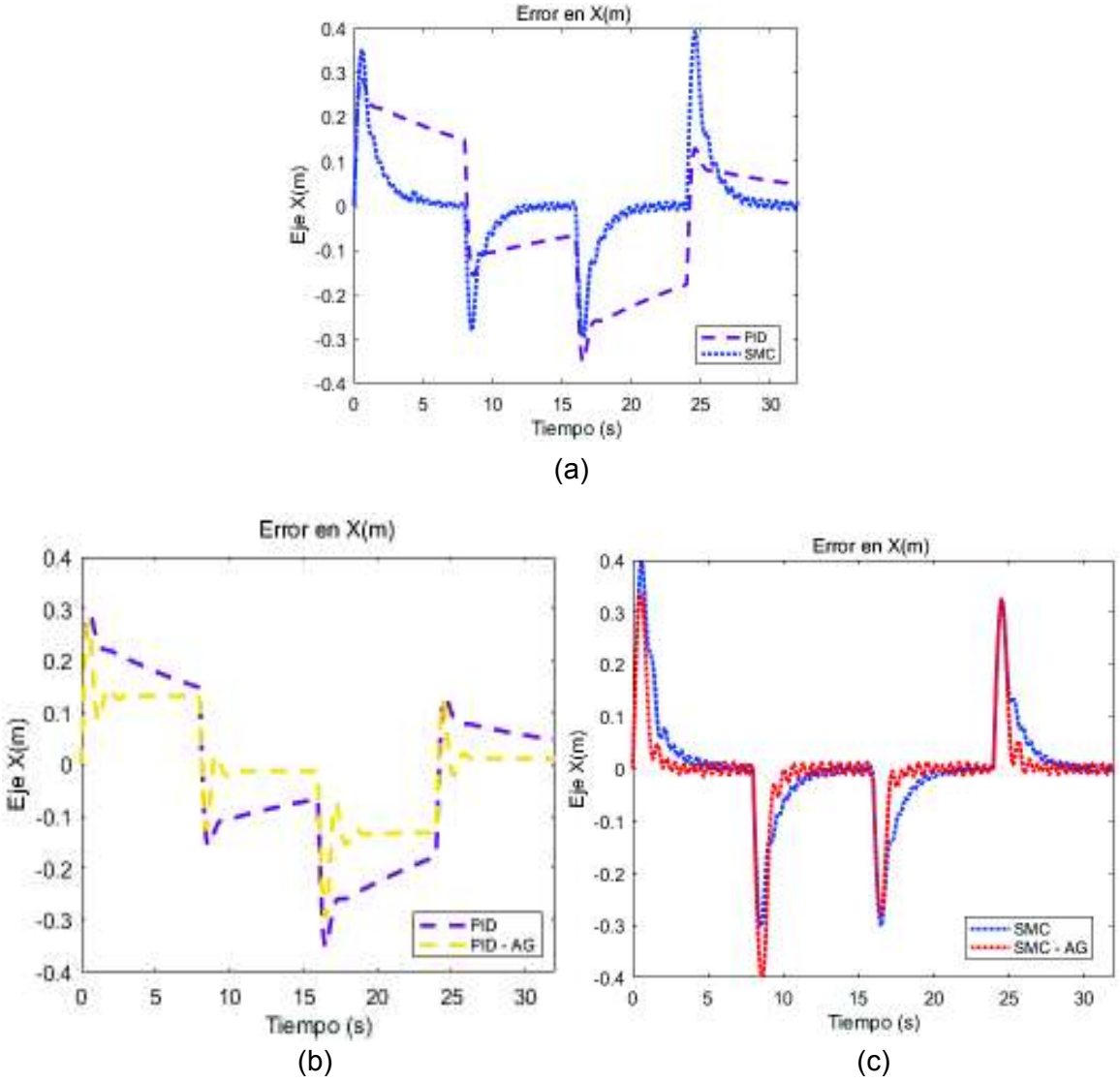


Figura 3.25. Errores en X (m) respecto al Tiempo (s) para una Trayectoria Cuadrada, (a) Mediante PID Y SMC, (b) Mediante PID y PID - AG, (c) Mediante SMC y SMC – AG

En la Figura 3.26, se muestran los errores en Y(m) para la Trayectoria Cuadrada, en (a) el SMC presenta un elevado sobre impulso en comparación al PID, sin embargo, a diferencia del PID, el error tiende a cero, en (b) al comparar el controlador PID sintonizado de forma heurística y mediante AG, se muestra que en la sintonización con AG el sobre impulso y el error de posición son menores, pero se presenta una etapa transitoria luego de cada esquina y posteriormente se estabiliza manteniendo un error constante, en (c) al analizar el SMC-AG se aprecia que el sobre impulso, el tiempo de establecimiento y el error de posición son menores. En la Figura 3.27, se muestra la comparación de los índices de desempeño obtenidos para cada posición, siendo el SMC-AG el controlador con mejor desempeño.

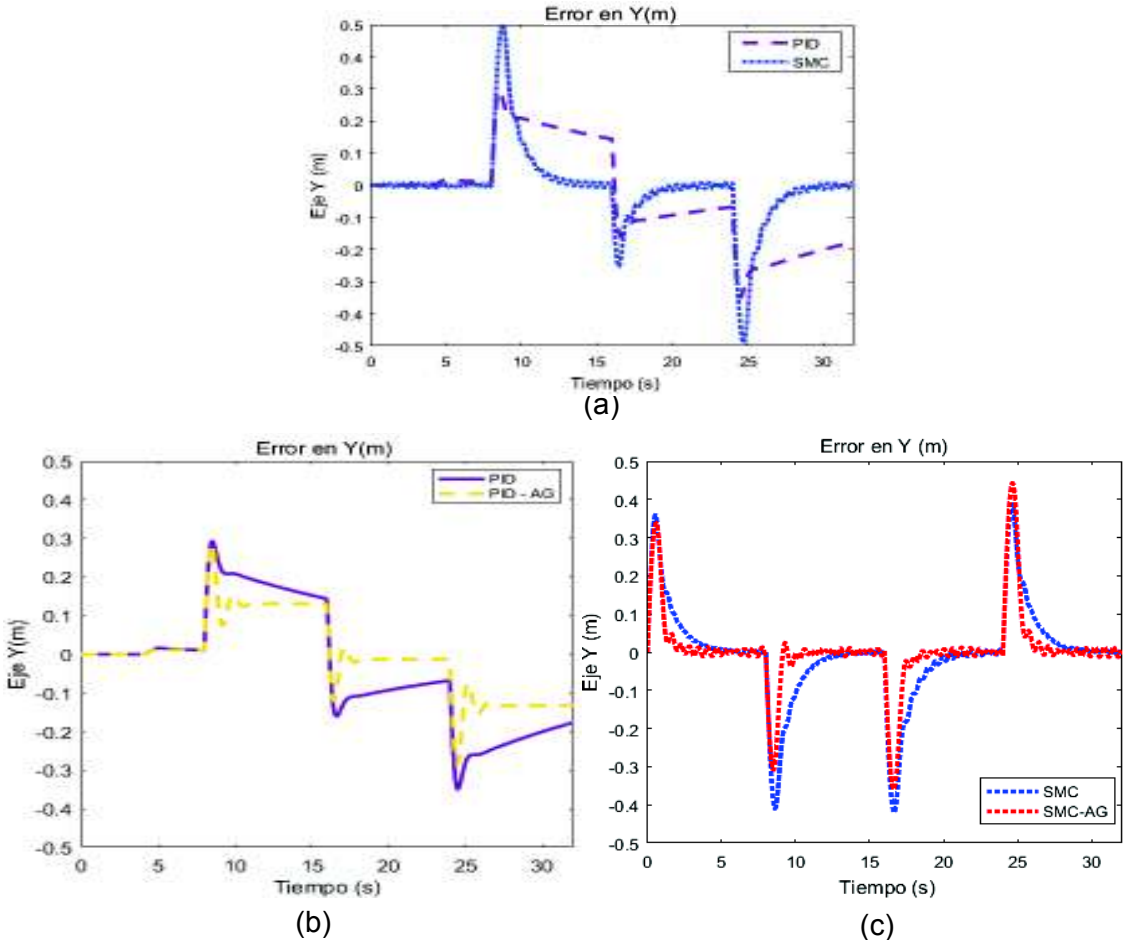


Figura 3.26. Errores en Y (m) respecto al Tiempo (s) para una Trayectoria Cuadrada, (a) Mediante PID y SMC, (b) Mediante PIDy PID–AG, (c) Mediante SMC y SMC-AG

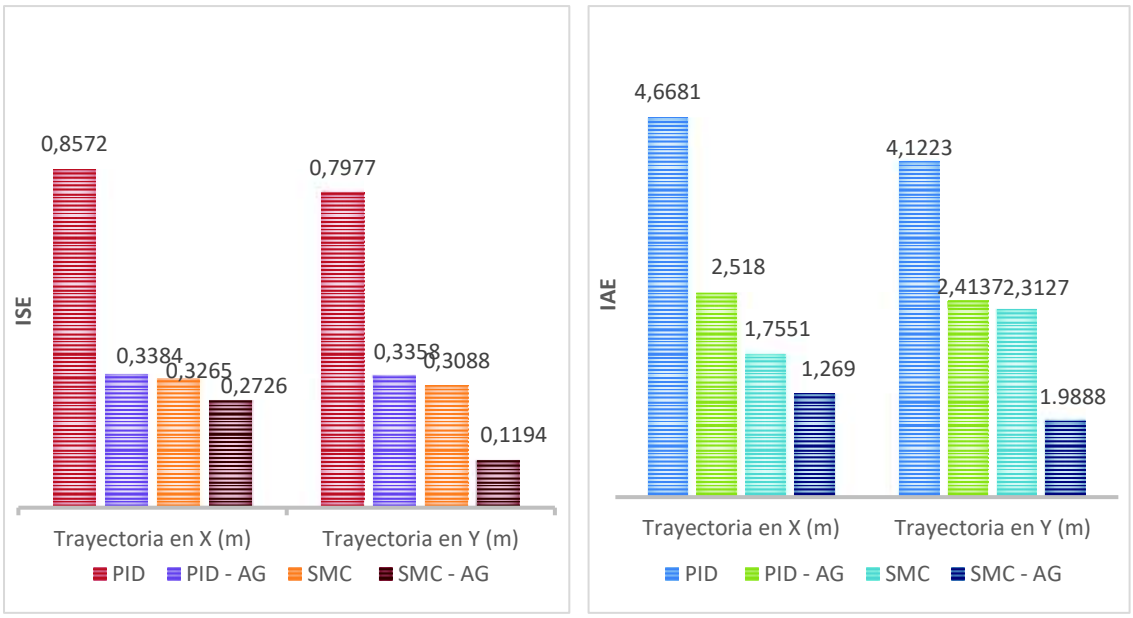


Figura 3.27. Comparación de índices de Desempeño entre PID y SMC sintonizados de forma heurística y mediante Algoritmos Genéticos.

En la Tabla 3.8, se muestran las mejoras obtenidas al utilizar los algoritmos genéticos como método de sintonización en los controladores PID y SMC, para el seguimiento de trayectoria cuadrada del cuadricóptero.

Tabla 3.8. Porcentaje de Mejora en las Posiciones X y Y, para una Trayectoria Cuadrada.

	Trayectorias	**Porcentaje de Mejora %**		
		SMC y PID	**PID-AG y PID**	**SMC-AG y SMC**
ISE	Trayectoria en X	61.91%	60.52%	16.50%
	Trayectoria en Y	61.28%	57.90%	61.33%
IAE	Trayectoria en X	62.40%	46.06%	27.70%
	Trayectoria en Y	43.89%	41.45%	59.26%

Finalmente, en la Tabla 3.9, se presentan los valores de los parámetros de sintonización de los controladores PID, SMC, PID-AG y SMC-AG.

Tabla 3.9. Parámetros para los controladores PID, SMC, PID - AG y SMC - AG, para el control en X y Y, de una trayectoria cuadrada.

	Control en X(m)			**Control en Y (m)**		
Parámetros	K_p	K_i	K_d	K_p	K_i	K_d
PID	29.076	1.851	7.0952	29.076	1.851	7.0952
PID - AG	29.2150	0.0395	3.8794	29.2150	0.0395	3.8794
	λ_x	K_{Dx}	δ	λ_y	K_{Dy}	δ
SMC	0.858	2	0.03	0.858	2	0.03
SMC - AG	1.6384	9.9896	0.03	2.4830	7.7393	0.03

Seguimiento de Trayectoria para una Trayectoria Helicoidal.

Esta prueba fue realizada con el objetivo de mover al cuadricóptero en el espacio, puesto que las trayectorias anteriores estaban fijas en el plano Z. Esta trayectoria tiene como base la curva de Lissajous por lo que no fue necesario sintonizar los parámetros de control mediante AG, dado que se utilizaron los mismos parámetros que para la trayectoria mencionada.
En esta prueba se parte de las condiciones iniciales $X_0 = 0,\ Y_0 = 0,\ Z_0 = 0$, con 70 (s) de simulación.

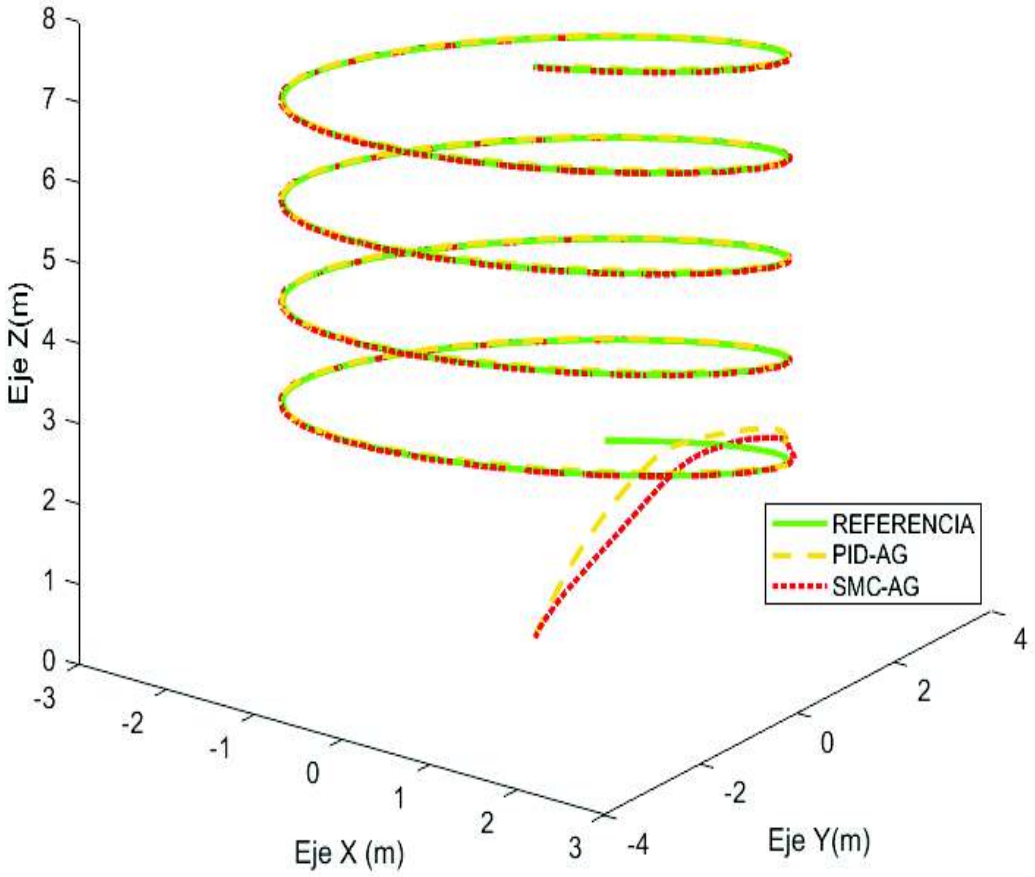


Figure 3.28. Seguimiento de una Trayectoria Helicoidal, mediante PID - AG y SMC -AG

En la Figura 3.29, se muestran las gráficas de las posiciones en X, Y y Z, donde se observa a simple vista que en las posiciones X y Y los controladores PID-AG y SMC-AG siguen la trayectoria sin problema alguno, sin embargo, el controlador que mejor desempeño presenta es el SMC-AG dado que posee índices de desempeño más bajos que el PID-AG, esto se muestra en la Figura 3.31, donde se presenta una comparación en columnas de los índices de desempeño obtenidos, en la posición en Y se aprecia que al inicio de la trayectoria ambos controladores presentan sobre impulso, pero a los 4 (s) consiguen estabilizarse, la línea amarilla que muestra el seguimiento efectuado por el PID-AG, se encuentra a un lado de la línea de referencia, a diferencia de la línea roja que se visualiza casi sobre puesta a la referencia. En la posición en Z se muestra la gráfica de la altura mediante SMC-AG, en este caso se trabaja con el punto inicial $Z_0 = 0$, donde se aprecia que el cuadricóptero alcanza la referencia de altura a los 3 (s) aproximadamente. En esta trayectoria a diferencia de las anteriores, no se realizó comparación entre los controladores PID y SMC, debido a que utiliza la misma sintonización que para la curva de Lissajous, por lo que la idea es compararla entre los dos mejores controladores que son los sintonizados con Algoritmos Genéticos.

En (c), solo se compara la referencia con el controlador de altura SMC-AG, debido a que para todas las trayectorias se trabajó directamente con el controlador de altura en Modo Deslizantes sintonizado con Algoritmos Genéticos.

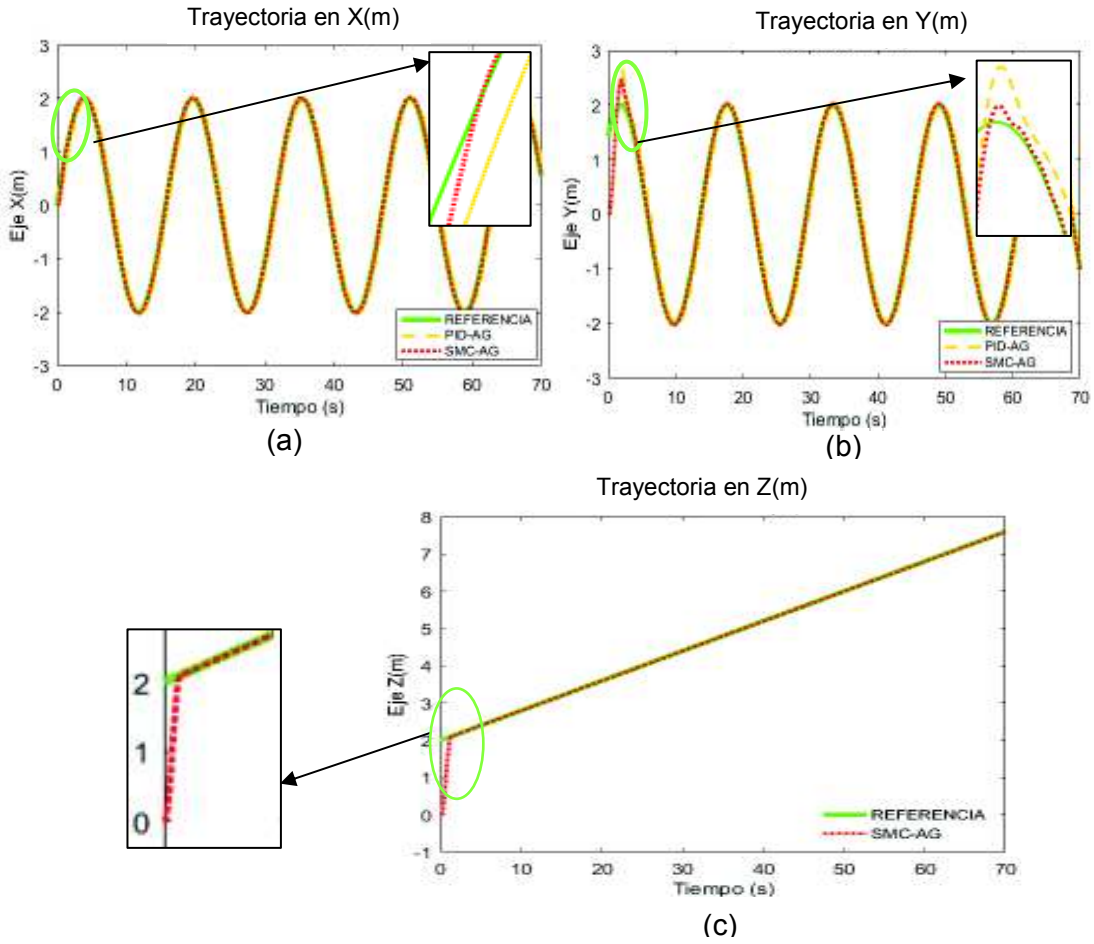


Figura 3.29 Trayectoria Helicoidal respecto al Tiempo (s), (a) Trayectoria en X (m) respecto al Tiempo (s), (b) Trayectoria en Y (m) respecto al Tiempo (s), (c) Trayectoria en Z(m) respecto al Tiempo (s)

En la Figura 3.30, se muestran los errores en cada uno de los ejes, se puede observar en X que el SMC-AG presenta menor error de posición puesto que se mantiene alrededor de cero, mientras que, el PID -AG mantiene un error constante a lo largo de la trayectoria alrededor de los 0.25 (m), también se aprecian las pequeñas oscilaciones del controlador SMC-AG que se encuentran por debajo de los 0.05 (m), por lo que no se considera un problema, en (b) al observar el error en Y(m) se puede afirmar que el SMC-AG presenta menor error de posición, al visualizar el error en Z (m) al inicio de la trayectoria es de 2(m), hasta alcanzar la referencia y posteriormente presenta un error muy cercano a 0. En la Figura 3.31, se muestran los valores de los índices de desempeño ISE e IAE obtenidos por el seguimiento de una

trayectoria helicoidal, los cuales ubican al SMC-AG como el controlador con mejor desempeño para este problema.

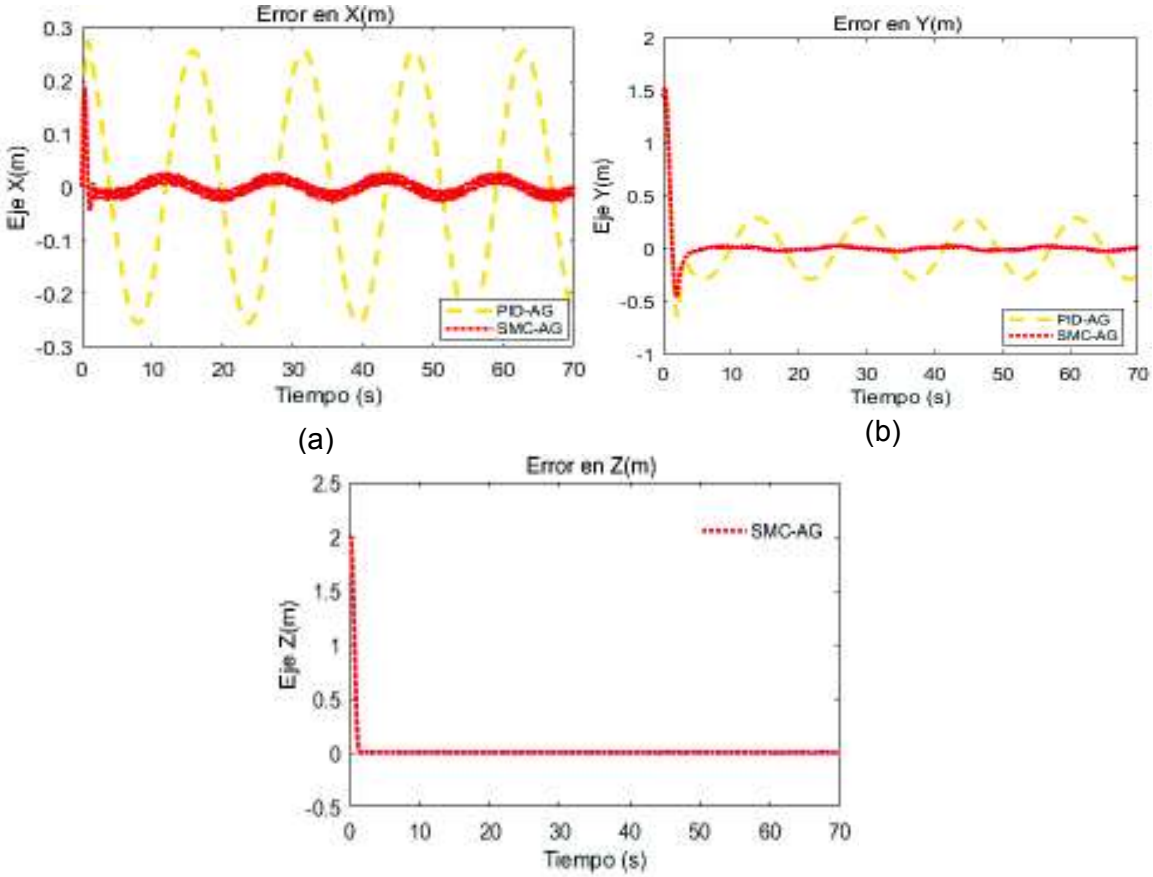


Figura 3.30. Errores de posición en X, Y y Z, de la Trayectoria Helicoidal respecto al Tiempo (s), (a) Error en X (m) respecto al Tiempo (s), (b) Error en Y (m) respecto al Tiempo (s), (c) Error en Z (m) respecto al Tiempo (s)

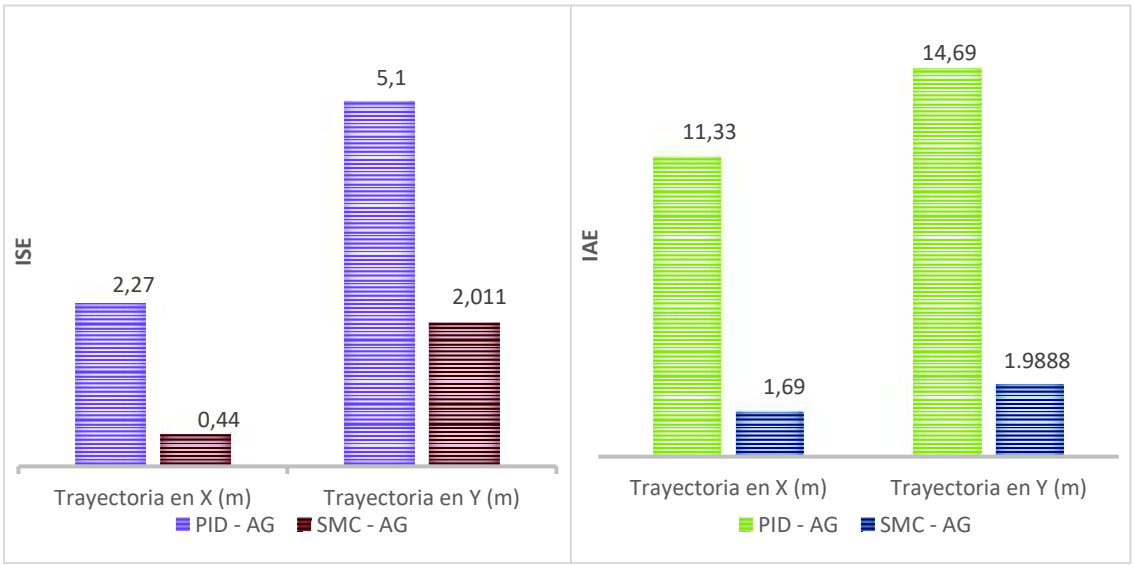


Figura 3.31. Comparación de índices de Desempeño entre PID y SMC sintonizados mediante Algoritmos Genéticos, para una trayectoria Helicoidal.

3.5 Pruebas de Robustez

Al desarrollar un sistema de control, éste siempre incluye algún grado de incertidumbre dado que puede cambiar de manera inesperada o estar sujeto a perturbaciones imprevistas, debido a factores como cambios en los parámetros, en los puntos de operación y perturbaciones no predichas [41]. En el presente trabajo para analizar la robustez del sistema se ha optado por hacer cambios en la masa del cuadricóptero e ingresar perturbaciones al seguimiento de trayectoria del cuadricóptero sin modificar los parámetros de sintonización.

Cambios de Masa

En la Figura 3.32, se muestran los cambios de masa y cambios de ISE en función del tiempo, posteriormente en la Tabla 3. 10, se detallan los valores de ISE donde se puede apreciar que el cuadricóptero soporta hasta un 63.46% de la masa adicional que corresponde a 0.35 Kg adicionales. El control para altura es realizado mediante Control en Modos Deslizantes (SMC) sintonizado con Algoritmos Genéticos. Esta prueba fue aplicada al cuadricóptero cuando este se encontraba estable a 2 metros de altura.

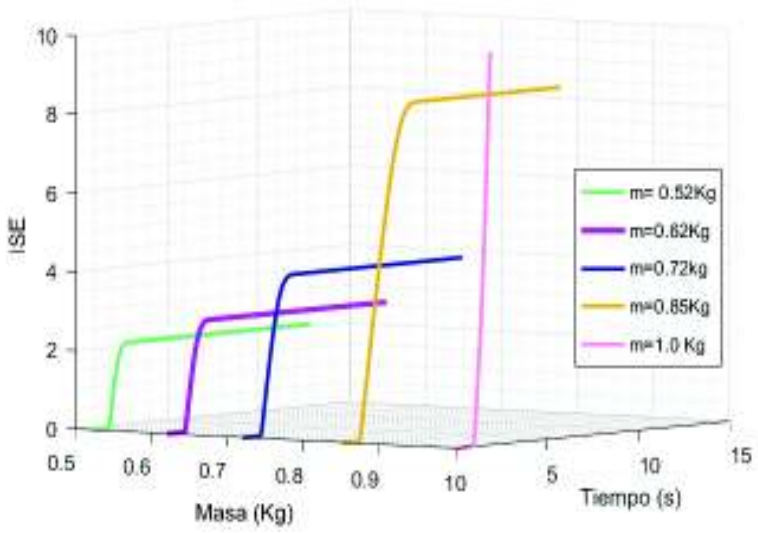


Figura 3.32. Cambios de Masa del Cuadricóptero respecto al ISE de altura

Tabla 3. 10 Cambios de Masa del Cuadricóptero respecto al ISE de altura

Cambios de Masa	ISE
m=0.52 Kg	2.13
m=0.62 Kg	2.79
m=0.72 Kg	4.02
m=0.85 Kg	8.48
m=1 Kg	----------

Rechazo a Perturbaciones

Otro de los puntos importantes para el análisis de robustez de un sistema de control corresponde al ingreso de perturbaciones, en las siguientes secciones se analizarán las respuestas de los controladores PID, PID–AG, SMC Y SMC–AG frente a las mismas.

Perturbaciones a una Curva de Lissajous

En la Figura 3.33, se observa la forma del seguimiento de trayectoria de una Curva de Lissajous con perturbaciones. Se aplicaron dos perturbaciones con una duración de 1.5 (s) cada una y una amplitud de 0.1 m equivalente a un 42.55% respecto a la longitud de brazos del cuadricóptero los mismos que miden 0.235 m, la primera perturbación se ejecuta a los 8 (s) y la segunda a los 13.5 (s).

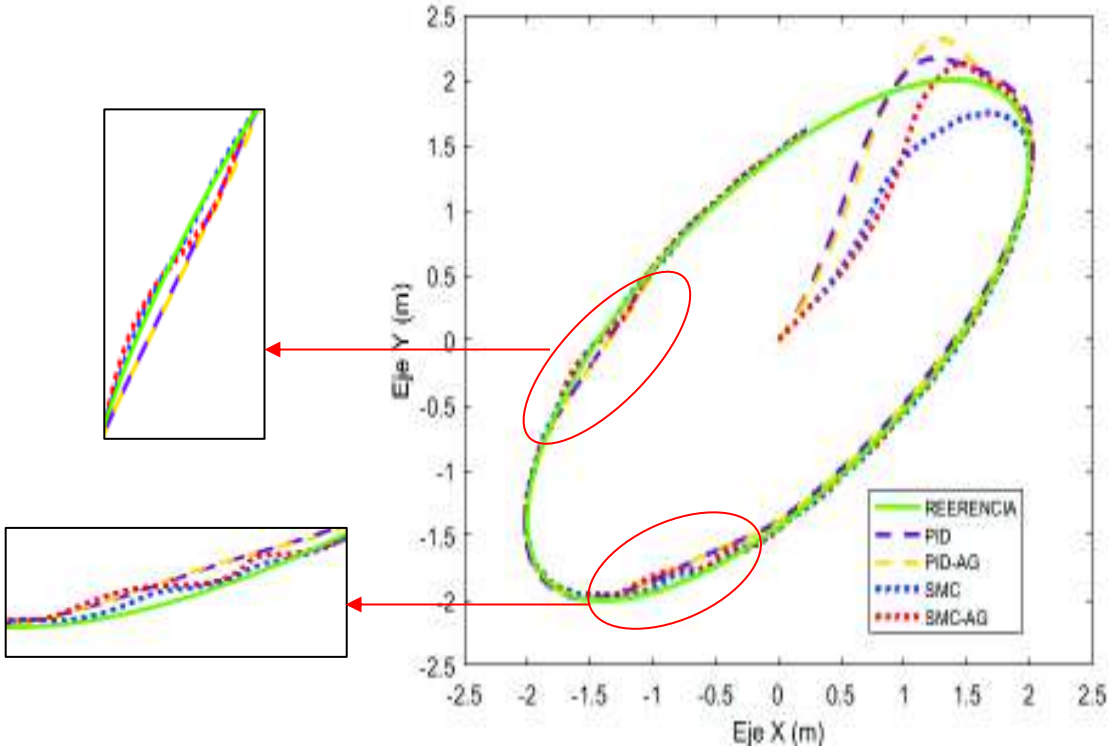


Figura 3.33. Seguimiento de Trayectoria de una Curva de Lissajous con Perturbaciones.

En la Figura 3.34 y Figura 3.35,se observan las gráficas de las posiciones en X (m) y Y(m), con los diferentes controladores y sus respectivas sintonizaciones, ambos controladores reaccionan bien ante perturbaciones, sin embargo, el SMC sintonizado con Algoritmos Genéticos es el que se estabiliza en menor tiempo luego de haber superado la perturbación.

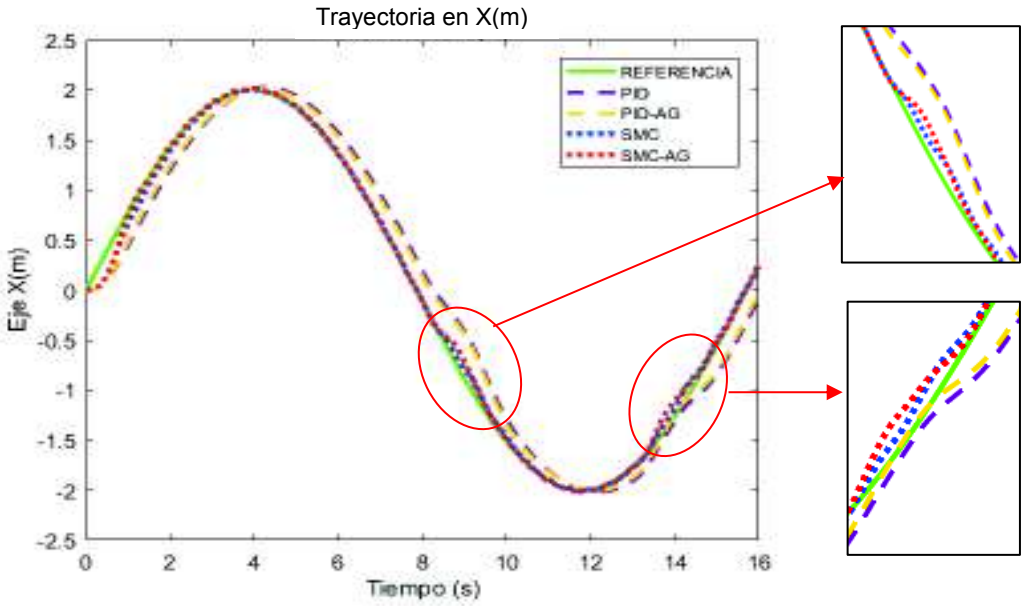


Figura 3.34. Trayectoria en X(m) respecto al Tiempo (s), para una Curva de Lissajous frente a perturbaciones.

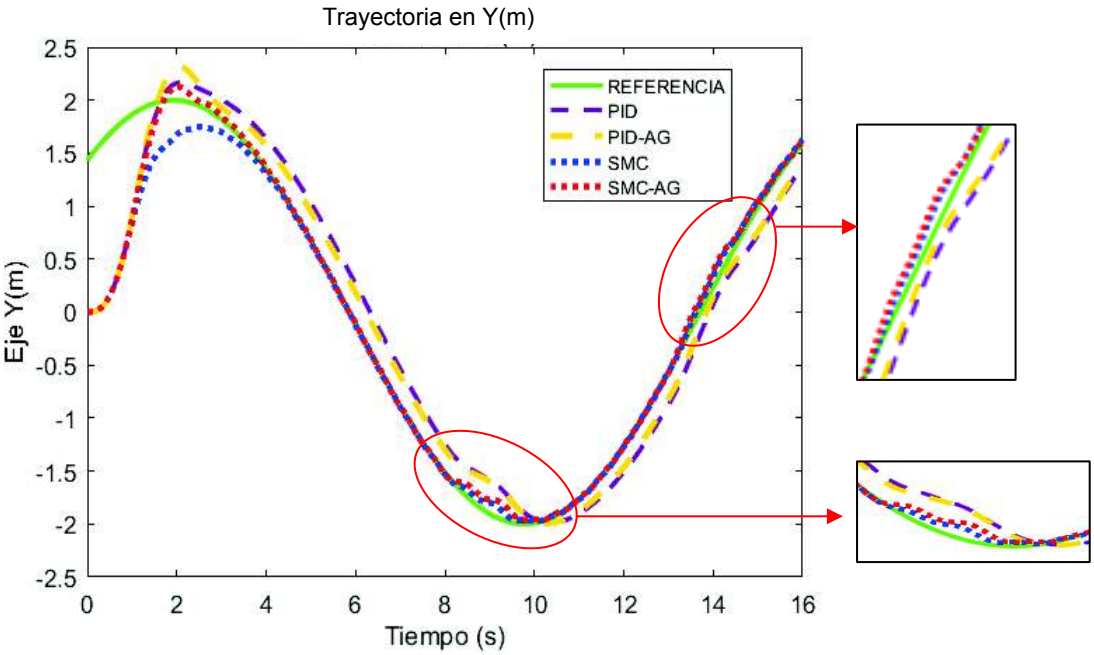


Figura 3.35. Posición en Y (m) respecto al Tiempo, para una Curva de Lissajous frente a perturbaciones.

En la Figura 3.36, se muestran los índices de desempeño obtenidos por el Seguimiento de Trayectoria de una curva de Lissajous con Perturbaciones, el controlador que presenta mejor desempeño es el SMC sintonizado con Algoritmos Genéticos.

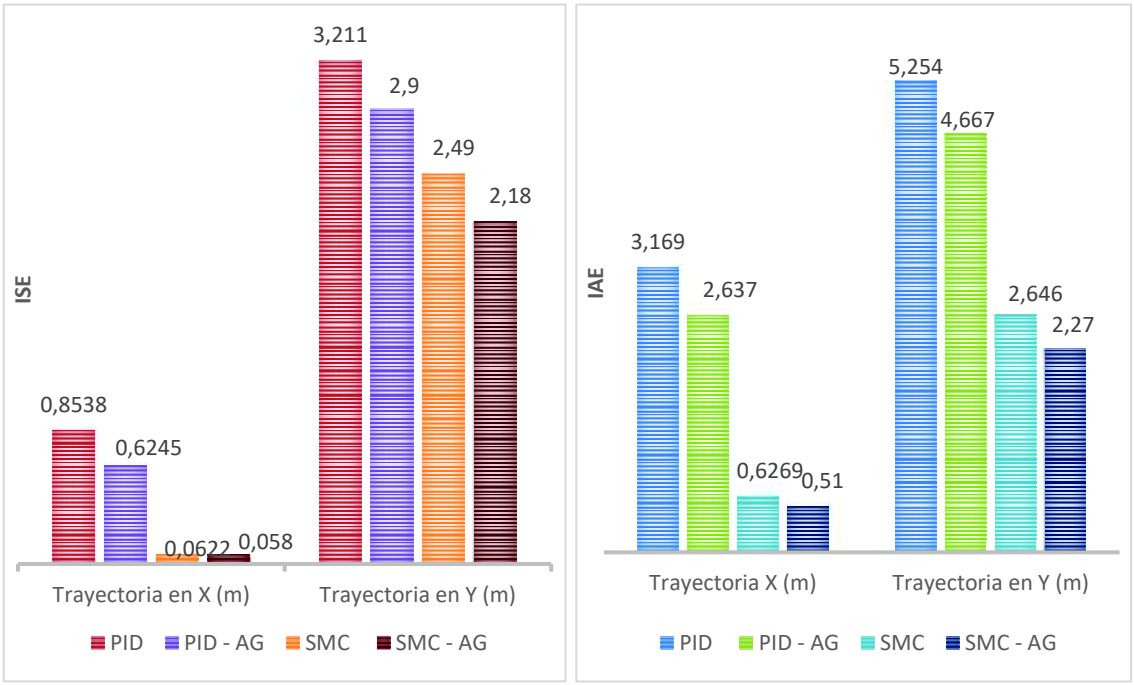


Figura 3.36. Comparación de los índices de desempeño obtenidos por el seguimiento de trayectoria de una curva de Lissajous con Perturbaciones.

Perturbaciones a una Sinusoide

En la Figura 3.37, se presenta el seguimiento de trayectoria de una sinusoide con perturbaciones. Se aplicaron dos perturbaciones con una duración de 1.5 (s) y amplitud de 0.1 m, la primera perturbación se ejecuta a los 8 (s) y la segunda a los 13.5 (s).

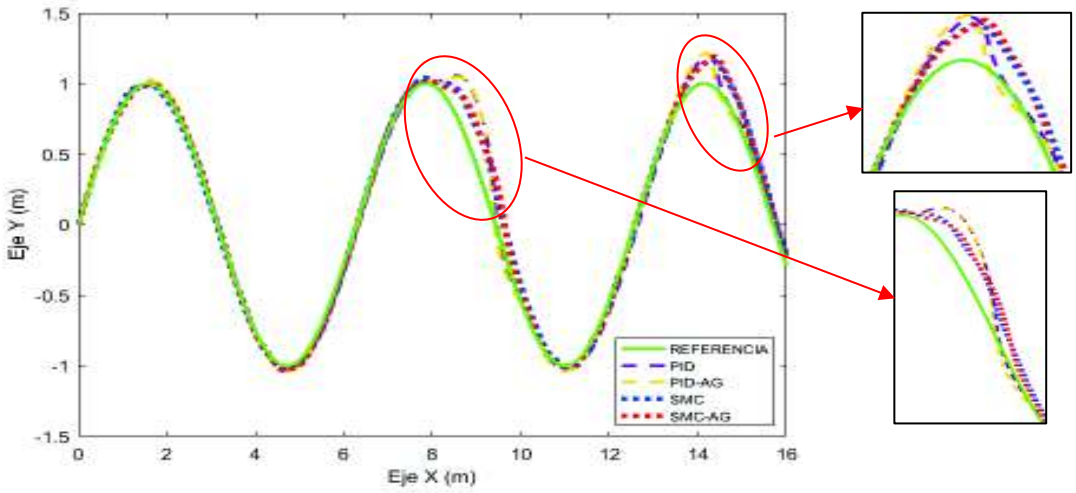


Figura 3.37. Seguimiento de una Trayectoria Senoidal con Perturbaciones.

En la Figura 3.38 y Figura 3.39, se presentan las gráficas de las trayectorias en X (m) y Y(m) de la trayectoria senoidal con perturbaciones, implementándose los controladores PID, SMC

y sus respectivas sintonizaciones, ambos controladores reaccionan bien ante perturbaciones, pero de acuerdo a los índices de desempeño obtenidos y mostrados en la Figura 3.40, el SMC sintonizado con Algoritmos Genéticos es el que presenta mejor desempeño, dado que posee los menores valores de ISE e IAE en ambas trayectorias.

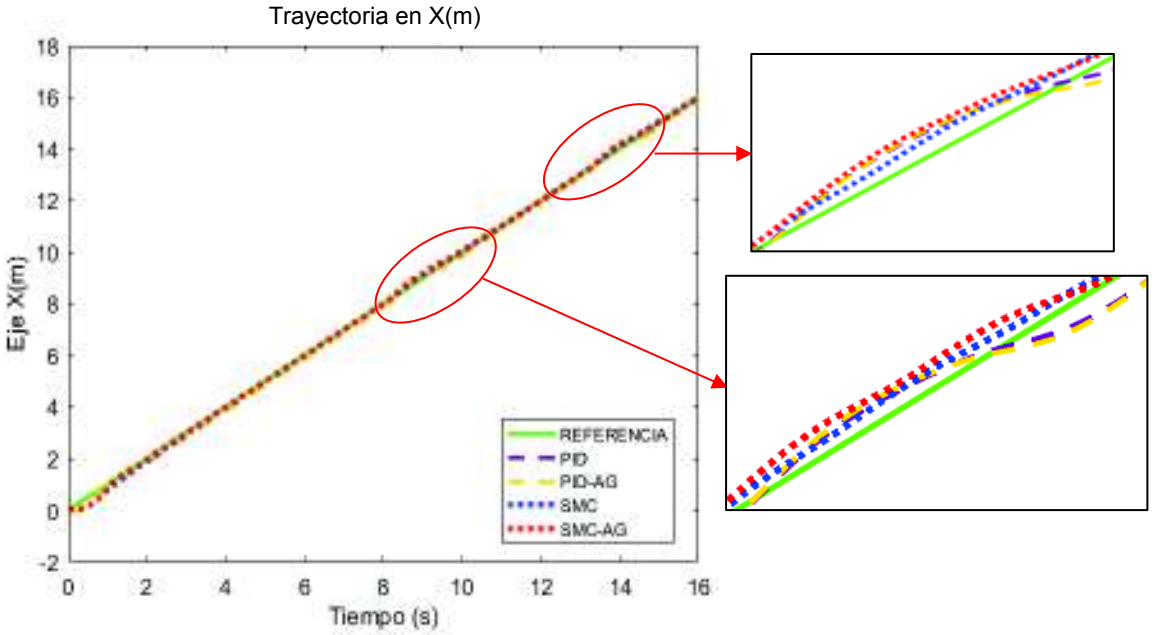


Figura 3.38. Trayectoria en X(m) respecto al Tiempo (s), para una trayectoria senoidal con perturbaciones.

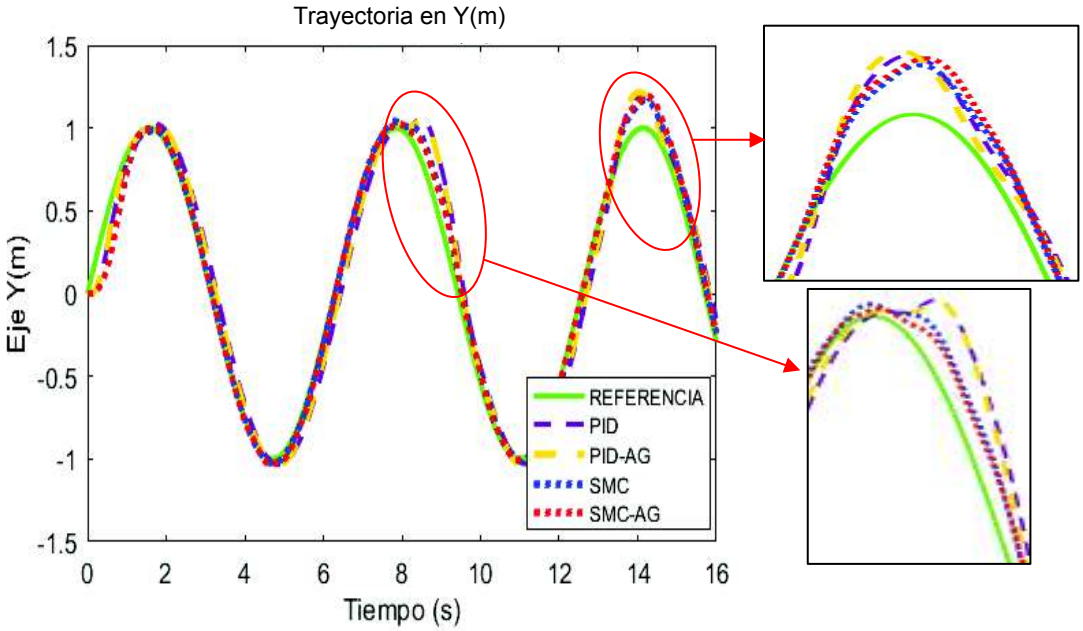


Figura 3.39. Trayectoria en Y (m) respecto al Tiempo, para una de trayectoria senoidal con perturbaciones.

En la Figura 3.40, se ilustra en un gráfico de columnas los valores obtenidos de los índices de desempeño para una trayectoria senoidal con perturbaciones.

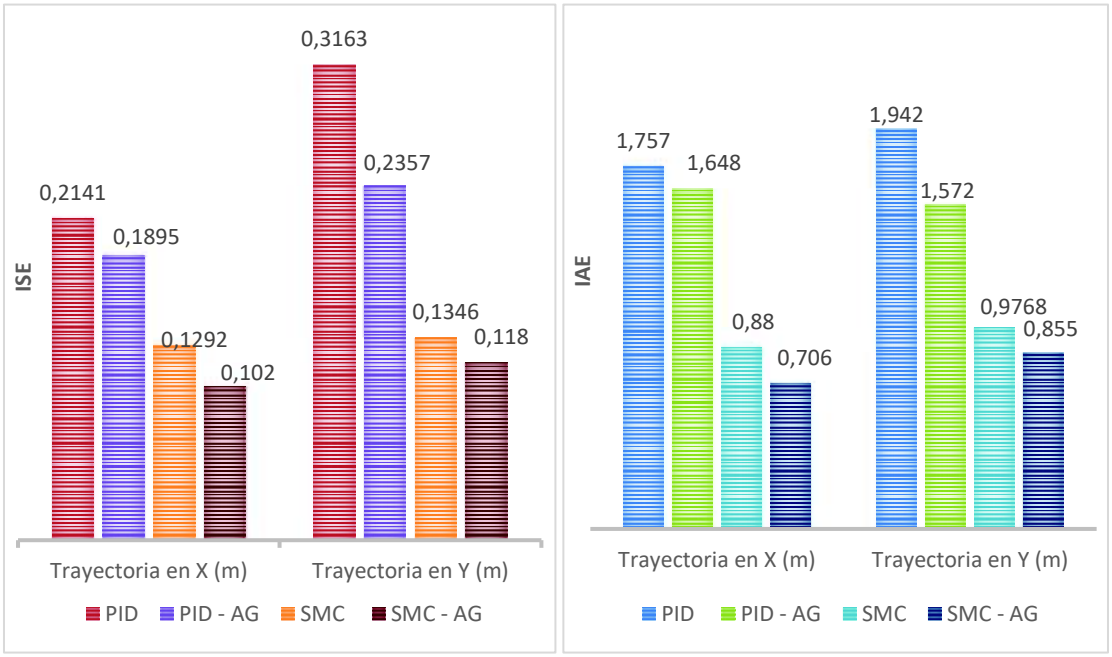


Figura 3.40. Comparación de los índices de desempeño obtenidos por el Seguimiento de una trayectoria senoidal con Perturbaciones.

Perturbaciones a una Trayectoria Cuadrada

Esta prueba, al igual que las anteriores, se realizó con dos perturbaciones de 0.1 m, equivalente al 42,7% de la longitud de los brazos del cuadricóptero, la primera a los 12 (s) y la segunda a los 26 (s), ambas con una duración de 1.5 (s)

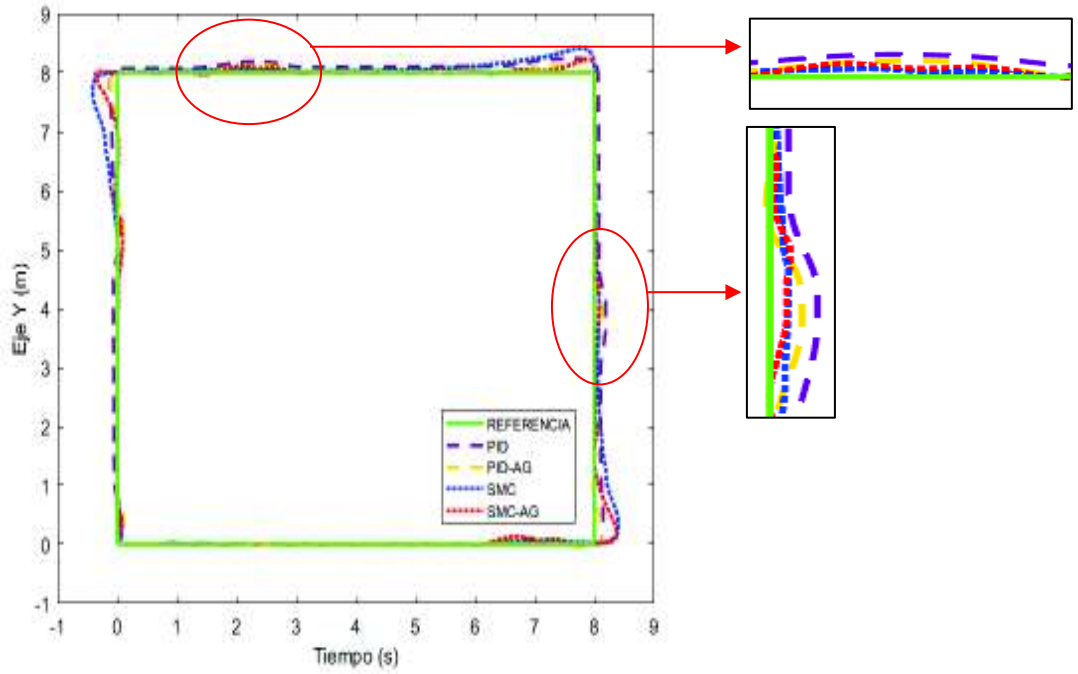


Figura 3.41 Seguimiento de Trayectoria Cuadrada con Perturbaciones.

Como se observa en la Figura 3.42 y Figura 3.43, ambos controladores en las diferentes sintonizaciones responden bien frente a perturbaciones, sin embargo, el controlador con mejor desempeño es el SMC-AG, de acuerdo a los valores de índices de desempeño obtenidos en la Figura 3.44.

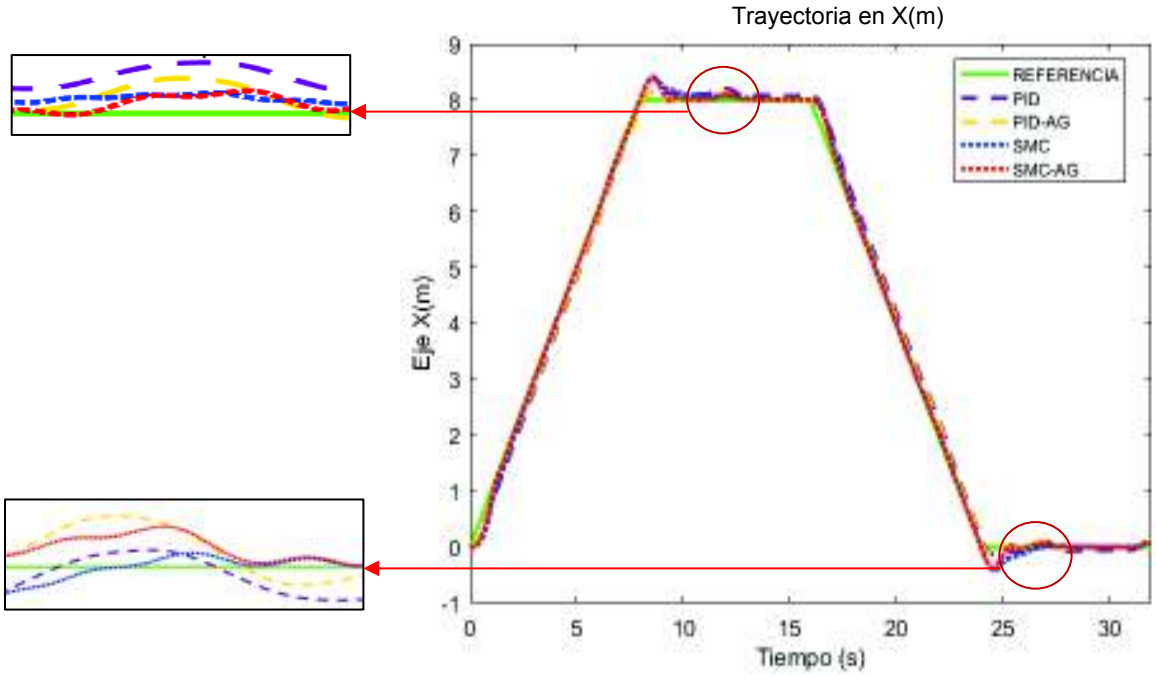


Figura 3.42. Trayectoria en X(m) respecto al Tiempo (s), para una trayectoria cuadrada con perturbaciones.

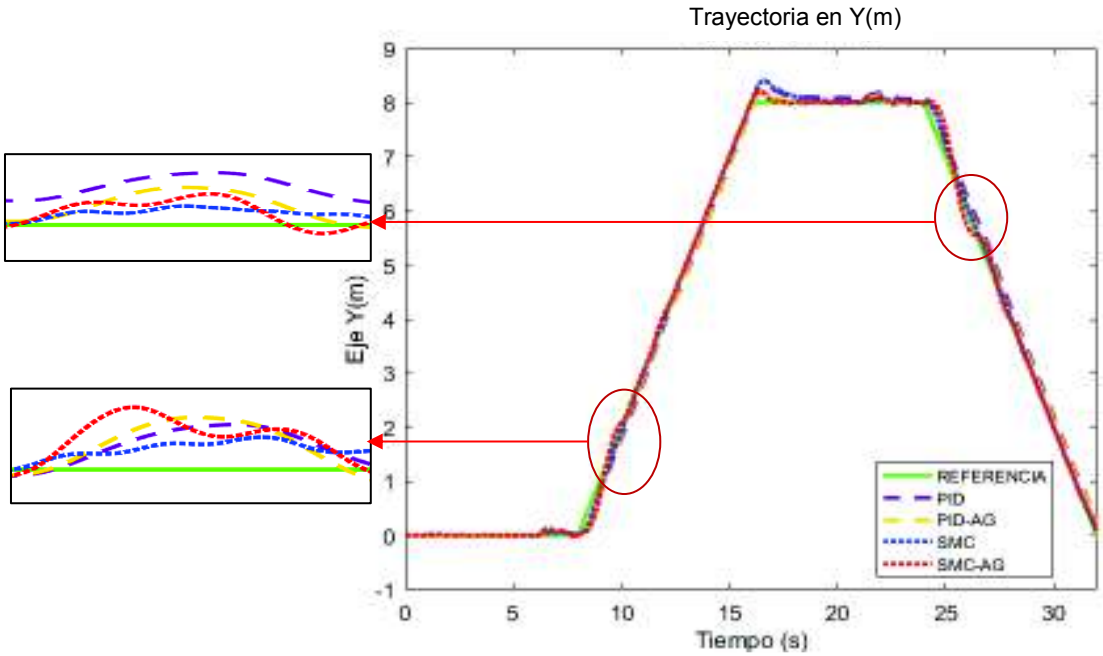


Figura 3.43 Trayectoria en Y (m) respecto al Tiempo, para una de trayectoria cuadrada con perturbaciones

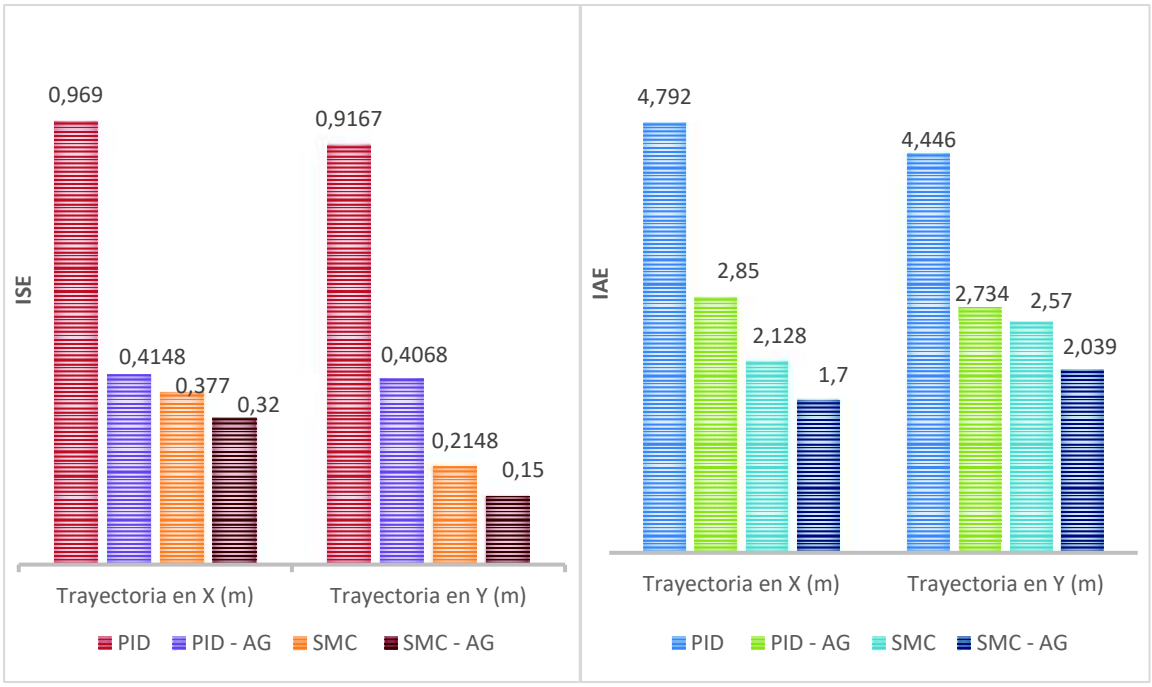


Figura 3.44. Comparación de los índices de desempeño obtenidos por el Seguimiento de una trayectoria cuadrada con Perturbaciones.

4. CONCLUSIONES

En esta sección se presentan las conclusiones obtenidas en el desarrollo del presente proyecto, así como las recomendaciones para la realización de este o futuros trabajos.

4.1 CONCLUSIONES

- Para el seguimiento de trayectoria del cuadricóptero se trabajó con el modelo dinámico del mismo, sin despreciar los efectos giroscópicos presentes en X y Y, el control del drone se presenta como un control en cascada de dos lazos, en el cual en el lazo interno se controla la altura mediante SMC, y en el externo se presenta dos opciones de control, PID y SMC.

- En el lazo externo de control se diseña un controlador PID y SMC para X y Y, el controlador en modos deslizantes se lo diseña en base a una superficie PD y la función de suavizamiento es una sigmoide. Para la sintonización de ambos controladores se observó el ISE obtenido convirtiéndose en una tarea de prueba y error.

- En el lazo externo de control ya sea mediante PID o SMC se ingresan valores de referencias en X y Y, de donde se obtienen las leyes de control Ux y Uy, se realiza una transformación de coordenadas de posición a rotación mediante el producto de las leyes de control y la matriz de rotación de Z respecto a yaw para poder tener las referencias de roll y pitch que ingresan al lazo de control interno.

- Para el desarrollo de los algoritmos genéticos, se determina parámetros iniciales como tamaño de población, porcentaje de cruce, porcentaje de mutación, límites entre los cuales se encuentran los parámetros a sintonizarse, establecer un amplio rango para estos valores no significa que se encuentren mejores soluciones, sólo se lograría un costo computacional adicional innecesario puesto que el AG converge a óptimos locales.

- Una de las partes más importantes del desarrollo del Algoritmo Genético es la selección de la función objetivo o función aptitud, puesto que de esto dependerá cuán buenos sean los parámetros que se desean optimizar, en este trabajo se ha utilizado como función objetivo el ISE en Z para la altura del cuadricóptero, y para el seguimiento de trayectoria la sumatoria de los ISE en X y Y, con la idea de dar igual peso a los errores existentes tanto en X como en Y.

- El esquema propuesto que implica la sumatoria de los ISE en X y Y, como función objetivo para la optimización y búsqueda de los mejores parámetros mediante algoritmos genéticos, permite encontrar soluciones que mejoran a ambos controladores (PID y SMC) de forma notoria.

- En todas las trayectorias implementadas al analizar los valores de ISE e IAE para las posiciones de X y Y, el IAE muestra valores más altos esto se debe a que analiza todos los errores a lo largo de la simulación, a diferencia del ISE que da mayor ponderación a los errores grandes que ocurren al inicio de la simulación.

- Al ser un control en cascada se sintonizó primero mediante AG la altura del cuadricóptero que se encuentra en el lazo interno controlada mediante SMC, donde su desempeño mejoró en aproximadamente 12%, posteriormente se sintonizaron los parámetros $K_p, K_i, K_d,$ para el PID y $\lambda_x, \lambda_x, K_{Dx}, K_{Dy},$ para el control en modos deslizantes, en donde su desempeño tomo ventaja en aproximadamente 30%.

- Para analizar la robustez del sistema se ha realizado pruebas en el cambio de masa del controlador, el cuál soporta hasta un 63.46%, equivalente a 0.35 Kg adicionales a la masa nominal (0.52 Kg), de igual forma se analizó el rechazo a perturbaciones en el control de altura y seguimiento de trayectoria, en ambos casos el sistema obtuvo una respuesta adecuada.

- Se aplicaron dos perturbaciones en forma de tren pulsos a cada trayectoria, con una amplitud de 10 cm, equivalentes al 42.5.% de la longitud de brazos del cuadricóptero, a las cuáles ambos controladores respondieron bien, sin embargo, el SMC sintonizado con algoritmos genéticos logró la estabilización en menor tiempo y mantuvo un error de posición menor.

- En los errores mostrados del seguimiento de trayectoria se observa unas pequeñas oscilaciones conocidas como "*chattering*", estas oscilaciones son características del control por modos deslizantes, además, dichas oscilaciones se encuentran por debajo de los 0.05 m, por lo que no se considera un problema para el elemento final de control, sin embargo, existen estudios dedicados a este problema donde se podría mejorarlo.

- El desarrollo de la herramienta computacional permite al usuario una manipulación sencilla para la configuración de los Algoritmos Genéticos, permite además trabajar con cuadricópteros que presenten otras características.

- El desarrollo de la interfaz gráfica permite realizar un análisis completo y sencillo del presente proyecto, gracias a que se visualizan los valores de los índices de desempeño obtenidos y sus porcentajes de mejora en cada posición.

4.2 RECOMENDACIONES

- Para la implementación del SMC en sistemas de control, se recomienda la elección adecuada de los parámetros de K_D y δ de la parte discontinua de la ley de control, pues esto asegurará la eliminación del chattering protegiendo al elemento final de control.

- Para trabajar con AG, es recomendable optimizar el código del modelo a ser ejecutado, en caso de trabajar con Matlab–Simulink, convertir todo el código a C para disminuir el costo computacional del AG.

- El método de sintonización de controladores basados en algoritmos genéticos propuesto en el presente trabajo presentó resultados satisfactorios, sin embargo, se recomienda a partir de los algoritmos genéticos desarrollar ecuaciones de sintonización que faciliten aún más el proceso de ajustar valores a los parámetros de los controladores.

5. REFERENCIAS BIBLIOGRÁFICAS

[1] E. García Martínez, “DESARROLLO DE UN CONTROLADOR PID INDUSTRIAL DE BAJO COSTE MEDIANTE RASPBERRY PI PARA CONTROL DE TEMPERATURA,” Escuela Técnica Superior Ingenieros Industriales Valencia, 2015.

[2] J. C. Terán and O. Camacho, “Genetics algorithms as a tuning tool for PID and sliding mode controllers,” *Rev. Técnica la Fac. Ing. Univ. del Zulia*, vol. 28, no. 3, pp. 10–16, 2005.

[3] Origen y desarrollo de los drones, “APLICACIONES Y OPERACIÓN CON DRONES/RPAS,” *Departamento de Comunicación*, 2015. [Online]. Available: http://drones.uv.es/origen-y-desarrollo-de-los-drones/. [Accessed: 08-Apr-2018].

[4] “Parrot AR.Drone 2.0 Power Edition | Sitio Web Official de Parrot,” 2018. [Online]. Available: https://www.parrot.com/es/drones/parrot-ardrone-20-power-edition#ar-drone-20-power-edition. [Accessed: 08-Apr-2018].

[5] T. Jiřinec, “Stabilization and control of unmanned quadcopter,” Czech Technical University in Prague, 2011.

[6] V. I. Espinosa Acosta and V. I. Moya González, “Diseño y simulación de un algoritmo de control robusto para formación cuadricópteros,” Quito, 2016., 2016.

[7] K. Ogata, E. Pinto Bermúdez, F. Matía, E. Pearson, and P. Hall, *Ingeniería de control moderna*, 5ta Edició. Madrid, 2010.

[8] K. Aström and T. Hägglund, *PID Controllers: Theory, Design and Tuning*, 2nd Editio. United States of America, 1994.

[9] K. J. Åström and T. Hägglund, *CONTROL PID AVANZADO*, Primera Edición. Madrid, España, 2009.

[10] O. Camacho and C. Smith, “Application of Sliding Mode Control to Nonlinear Chemical Processes with Variable Deadtime,” *2nd Congr. Colomb. Assoc. Autom.*, no. March 1997, pp. 122–128, 1997.

[11] L. J. Capito Ruiz and P. A. Proaño Chamorro, “Seguimiento de Trayectorias mediante cuatro Técnicas de Control utilizando una Plataforma Robótica Pioneer 3DX y el Sistema Operativo Robótico ROS,” Quito, 2015., 2015.

[12] F. Gordillo, “Estabilidad de Sistemas No Lineales Basada en la Teoría de Liapunov,” *Rev. Iberoam. Automática e Informática Ind. RIAI*, vol. 6, no. 2, pp. 5–16, 2009.

[13] H. J. Marquez, *Nonlinear Control Systems: Analysis and Design*, no. 4. United States of America, 2003.

[14] M. A. Yepez, “Funciones de Lyapunov y Algunas Aplicaciones,” Universidad Veracruzana, 2013.

[15] P. Samaniego and E. Vaca, “Diseño y Simulación de dos estrategias de control basado

en espacio nulo y tipo pid para una formación de cuadricópteros.," Escuela Politécnica Nacional, 2017.

[16] P. Ponce Cruz, *INTELIGENCIA ARTIFICIAL CON APLICACIONES A LA INGENIERÍA*, 1era ed. México, 2010.

[17] I. A. Ruge and M. A. Alvis, "Aplicación de los algoritmos genéticos para el diseño de un controlador PID adaptativo," *Tecnura*, vol. 13, no. 25, pp. 81–87, 2009.

[18] R. W. Anderson and M. Conrad, "Hans J. Bremermann: A pioneer in mathematical biology," *Biosystems*, vol. 34, no. 1–3, pp. 1–10, 1995.

[19] I. L.- Juarez and G. Lefranc, "Aplicación de Algoritmos Genéticos en la Navegación y Evasión de Obstáculos en un Robot Móvil SRV1 Aplicación de Algoritmos Genéticos en la Navegación y Evasión de Obstáculos en un Robot Móvil SRV1," no. October, 2014.

[20] C. Patiño, J. Molina, and C. Torres, "Algoritmo genético para la ubicación óptima de sensores en un robot seguidor de línea.," *Scientia*, no. 41, pp. 87–92, 2009.

[21] J. Arranz de la Peña and A. Parra, "ALGORITMOS GENÉTICOS," no. Algoritmos Genéticos, pp. 1–8.

[22] D. Medrano, "Application of Genetic Algoritms To Solve Warehouse Space Allocation Problem," University of Technology Sydney, 2015.

[23] R. L. Haupt and S. E. Haupt, *Algorithms Practical Genetic Algorithms*, Second Edi., no. June. New Jersey, 2004.

[24] P. Estevez, "Optimización Mediante Algoritmos Genéticos OPTIMIZACIÓN MEDIANTE ALGORITMOS GENÉTICOS," no. May, 2014.

[25] R. M. Solórzano, "Ecuaciones para la sintonización de controladores PID con acción derivativa aplicada a la señal realimentada," Universidad de Costa Rica, 2007.

[26] A. Emilov, "Diseño e Implementación de un Cuadricóptero Basado en el Microcontrolador STM32F4," Universidad Politécnica de Valencia, 2016.

[27] F. R. Guerrero Noboa and O. A. Menéndez Granizo, "Modelación, Simulación y Control de Sistemas Áereos No Tripulados utilizando Intelegencia Artificial," p. 166, 2013.

[28] M. Reinoso, "Diseño de un Sistema de Control por Régimen Deslizante para el Seguimiento de Trayectoria Lineal de Un Quadrotor," Universidad Politécnica Salesiana, 2014.

[29] M. Trejo Rodríguez, "Universidad Autónoma De Nuevo León," Universidad Autónoma de Nuevo León, 2014.

[30] L. M. Vargas Fonseca, "Desarrollo de algoritmos para el seguimiento de trayectorias de un quadrotor utilizando técnicas modernas de control con algebra lineal," Quito, 2015., 2015.

[31] Z. Benic, P. Piljek, and D. Kotarski, "Mathematical Modelling of Unmanned Aerial

Vehicles with Four Rotors," *Interdiscip. Descr. Complex Syst.*, vol. 14, no. 1, pp. 88–100, 2016.

[32] J. P. Ortiz Gonzales, "Modelado y control robusto de un vehículo aéreo no tripulado quadrotor, en ambientes cerrados," Universidad Politécnica Salesiana, 2014.

[33] G. V. Raffo, "Modelado y control de un helicóptero quadrotor," p. 115, 2007.

[34] A. M. Elhennawy and M. Habib, "Trajectory Tracking of a Quadcopter Flying Vehicle using Sliding Mode Control," pp. 6264–6269, 2017.

[35] Z. He and L. Zhao, "Quadrotor Trajectory Tracking Based on Quasi-LPV System and Internal Model Control," *Math. Probl. Eng.*, vol. 2015, pp. 1–13, 2015.

[36] M. Herrera, A. P. Gomez, W. Chamorro, and O. Camacho, "Sliding Mode Control: An Approach to Control a Quadrotor," *2015 Asia-Pacific Conf. Comput. Aided Syst. Eng.*, no. July, pp. 314–319, 2015.

[37] W. G. Aguilar, R. Costa-castelló, and C. Angulo, "Control Autónomo de Cuadricopteros para Seguimiento de Trayectorias," *IX Congr. Cienc. y Tecnol. ESPE 2014*, no. September, pp. 144–149, 2014.

[38] S. A. Galeas Hurtado, "Estudio y Aplicación de loa Algoritmos Genéticos para el Control," Escuela Politécnica Nacional, 2006.

[39] C. Pillajo, P. Bonilla, and R. Hincapié, "Algoritmo genético para sintonización de pid basado en la integral del error absoluto Genetic Algorithm for Tuning PID Integral Based on the absolute error," *N°*, vol. 13, pp. 1390–1869, 2016.

[40] MathWorks, "Create genetic algorithm options structure - MATLAB gaoptimset - MathWorks America Latina." [Online]. Available: https://la.mathworks.com/help/gads/gaoptimset.html. [Accessed: 03-Jun-2018].

[41] J. Franco and C. M. Vélez, "Criterios de robustez para sistemas lineales," pp. 1–3, 2010.

6. ANEXOS

ANEXO I. MANUAL DE USUARIO DE LA INTERFAZ GRÁFICA

El presente trabajo fue desarrollado en Matlab R2016a, juntamente con Simulink y GUIDE. La presente Interfaz desarrollada en GUIDE, permite manipular variables, visualizar valores de los índices de desempeño y comparar errores correspondientes a los resultados obtenidos por los controladores PID y SMC sintonizados heurísticamente y mediante Algoritmos Genéticos.

ARCHIVOS CONTENIDOS

Para que el usuario pueda interactuar con la Interfaz es necesario, que se cumplan los siguientes pasos:

Paso 1. Instalar el software MATLAB R2016a o una versión superior.

Paso 2. Abrir la Carpeta “Interfaz-Proyecto-de-Titulación”.

Figura I.1 Carpeta que contiene la interfaz gráfica

Paso 3. Ejecutar el archivo “INTERFAZ_TESIS.m”

Dentro de la carpeta “Interfaz-Proyecto-de-Titulación”, se encuentran archivos .fig, correspondientes a cada ventana de interfaz, con su respectivo .m, además encontraremos archivos .slx, desarrollados en Simulink los cuales serán llamados desde GUIDE para su respectiva ejecución. Al ejecutar el archivo “INTERFAZ_TESIS.m”, asegurarse que Matlab esté direccionado a la carpeta “Interfaz-Proyecto-de-Titulación”.

PRESENTACIÓN DE LA INTERFAZ

En la Figura I.2, se ilustra la portada del presente proyecto, la cual aparecerá inmediatamente después de ejecutar el archivo INTERFAZ_TESIS.m, para acceder al menú presionar el botón en verde y para salir de la interfaz pulsar el botón en rojo.

Figura I.2 Pantalla de presentación de la Interfaz Gráfica de Usuario

En la Figura I.3, se muestran las opciones de Menú, se presentan cuatro opciones de comparación, la primera de altura con SMC y SMC – AG, las restantes para el Análisis del Seguimiento de Trayectoria con PID y SMC con sus respectivas sintonizaciones.

Figura I.3 Pantalla de Menú de la Interfaz Gráfica de Usuario

En la Figura I.4, se realiza el análisis del Control de Orientación y Altura y se muestra la distribución de botones utilizados en esta interfaz, a continuación, se presenta la función de cada uno de ellos.

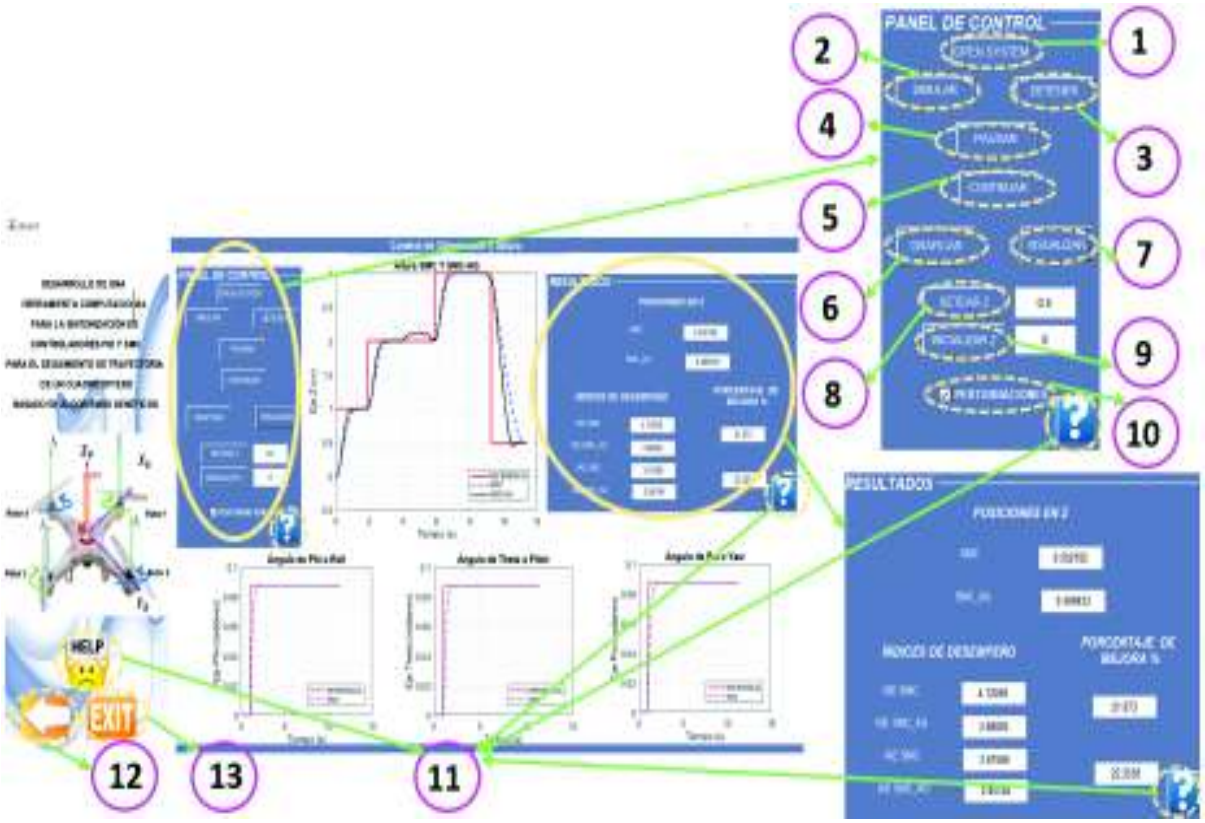


Figura I.4 Ventana asignada a la opción “ORIENTACIÓN Y ALTURA

- Botón 1 - **OPEN SYSTEM**: Permite abrir el archivo .slx que contiene el modelo para control de orientación y altura del cuadricóptero.
- Botón 2 – **SIMULAR:** Este botón ejecuta el archivo .slx
- Botón 3 – **DETENER**: Detiene y da por terminada la simulación el momento que el usuario desee y lo presione.
- Botón 4 – **PAUSAR:** Pausa la simulación al ser pulsado y la misma puede ser retomada nuevamente.
- Botón 5 – **CONTINUAR**: Si se desea continuar la simulación luego de haberla pausado se puede hacerlo al presionar este botón.
- Botón 6 – **GRAFICAR:** Muestra la gráfica obtenida de la altura contralada con SMC y SMC -AG, para su respectiva comparación.
- Botón 7 – **VISUALIZAR:** Al presionar este botón se muestran los índices de desempeño de la altura (ISE, IAE), así como sus porcentajes de mejora al comparar el SMC y SMC – AG.
- Botón 8 - **SETEAR Z**: Al presionar este botón se ingresa el valor de referencia de altura que se desee que el cuadricóptero alcance, este valor puede ser cambiado a lo

largo de la simulación, el tiempo de simulación en esta ventana es de inf, por lo que se debe presionar necesariamente el Botón 3, para dar por terminada la simulación.

- Botón 9 – **INICIALIZAR Z**: Este botón permite al usuario ingresar el valor inicial del que se desea que parta el cuadricóptero, por default este valor es de 0.
- Botón 10 - **PERTURBACIONES**: Este casillero permite elegir si se desean perturbaciones o no, el visto en el mismo indica que el sistema está trabajando con perturbaciones, y si está vació indica que no han sido consideradas las perturbaciones en la simulación.
- Botón 11 - **HELP**: Al presionar este botón se muestra un cuadro de ayuda, donde se informa al usuario que se va a realizar en la interfaz.

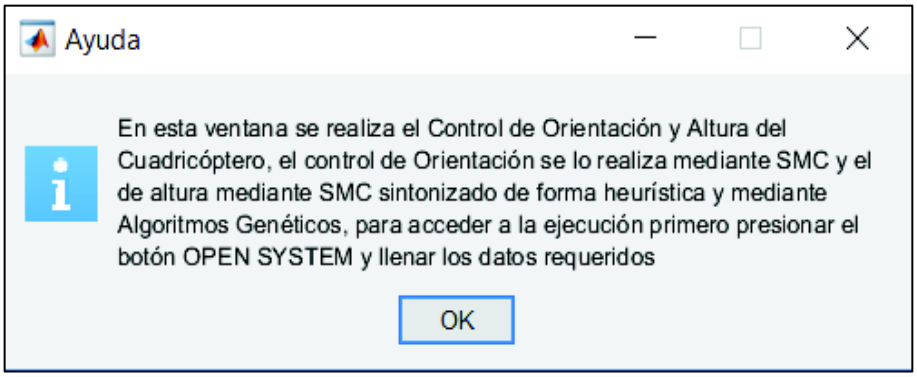


Figura I.5 Ventana de Ayuda que se presenta al presionar el botón HELP

Si se observa los íconos pequeños de ayuda, en cada uno de ellos hay información pertinente a cada etapa de trabajo

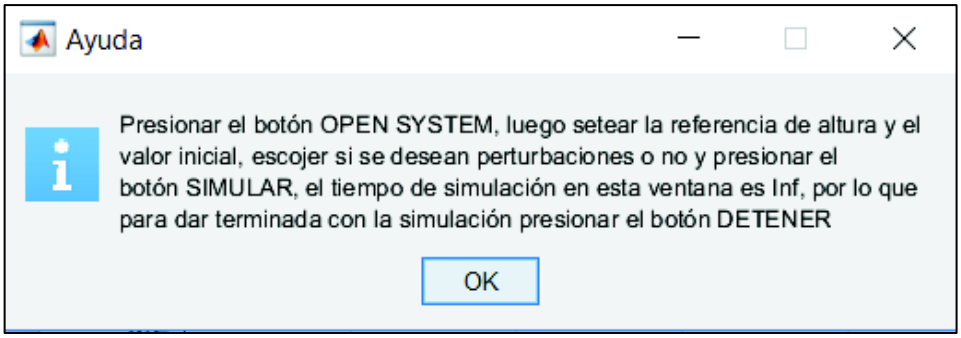


Figura I.6 Cuadro de ayuda para simulación del archivo

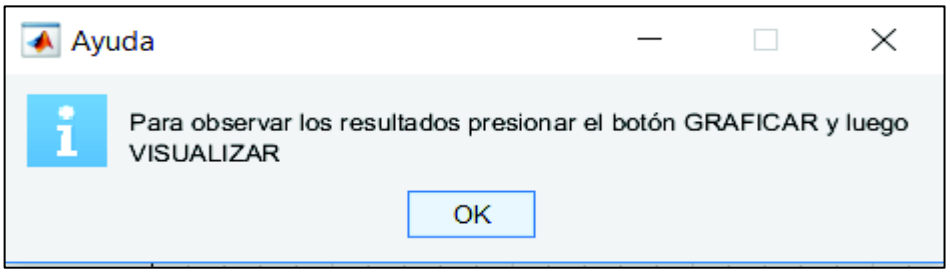


Figura I.7 Cuadro de ayuda para visualización de resultados

- Botón 12 – **REGRESAR**: Al presionar este botón que se muestra con la flecha hacia atrás, regresaremos a la ventana del menú de opciones.
- Botón 13 – **EXIT**: Al pulsar este botón el usuario ha elegido salir de la Interfaz Gráfica.

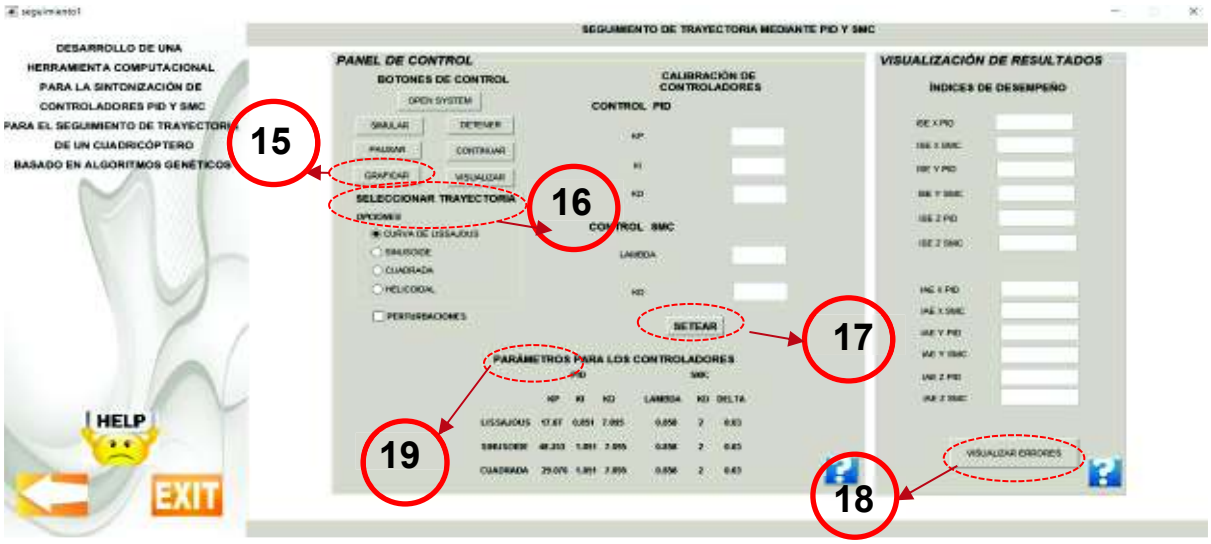


Figura I.8 Ventana de Interfaz para la opción “SEGUIMIENTO DE TRAYECTORIA MEDIANTE PID Y SMC”

- Botón 15 – **GRAFICAR**: Al presionar este botón se abre una ventana externa que permite al usuario visualizar el seguimiento de trayectoria en 3D, como se observa en la Figura I.9, este botón debe ser presionado luego de finalizar la simulación, dado que cada trayectoria tiene su propio tiempo de simulación.

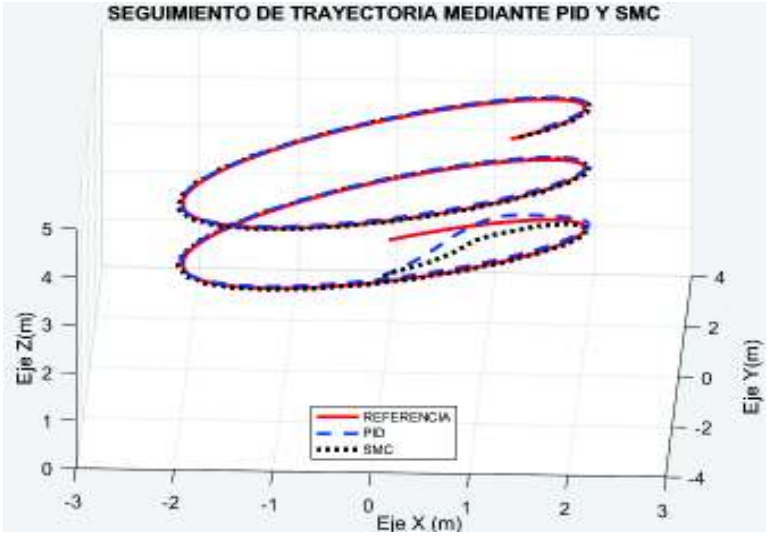


Figura I.9 Ventana externa para visualización de la trayectoria en 3D.

- Botón 16 - **SELECCIONAR TRAYECTORIA**: Esta sección cuenta con cuatro opciones, si se desea la trayectoria helicoidal se deben calibrar los controladores con los mismos parámetros que la trayectoria de Lissajous.
- Botón 17– **SETEAR**: En esta pantalla este botón es utilizado para la calibración de los controladores PID y SMC, cuyos parámetros se muestran en la sección **19- PARÁMEROS PARA LOS CONTROLADORES.**
- Botón 18 - **VISUALIZAR ERRORES**: Al presionar este botón se abre una nueva ventana, donde el usuario podrá visualizar las gráficas de posiciones y errores en cada uno de los ejes, ver

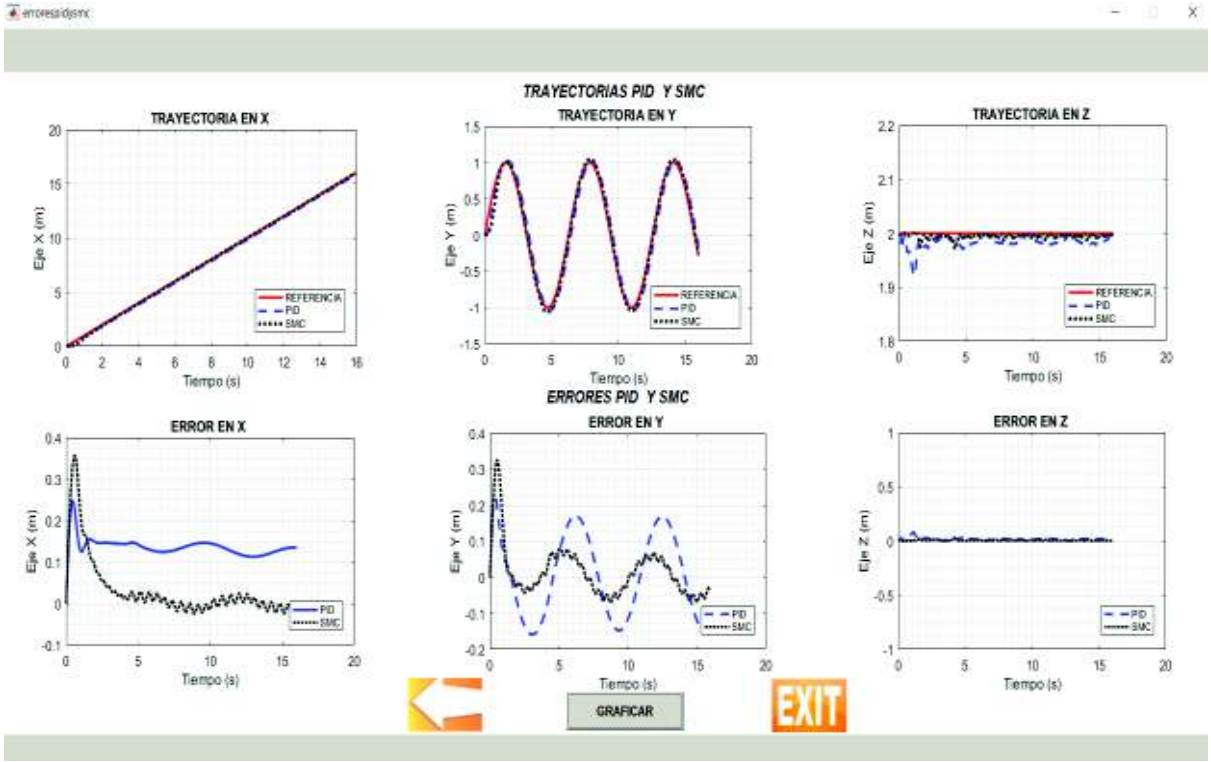


Figura I.10 Ventana para la visualización de las trayectorias y errores en cada eje.

En la Figura I.11 y Figura I.12, se muestran las ventanas para las opciones 3 y 4, ambas presentan las mismas opciones que la ventana mostrada para el seguimiento del PID y SMC, cada una de las ventanas de interfaz permite visualizar las gráficas de posiciones y errores en los tres ejes y trabajar con o sin perturbaciones.

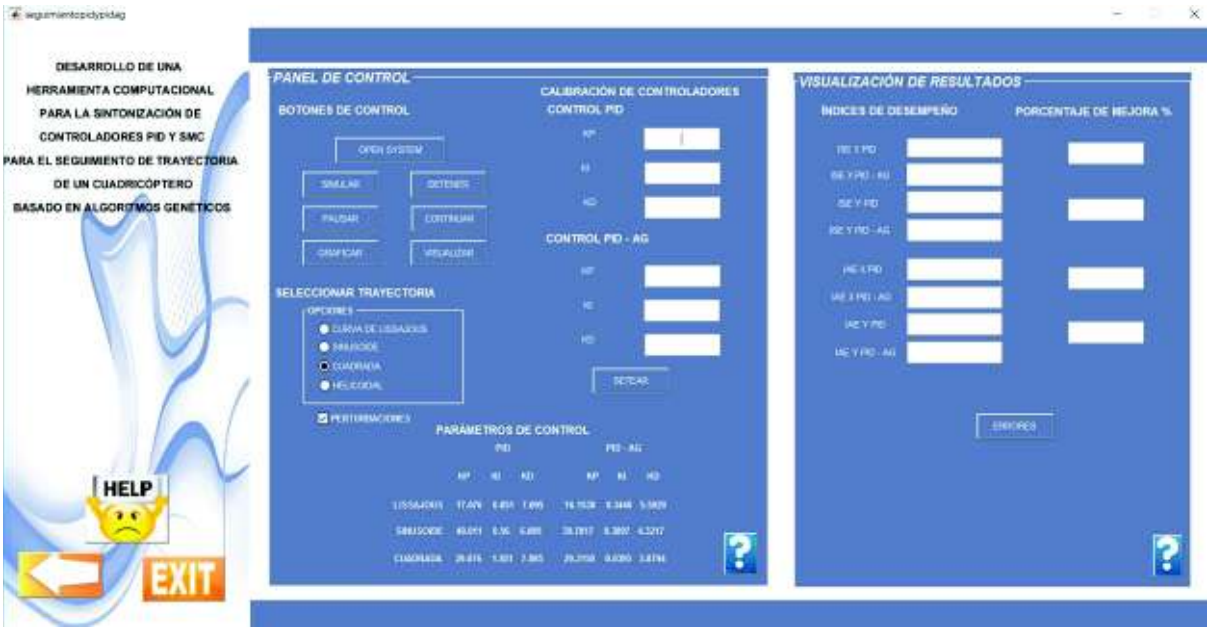


Figura I.11 Ventana de Interfaz Gráfica para la opción “SEGUIMIENTO DE TRAYECTORIA MEDIANTE PID Y PID -AG”

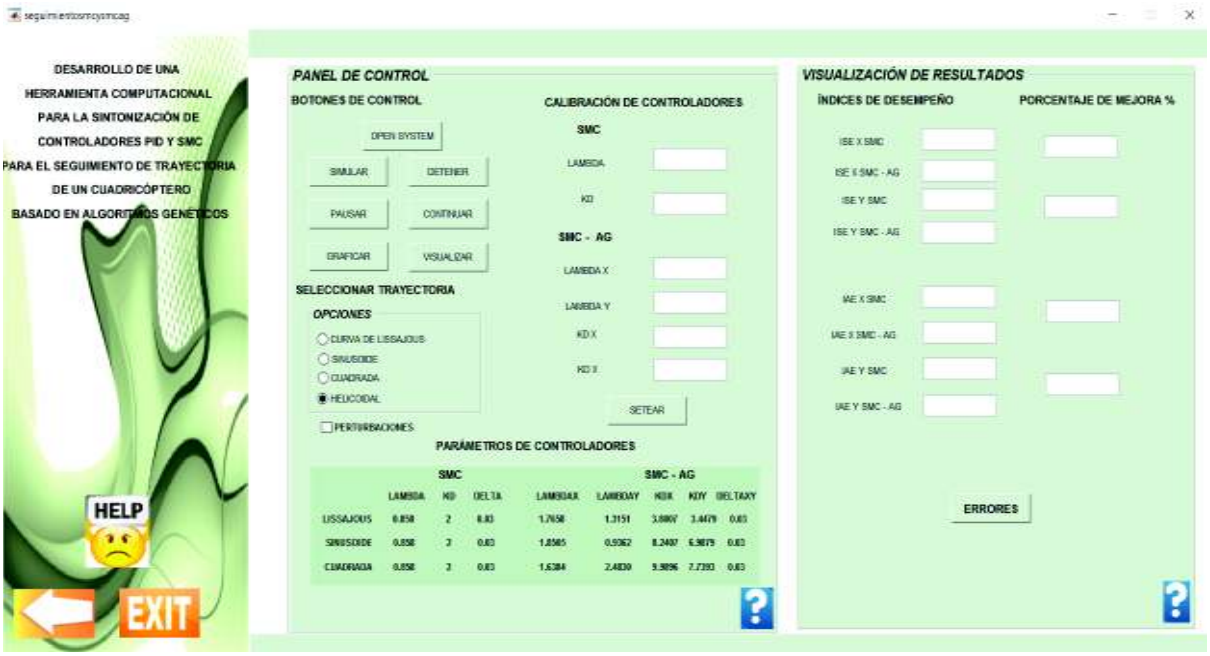


Figura I.12 Ventana de Interfaz Gráfica para la opción “SEGUIMIENTO DE TRAYECTORIA MEDIANTE SMC Y SMC – AG”

ANEXO II. SOLUCIONES OBTENIDAS MEDIANTE ALGORITMOS – GENÉTICOS

```
Optimization terminated: maximum number of generations exceeded.

x =

   16.1538    0.3440    5.5028


fval =

    4.1251
```

Figura II. 1 Parámetros Obtenidos Mediante AG Para Una Curva de Lissajous con PID

```
Optimization terminated: maximum number of generations exceeded.

x =

   39.7817    0.3897    4.3217


fval =

    1.1100
```

Figura II.2 Parámetros Obtenidos Mediante AG para una Sinusoide con PID

```
Optimization terminated: maximum number of generations exceeded.

x =

   29.2150    0.0395    3.8794


fval =

    1.1860
```

Figura II.3 Parámetros Obtenidos Mediante AG para una Trayectoria Cuadrada con PID

```
Optimization terminated: maximum number of generations exceeded.

x =

    1.7650    1.3151    3.8807    3.4479


fval =

    2.2671
```

Figura II.4 Parámetros Obtenidos Mediante AG para una Curva de Lissajous con SMC

```
Optimization terminated: maximum number of generations exceeded.

x =

    1.8505    0.9362    8.2407    6.9879


fval =

    0.1415
```

Figura II.5 Parámetros Obtenidos Mediante AG para una Trayectoria Senoidal con SMC

```
Optimization terminated: maximum number of generations exceeded.

x =

    1.6384    2.4830    9.9896    7.7393


fval =

    1.0702
```

Figura II.6 Parámetros Obtenidos Mediante AG para una Trayectoria Cuadrada con SMC

ANEXOIII. PARTE DE LA PROGRAMACIÓN DESARROLLADA EN SIMULINK

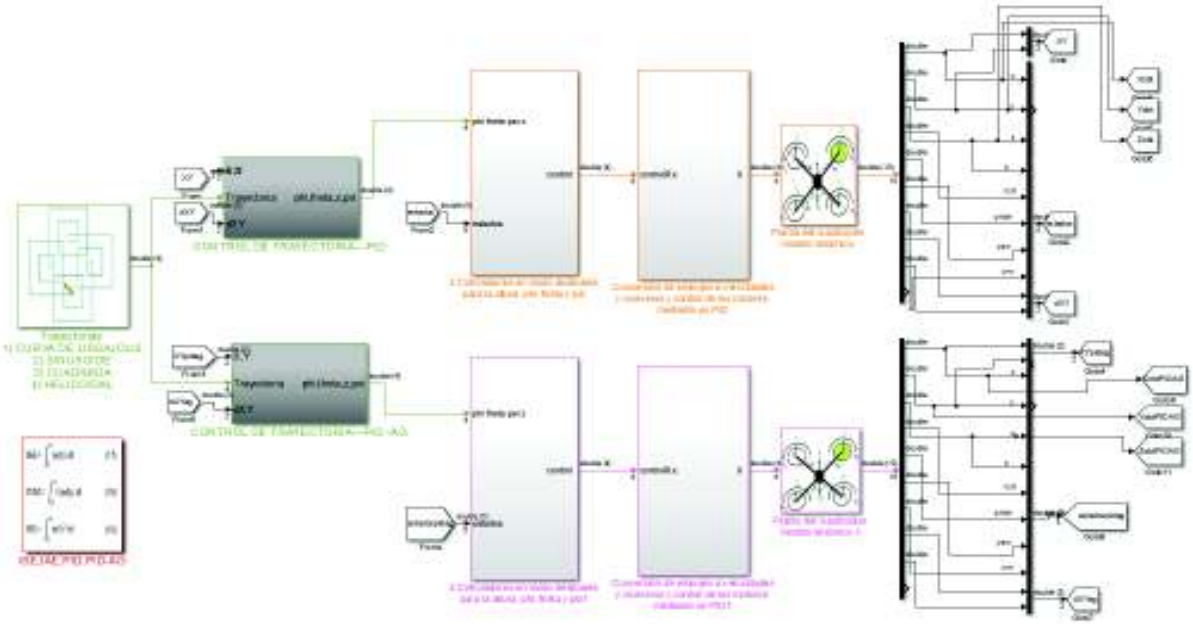


Figura III. 1 Programa en Simulink del Seguimiento de Trayectoria de un Cuadricóptero mediante PID y PID – AG.

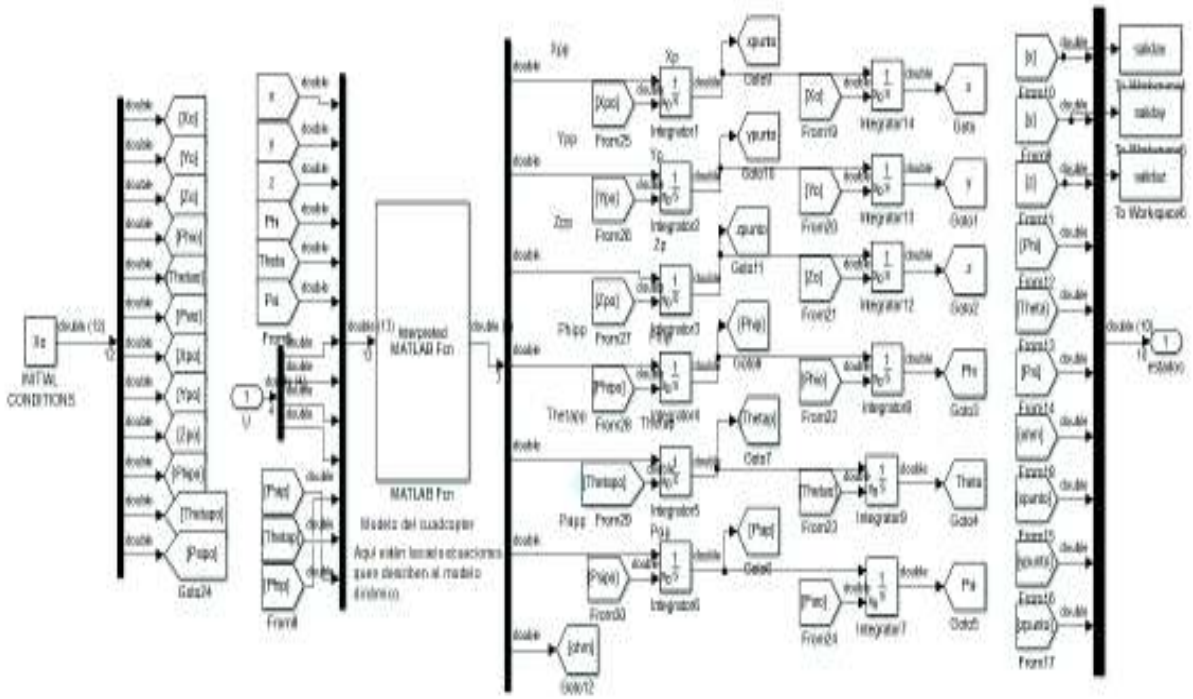


Figura III. 2 Etapa en Simulink del Modelado del Cuadricóptero y condiciones Iniciales

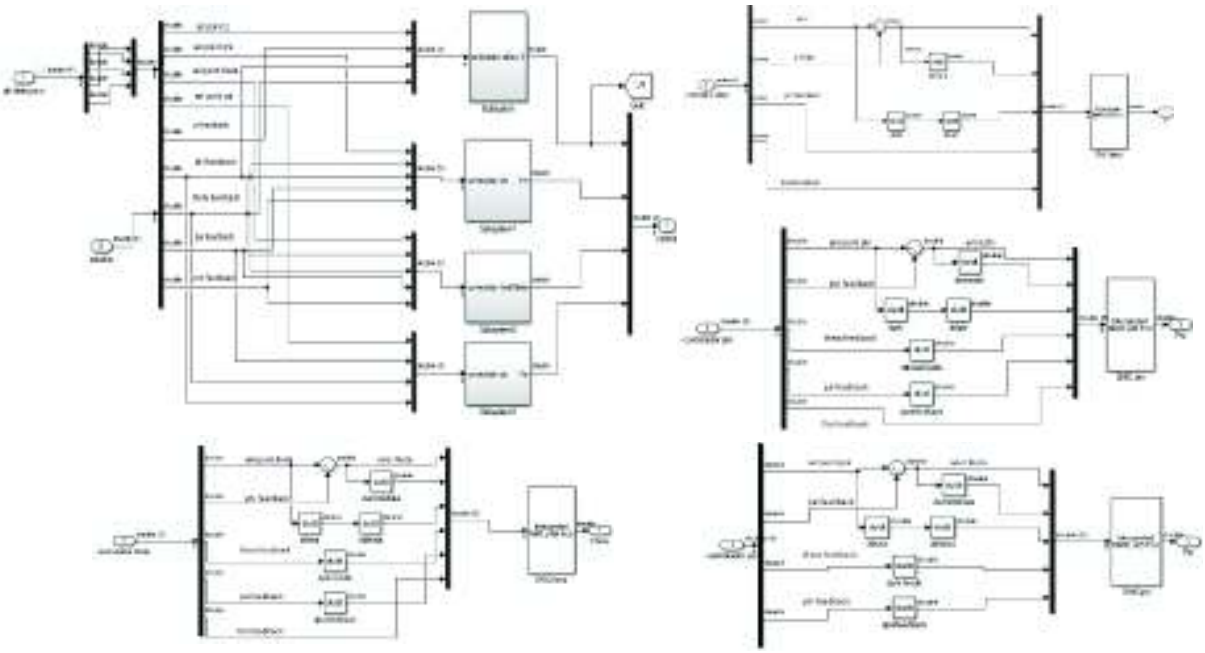


Figura III. 3 Control de Orientación y Altura del Cuadricóptero

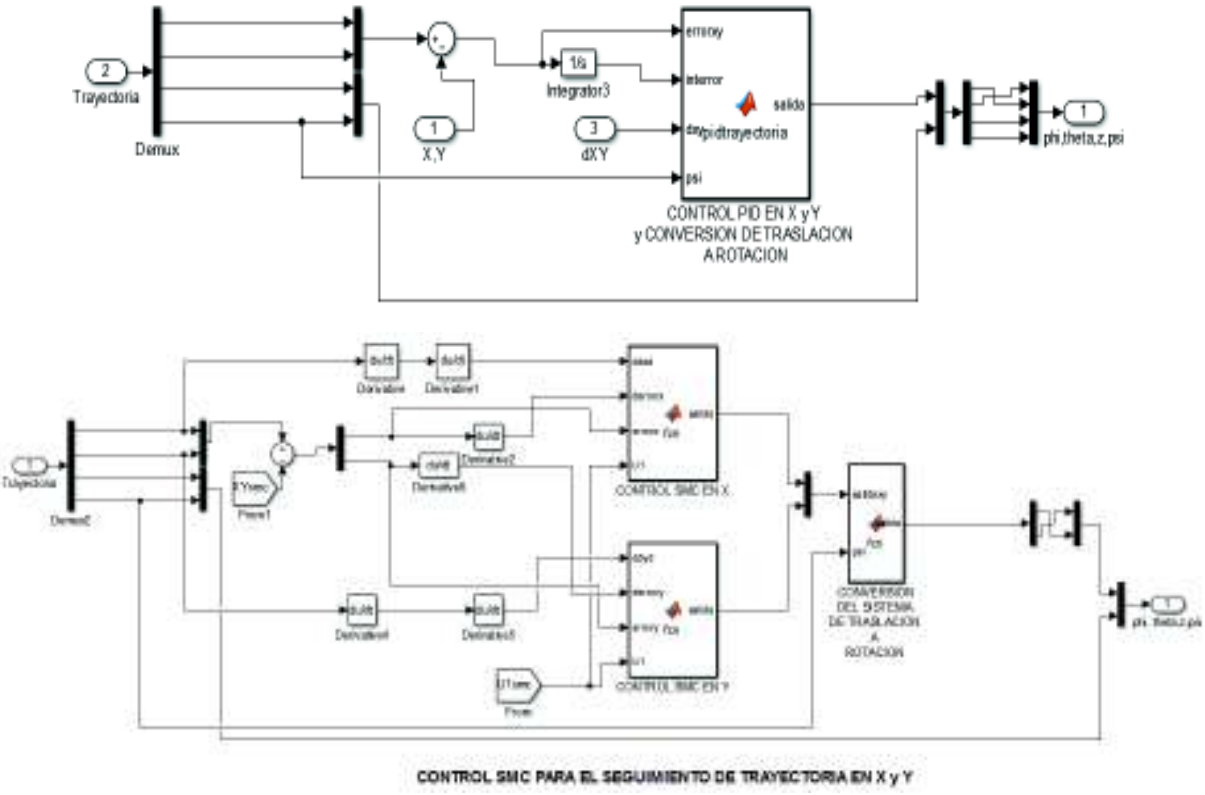


Figura III.4 Control del Seguimiento de Trayectoria de un Cuadricóptero mediante PID y SMC

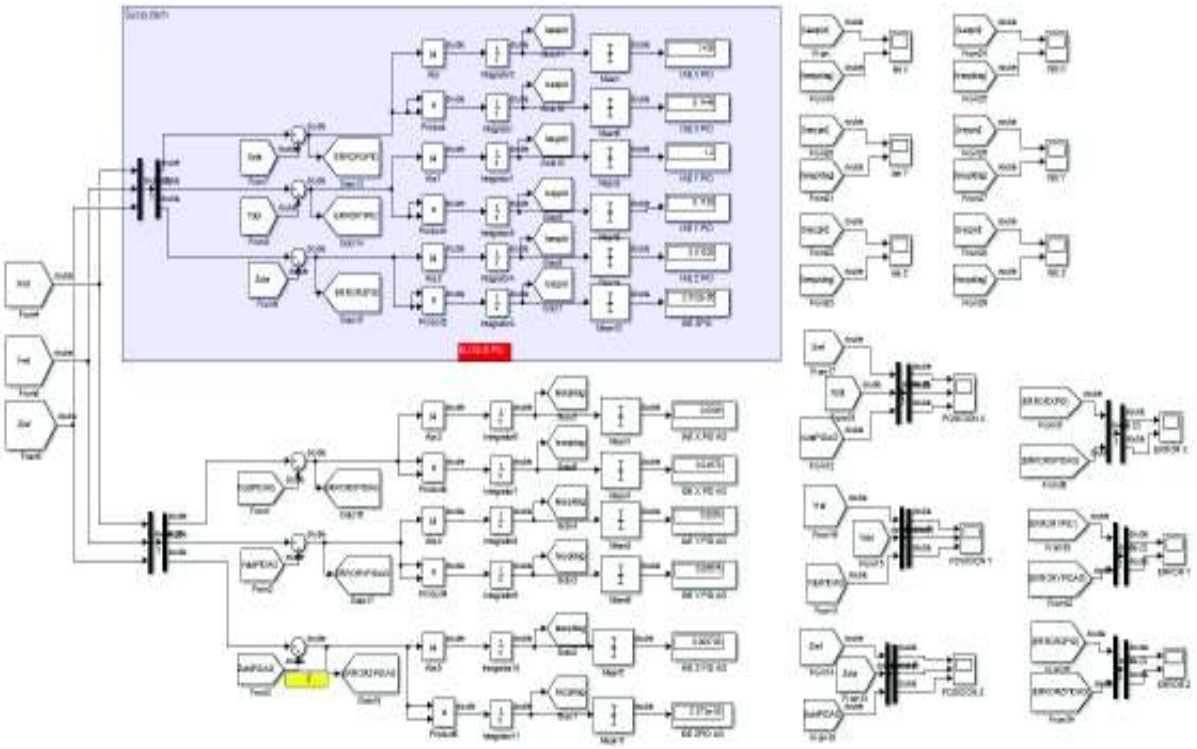


Figura III.5 Obtención de Gráficas, Errores y Valores Promedios de los Índices de Desempeño

ANEXO IV. AYUDA DE MATLAB PARA LA CREACIÓN DE UN AG

gaoptimset

Create genetic algorithm options structure

R2018a

Syntax

```
gaoptimset
options = gaoptimset
options = gaoptimset(@ga)
options = gaoptimset(@gamultiobj)
options = gaoptimset('param1',value1,'param2',value2,...)
options = gaoptimset(oldopts,'param1',value1,...)
options = gaoptimset(oldopts,newopts)
```

Description

`gaoptimset` with no input or output arguments displays a complete list of parameters with their valid values.

`options = gaoptimset` (with no input arguments) creates a structure called `options` that contains the options, or parameters, for the genetic algorithm and sets parameters to `[]`, indicating default values will be used.

`options = gaoptimset(@ga)` creates a structure called `options` that contains the default options for the genetic algorithm.

`options = gaoptimset(@gamultiobj)` creates a structure called `options` that contains the default options for `gamultiobj`.

`options = gaoptimset('param1',value1,'param2',value2,...)` creates a structure called `options` and sets the value of `'param1'` to `value1`, `'param2'` to `value2`, and so on. Any unspecified parameters are set to their default values. It is sufficient to type only enough leading characters to define the parameter name uniquely. Case is ignored for parameter names.

`options = gaoptimset(oldopts,'param1',value1,...)` creates a copy of `oldopts`, modifying the specified parameters with the specified values.

`options = gaoptimset(oldopts,newopts)` combines an existing options structure, `oldopts`, with a new options structure, `newopts`. Any parameters in `newopts` with nonempty values overwrite the corresponding old parameters in `oldopts`.

Options

The following table lists the options you can set with `gaoptimset`. See Genetic Algorithm Options for a complete description of these options and their values. Values in `[]` denote the default value. `[]'` means the default when there are linear constraints, and for `MutationFcn` also when there are bounds. You can also view the optimization parameters and defaults by typing `gaoptimset` at the command line. `I'` indicates that `ga` ignores or overwrites the option for mixed-integer optimization problems.

`optimoptions` hides the options listed in italics. But `gaoptimset` does not. See Options that optimoptions Hides.

Options for `ga`, integer `ga`, and `gamultiobj`

Option	Description	Values
`ConstraintTolerance`	Determines the feasibility with respect to nonlinear constraints. Also, `max(sort(c(:)),ConstraintTolerance)` determines feasibility with respect to linear constraints. For `gaoptimset`, use `TolCon`.	Positive scalar | `1e-3`
`CreationFcn`	I' Function that creates the initial population. Specify as a name of a built-in creation function or a function handle. See Population Options.	`'gacreationuniform'` | `{'gacreationlinearfeasible'}'` | Custom creation function
`CrossoverFcn`	I' Function that the algorithm uses to create crossover children. Specify as a name of a built-in crossover function or a function handle. See Crossover Options.	`'crossoverscattered'` for `ga`; `{'crossoverintermediate'}'` for `gamultiobj` | `'crossoverheuristic'` | `'crossoversinglepoint'` | `'crossovertwopoint'` | `'crossoverarithmetic'` | Custom crossover function
`CrossoverFraction`	The fraction of the population at the next generation, not including elite children, that the crossover function creates.	Positive scalar | `0.8`
`Display`	Level of display.	`'off'` | `'iter'` | `'diagnose'` | `'final'`
`DistanceMeasureFcn`	Function that computes distance measure of individuals. Specify as a name of a built-in distance measure function or a function handle. The value applies to decision variable or design space (genotype) or to function space (phenotype). The default `'distancecrowding'` is on function space (phenotype). For `gamultiobj` only. See Multiobjective Options. For `gaoptimset`, use a function handle, not a name.	`{'distancecrowding'}` means the same as `{@distancecrowding,'phenotype'}` | `{@distancecrowding,'genotype'}` | Custom distance function
`EliteCount`	**NM** Positive integer specifying how many individuals in the current generation are guaranteed to survive to the next generation. Not used in `gamultiobj`.	Positive integer | `ceil(0.05*PopulationSize)` | `(0.05*(default PopulationSize))` for mixed-integer problems
`FunctionTolerance`	The algorithm stops if the average relative change in the best fitness function value over `MaxStallGenerations` generations is less than or equal to `FunctionTolerance`. If `StallTest` is `'geometricweighted'`, then the algorithm stops if the weighted average relative change is less than or equal to `FunctionTolerance`. For `gamultiobj`, the algorithm stops when the geometric average of the relative change in value of the spread over `options.MaxStallGenerations` generations is less than `options.FunctionTolerance`, and the final spread is less than the mean spread over the past `options.MaxStallGenerations` generations. See gamultiobj Algorithm. For `gaoptimset`, use `TolFun`.	Positive scalar | `1e-6` for `ga`, `1e-4` for `gamultiobj`
`HybridFcn`	I' Function that continues the optimization after `ga` terminates. Specify as a name or a function handle. Alternatively, a cell array specifying the hybrid function and its options structure. See ga Hybrid Function. For `gamultiobj`, the only hybrid function is `@fgoalattain`. See gamultiobj Hybrid Function.	Function name or handle | `'fminsearch'` | `'patternsearch'` | `'fminunc'` | `'fmincon'` | `[]`[] or 1-by-2 cell array `{@solver, hybridoptions}`, where `solver` = `fminsearch`, `patternsearch`, `fminunc`, or `fmincon` | `[]`[]
`InitialPenalty`	**NM** I' Initial value of penalty parameter.	Positive scalar | `10`
`InitialPopulationMatrix`	Initial population used to seed the genetic algorithm. Has up to `PopulationSize` rows and `N` columns, where `N` is the number of variables. You can pass a partial population, meaning one with fewer than `PopulationSize` rows. In that case, the genetic algorithm uses `CreationFcn` to generate the remaining population members. See Population Options. For `gaoptimset`, use `InitialPopulation`.	Matrix | `[]`[]
`InitialPopulationRange`	Matrix or vector specifying the range of the individuals in the initial population. Applies to `gacreationuniform` creation function. `ga` shifts and scales the default initial range to match any finite bounds. For `gaoptimset`, use `PopInitRange`.	Matrix or vector | `[-10;10]` for unbounded components, `[-1e4+1;1e4+1]` for unbounded components of integer constrained problems, `[lb;ub]` for bounded components, with the default range modified to match one-sided bounds

MaxGenerations	Maximum number of iterations before the algorithm halts. For gaoptimset, use Generations.	Positive integer | [100*numberOfVariables] for ga. [200*numberOfVariables] for gamultiobj
MaxStallGenerations	The algorithm stops if the average relative change in the best fitness function value over MaxStallGenerations generations is less than or equal to FunctionTolerance. If StallTest is 'geometricWeighted', then the algorithm stops if the weighted average relative change is less than or equal to FunctionTolerance. For gamultiobj, the algorithm stops when the geometric average of the relative change in value of the spread over options.MaxStallGenerations generations is less than options.FunctionTolerance, and the final spread is less than the mean spread over the past options.MaxStallGenerations generations. See gamultiobj Algorithm. For gaoptimset, use StallGenLimit.	Positive integer | [50] for ga. [100] for gamultiobj
MaxStallTime	NN The algorithm stops if there is no improvement in the objective function for MaxStallTime seconds, as measured by tic and toc. For gaoptimset, use StallTimeLimit.	Positive scalar | [Inf]
MaxTime	The algorithm stops after running after MaxTime seconds, as measured by tic and toc. This limit is enforced after each iteration, so ga can exceed the limit when an iteration takes substantial time. For gaoptimset, use TimeLimit.	Positive scalar | [Inf]
MigrationDirection	Direction of migration. See Migration Options.	'both' | {'forward'}
MigrationFraction	Scalar from 0 through 1 specifying the fraction of individuals in each subpopulation that migrates to a different subpopulation. See Migration Options.	Scalar | [0.2]
MigrationInterval	Positive integer specifying the number of generations that take place between migrations of individuals between subpopulations. See Migration Options.	Positive integer | [20]

MutationFcn	F Function that produces mutation children. Specify as a name of a built-in mutation function or a function handle. See Mutation Options.	{'mutationgaussian'} for ga. {'mutationadaptfeasible'}* for gamultiobj | 'mutationuniform' | Custom mutation function
NonlinearConstraintAlgorithm	Nonlinear constraint algorithm. See Nonlinear Constraint Solver Algorithms. Option unchangeable for gamultiobj. For gaoptimset, use NonlinConAlgorithm.	{'auglag'} for ga. {'penalty'} for gamultiobj
OutputFcn	Functions that ga calls at each iteration. Specify as a function handle or a cell array of function handles. See Output Function Options. For gaoptimset, use OutputFcn.	Function handle or cell array of function handles | [[]]
ParetoFraction	Scalar from 0 through 1 specifying the fraction of individuals to keep on the first Pareto front while the solver selects individuals from higher fronts, for gamultiobj only.	Scalar | [0.35]
PenaltyFactor	NN If Penalty updata parameter.	Positive scalar | [100]
PlotFcn	Function that plots data computed by the algorithm. Specify as a name of a built-in plot function, a function handle, or a cell array of built-in names or function handles. See Plot Options. For gaoptimset, use PlotFcns.	ga or gamultiobj: {[]} | 'gaplotdistance' | 'gaplotgeneology' | 'gaplotselection' | 'gaplotscorediversity' | 'gaplotscores' | 'gaplotstopping' | 'gaplotmaxconstr' | Custom plot function ga only: 'gaplotbestf' | 'gaplotbestindiv' | 'gaplotexpectation' | 'gaplotrange' gamultiobj only: 'gaplotpareto' | 'gaplotparetodistance' | 'gaplotrankhist' | 'gaplotspread'
PlotInterval	Positive integer specifying the number of generations between consecutive calls to the plot functions.	Positive integer | [1]
PopulationSize	Size of the population.	Positive integer | [50] when numberOfVariables <= 5, [200] otherwise | [min(max(10*nvars,40),100)] for mixed-integer problems
PopulationType	Data type of the population. Must be 'doubleVector' for mixed-integer problems. ga ignores all constraints when PopulationType is set to 'bitString' or 'custom'. See Population Options.	'bitstring' | 'custom' | 'doubleVector'

SelectionFcn	F Function that selects parents of crossover and mutation children. Specify as a name of a built-in selection function or a function handle. gamultiobj uses only 'selectiontournament'.	{'selectionstochunif'} for ga. {'selectiontournament'} for gamultiobj | 'selectionremainder' | 'selectionuniform' | 'selectionroulette' | Custom selection function
StallTest	NN Stopping test type.	'geometricweighted' | {'averageChange'}
UseParallel	Compute fitness and nonlinear constraint functions in parallel. See Vectorize and Parallel Options (User Function Evaluation) and How to Use Parallel Processing.	true | {false}
UseVectorized	Specifies whether functions are vectorized. See Vectorize and Parallel Options (User Function Evaluation) and Vectorize the Fitness Function. For gaoptimset, use Vectorized with the values 'on' or 'off'.	true | {false}

See Also

ga | gamultiobj | gaoptimset

Topics

Genetic Algorithm Options

ORDEN DE EMPASTADO