



REPÚBLICA DEL ECUADOR

Escuela Politécnica Nacional

"E SCIENTIA HOMINIS SALUS"

La versión digital de esta tesis está protegida por la Ley de Derechos de Autor del Ecuador.

Los derechos de autor han sido entregados a la "ESCUELA POLITÉCNICA NACIONAL" bajo el libre consentimiento de los autores.

Al consultar esta tesis deberá acatar con las disposiciones de la Ley y las siguientes condiciones de uso:

- Cualquier uso que haga de estos documentos o imágenes deben ser sólo para efectos de investigación o estudio académico, y usted no puede ponerlos a disposición de otra persona.
- Usted deberá reconocer el derecho del autor a ser identificado y citado como el autor de esta tesis.
- No se podrá obtener ningún beneficio comercial y las obras derivadas tienen que estar bajo los mismos términos de licencia que el trabajo original.

El Libre Acceso a la información, promueve el reconocimiento de la originalidad de las ideas de los demás, respetando las normas de presentación y de citación de autores con el fin de no incurrir en actos ilegítimos de copiar y hacer pasar como propias las creaciones de terceras personas.

Respeto hacia sí mismo y hacia los demás.

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DISEÑO E IMPLEMENTACIÓN EN UNA TARJETA EMBEBIDA DE UN ALGORITMO DE CONTROL POR MODOS DESLIZANTES (SMC) PARA LOS ÁNGULOS DE ROLL, PITCH Y YAW DE UN HEXAROTOR DJI F550

**TESIS DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE MAGÍSTER EN
AUTOMATIZACIÓN Y CONTROL ELECTRÓNICO INDUSTRIAL**

**ALBUJA SÁNCHEZ BYRON MAURICIO - bry_alb_9002@yahoo.com
SÁNCHEZ GÓMEZ ELIZABETH VERÓNICA - ev.sanchez.gomez@gmail.com**

**DIRECTOR: DR. CHÁVEZ GARCÍA GEOVANNY DANILO -
danilo.chavez@epn.edu.ec**

**CODIRECTOR: DR. CAMACHO QUINTERO OSCAR EDUARDO -
oscar.camacho@epn.edu.ec**

Quito, Mayo 2018

AVAL

Certificamos que el presente trabajo fue desarrollado por Byron Mauricio Albuja Sánchez y Elizabeth Verónica Sánchez Gómez, bajo nuestra supervisión.

Dr. Geovanny Danilo Chávez
DIRECTOR DEL TRABAJO DE TITULACIÓN

Dr. Oscar Camacho
CODIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Nosotros, Byron Mauricio Albuja Sánchez y Elizabeth Verónica Sánchez Gómez, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros 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.

Ing. Byron Mauricio Albuja Sánchez

Ing. Elizabeth Verónica Sánchez Gómez

DEDICATORIA

Con todo cariño le doy gracias a Dios por permitirme cumplir mi crecimiento profesional, por darme a mi familia y amigos que son parte importante en mi vida especialmente a mi madre, a ellos les dedico cada página de este proyecto de grado, a mi padre que desde arriba me mira le agradezco y dedico este proyecto por cuidarme y estar a mi lado. También a mi novio Mauricio que ha sido una persona incondicional y ha estado a mi lado dándome su amor y apoyo le dedico este proyecto de grado.

Elizabeth Sánchez

Quiero dedicar este proyecto de grado especialmente a mis padres, familia y amigos por brindarme su apoyo durante todos mis años de vida enseñándome a crecer como persona. Se la dedico a mi madre por su sacrificio y dedicación y a mi padre por su compromiso y apoyo. A mi novia Elizabeth le dedico este proyecto por estar a mi lado brindándome su apoyo.

Byron Albuja

Este proyecto se lo dedicamos a nuestras familias, amigos y maestros por animarnos a seguir adelante con la realización de este proyecto.

Nosotros

AGRADECIMIENTO

De todo corazón agradezco a mi madre y familia por ser un soporte en mi vida profesional y permitirme lograr una parte de mis sueños, especialmente a mi madre por motivarme a seguir adelante cumpliendo mis metas y objetivos como estudiante. También agradezco la paciencia, comprensión y el sacrificio de su tiempo a Mauricio por estar siempre a mi lado apoyándome.

Elizabeth Sánchez

Quiero agradecer especialmente a mis padres por haberme guiado todo este tiempo brindándome su amor y comprensión, también a mi abuelito Aníbal por sus consejos y palabras de aliento y a toda mi familia por su confianza y apoyo. Un agradecimiento especial a Elizabeth por ser siempre un apoyo y una fuente de inspiración para seguir adelante.

Byron Albuja

Queremos dar gracias a nuestras familias por apoyarnos incondicionalmente en el desarrollo de este proyecto, a nuestro director y codirector del proyecto por guiarnos y estar pendientes durante la realización del mismo.

Gracias a nuestros amigos que nos han sabido dar ánimos y apoyo para continuar con la realización del proyecto.

Nosotros

ÍNDICE DE CONTENIDO

AVAL	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
RESUMEN	VIII
ABSTRACT	IX
1. INTRODUCCIÓN.....	1
1.1 OBJETIVO GENERAL	2
1.2 OBJETIVOS ESPECÍFICOS	3
1.3 ALCANCE	3
1.4 MARCO TEÓRICO.....	4
1.4.1 VEHÍCULOS AÉREOS NO TRIPULADOS - HEXAROTOR.....	4
1.4.1.1 Historia del UAV.....	4
1.4.1.2 Hexarotor	5
1.4.1.3 Modelo matemático del hexarotor.....	6
1.4.1.3.1 Ecuaciones de Movimiento	10
1.4.1.3.2 Fuerzas	10
1.4.1.3.3 Torques	11
1.4.2 COMPONENTES.....	14
1.4.2.1 Estructura del Hexarotor.....	15
1.4.2.2 Controlador Electrónico de Velocidad (ESC)	16
1.4.2.3 Motores <i>Brushless</i>	18
1.4.2.4 Hélices.....	18
1.4.2.5 Arduino DUE.....	19
1.4.2.5.1 Comparación Ardupilot Mega y tarjeta embebida Arduino DUE.....	21
1.4.2.6 Tarjeta IMU MPU6050	21
1.4.2.7 Batería LIPO 11.1V 9000mAh 3S1P CELL 60C-120C.....	22
1.4.2.8 Batería LIPO TURNIGY NANO-TECH 300mAh 2S 35~70C.....	23
1.4.2.9 Conversor USB A UART	24
1.4.2.10 Control de manejo GENIUS G-08XU	24
1.4.3 CONTROL POR MODOS DESLIZANTES (SMC)	25
1.4.3.1 Métodos de suavizamiento del <i>chattering</i>	26

1.4.3.2	Sintonización de controladores SMC	29
1.4.3.2.1	Por curva de reacción	29
1.4.3.2.2	Por algoritmos genéticos	31
1.4.3.2.3	Por metodología heurística	32
1.4.4	CONTROL PID.....	32
1.4.4.1	Control en cascada	33
1.4.4.2	Sintonización para el control PID	34
1.4.4.2.1	Zeigler y Nichols	34
1.4.4.2.2	Por curva de reacción	35
1.4.4.2.3	Por metodología heurística	35
1.4.4.2.4	Para PID en cascada	35
1.4.5	ESTIMACIÓN DE ORIENTACIÓN EN UAVs.....	36
1.4.5.1	Magnetómetro	36
1.4.5.2	Giroscopio.....	36
1.4.5.3	Acelerómetro	37
1.4.5.4	Unidad de medición inercial (IMU).....	37
1.4.5.5	Fusión de sensores	37
1.4.6	FILTRO DE KALMAN	39
2	METODOLOGÍA	41
2.1	CONTROLADORES DE ORIENTACIÓN	42
2.1.1	DISEÑO DE LOS CONTROLADORES SMC	42
2.1.2	CONTROLADORES PID.....	47
2.2	DISEÑO E IMPLEMENTACIÓN DEL SOFTWARE DE CONTROL.....	48
2.2.1	DECLARACIÓN DE LIBRERÍAS Y VARIABLES	48
2.2.2	FUNCIÓN VOID SETUP	50
2.2.3	FUNCIÓN <i>LOOP</i>	52
2.2.4	SOFTWARE EN EL COMPUTADOR.....	62
2.3	PROCEDIMIENTO DE SINTONIZACIÓN DE LOS CONTROLADORES SMC	67
2.3.1	SINTONIZACIÓN DE P_ϕ Y λ_1	70
2.3.2	SINTONIZACIÓN DE K_d Y δ	74
2.3.3	SINTONIZACIÓN DE λ_2	76
2.3.4	SINTONIZACIÓN DE $P_{\dot{\phi}}$	79
2.4	PROCEDIMIENTO DE SINTONIZACIÓN DE LOS CONTROLADORES PID	82

2.4.1	SINTONIZACIÓN DEL CONTROL EXTERNO.....	85
2.4.2	SINTONIZACIÓN DEL PARÁMETRO P DEL CONTROL INTERNO	85
2.4.3	SINTONIZACIÓN DEL PARÁMETRO I DEL CONTROL INTERNO	86
2.4.4	SINTONIZACIÓN DEL PARÁMETRO D DEL CONTROL INTERNO.....	88
2.5	CALIBRACIÓN DEL CONTROL DE MANEJO	89
3	RESULTADOS Y DISCUSIÓN	90
3.1	RESULTADOS.....	90
3.1.1	PRUEBAS REALIZADAS.....	90
3.1.1.1	<i>Offset</i> del giroscopio	90
3.1.2	RESULTADOS DEL CONTROL SMC	93
3.1.3	RESULTADOS DEL CONTROL PID.....	98
3.1.4	COMPARACIÓN ENTRE LOS CONTROLADORES PID Y SMC.....	103
3.1.4.1	Seguimiento de <i>setpoints</i>	103
3.1.4.2	Perturbaciones.....	106
3.2	DISCUSIÓN	107
4	CONCLUSIONES	110
5	REFERENCIAS BIBLIOGRÁFICAS	112
	ORDEN DE EMPASTADO.....	116

RESUMEN

Los vehículos aéreos no tripulados (UAVs) son aeronaves capaces de mantener de manera autónoma un vuelo controlado, tienen una gran variedad de formas, tamaños, configuraciones y características.

Los hexarotores son UAVs del tipo multirotor, sus características principales son: su despegue y aterrizaje se realizan de forma vertical y tienen la capacidad de quedarse suspendidos en una posición.

El hexarotor DJI F550 utiliza una tarjeta embebida Arduino DUE como control electrónico o cerebro del dron. La tarjeta está conectada con una unidad de medición inercial (IMU), cuyas mediciones pasan por filtro de alisado de Kalman.

Los datos filtrados se procesan en la tarjeta embebida para estimar los ángulos de orientación y posteriormente se realiza el control SMC de los ángulos de orientación (ángulos roll, pitch y yaw).

Para el movimiento del hexarotor se utiliza un control de manejo, el cual permite variar la altura y los ángulos de orientación del hexarotor con el fin de controlar su desplazamiento.

En el hexarotor se implementaron dos tipos de controladores; control SMC y PID para los ángulos de orientación. Se compara el funcionamiento de ambos controladores en condiciones similares para mediante el método de la integral del error cuadrático verificar cual es el mejor.

PALABRAS CLAVE: control por modos deslizantes, control SMC, hexarotor, control PID.

ABSTRACT

Unmanned aerial vehicles (UAVs) are aircrafts capable of maintaining a controlled flight in an autonomous way; they have a great variety of shapes, sizes, configurations and characteristics.

Hexarotors are UAVs within the type of multirotor, their main characteristics are: their takeoff and landing are performed vertically and they have the ability to stay suspended in one position.

The DJI F550 hexarotor uses an Arduino DUE embedded card as electronic controller or drone brain. The card is connected to an inertial measurement unit (IMU), whose measurements are passed through a smoothing Kalman filter.

The filtered data is processed at the embedded card to estimate the orientation angles and then the SMC control for the orientation angles (roll, pitch and yaw angles) is performed.

For the movement of the hexarotor a hand control is used, which allows to vary the height and the angles of orientation of the hexarotor in order to control its displacement.

In the hexarotor two types of controllers were implemented, the SMC and the PID controllers for the angles of orientation. The operation of these controllers in similar conditions is compared and then using the quadratic error integral method the best controller is shown.

KEYWORDS: sliding mode control, SMC control, hexarotor, PID control.

1. INTRODUCCIÓN

Un vehículo aéreo no tripulado, drone o UAV (*Unmanned Aerial Vehicle*) es una aeronave que vuela sin necesidad de una tripulación. Los UAVs son capaces de mantener de manera autónoma un vuelo controlado [1].

Los UAVs tienen una gran variedad de formas, tamaños, configuraciones y características. Históricamente surgen como aviones pilotados remotamente y en la actualidad pueden ser controlados desde una ubicación remota o ser de vuelo autónomo al seguir planes de vuelo pre programados [1].

Las aplicaciones de los UAVs abarcan varios campos como: uso bélico [2], patrullaje de fronteras [2], entrega de paquetes [3], seguridad y monitoreo [4], entre otras. Actualmente los UAVs cuentan con tarjetas dedicadas para su manejo y funcionamiento, los cuales usan control PID (Proporcional Integral Derivativo) [1], [5].

La dinámica de un hexarotor es multivariable, altamente no lineal y acoplada. Las técnicas de control comúnmente utilizadas como el control PID solo permiten un buen desempeño cuando la orientación del hexarotor no está muy alejada de la posición de suspensión en el aire sin desplazamiento, es decir que el valor de los ángulos de orientación del drone estén en cero [5].

Por este motivo, se implementó un control distinto al usado comúnmente en drones. Específicamente, se usó un control SMC (Control por modos deslizantes – *Sliding mode control*) para el control de orientación del hexarotor. El control SMC es un control robusto y no lineal que puede ser utilizado en sistemas lineales y no lineales, además presenta buen seguimiento de *setpoints*. El objetivo de este control es lograr que el proceso se deslice hacia el valor deseado a lo largo de una superficie deslizante. La ley de control consta de dos partes: fase de alcanzabilidad que es la parte discontinua responsable que la variable controlada alcance la superficie y fase deslizante que es la parte continua que mantiene la variable controlada sobre la superficie deslizante [5].

Un problema considerable en el control SMC es el fenómeno *chattering*, el cual se refiere a oscilaciones de amplitud y frecuencia finitas en la respuesta del sistema. Este fenómeno tiene lugar debido a que la fase de alcanzabilidad excita las dinámicas rápidas

no consideradas en el modelo del sistema y también debido a la discretización causada por implementaciones digitales. El problema del *chattering* se puede reducir usando métodos de suavizamiento [5].

En el presente proyecto se realiza la implementación de controladores SMC para los ángulos de orientación en un hexarotor DJI F550 utilizando una tarjeta embebida Arduino DUE como tarjeta de control, además se propone un procedimiento para sintonizar los parámetros de los controladores SMC.

A lo largo del documento se abordarán las diferentes fases del desarrollo del proyecto. En el primer capítulo se abordan los aspectos teóricos de los componentes y conceptos usados para la implementación del control SMC en los ángulos de orientación del hexarotor DJI F550, como métodos de suavizamiento que permiten disminuir el efecto de *chattering*, componentes del drone, filtro de Kalman que se utiliza en el procesamiento de datos del sensor a utilizar y el software desarrollado.

El segundo capítulo detalla: el diseño de los controladores SMC y PID, el software de control implementado en la tarjeta de control, el procedimiento propuesto para la sintonización de los controladores SMC, la sintonización de los controladores PID, el control de mano usado para el manejo del hexarotor y la calibración de las acciones de manejo.

En el tercer capítulo se muestran las pruebas realizadas, los resultados obtenidos con los controladores SMC y PID, la comparación de los controladores y una discusión sobre los resultados obtenidos.

Finalmente, en el cuarto capítulo se enuncian las conclusiones del trabajo realizado.

1.1 OBJETIVO GENERAL

Diseñar e implementar en una tarjeta embebida un algoritmo de control por modos deslizantes (SMC) para los ángulos de roll, pitch y yaw de un hexarotor DJI F550.

1.2 OBJETIVOS ESPECÍFICOS

- Realizar la revisión bibliográfica del modelo matemático del hexarotor y del control SMC en hexarotores.
- Diseñar las leyes de control SMC para los ángulos de orientación del hexarotor.
- Desarrollar el algoritmo de control SMC para los ángulos roll, pitch y yaw en una tarjeta embebida Arduino DUE.
- Proponer un procedimiento para sintonizar los parámetros de los controladores SMC de los ángulos de orientación roll, pitch y yaw.
- Comparar el control SMC con un control PID programado en la tarjeta embebida Arduino DUE

1.3 ALCANCE

El alcance del proyecto consiste en:

- Definir el modelo matemático a utilizar para el diseño de las leyes de control SMC.
- Diseñar las leyes de control SMC para los ángulos de orientación del hexarotor (roll, pitch y yaw).
- Implementar el control SMC para los ángulos de orientación del hexarotor en la tarjeta embebida Arduino DUE.
- Proponer un procedimiento para sintonizar los parámetros de los controladores SMC para los ángulos de orientación (roll, pitch y yaw) de un hexarotor.
- Lograr un error de seguimiento de *setpoints* menor a $\pm 10^\circ$ en los ángulos de roll, pitch y yaw.
- Implementar y sintonizar un control PID para los ángulos de orientación del hexarotor en la tarjeta embebida Arduino DUE.
- Utilizar el criterio de la integral del error cuadrático para comparar los resultados del control de orientación del control SMC con el control PID.

1.4 MARCO TEÓRICO

En esta sección se describen los aspectos teóricos concernientes al trabajo de titulación tales como: introducción al hexarotor, componentes utilizados en el hexarotor, aspectos teóricos del control SMC y métodos de sintonización, aspectos teóricos del control PID y métodos de sintonización, sensores para la estimación de orientación de un drone y los aspectos teóricos del filtro de Kalman.

Para la estimación de los ángulos de orientación se necesita un filtro en la medición del sensor para lo cual se utiliza un filtro alisado de Kalman. La fusión de sensores permite combinar las ventajas del acelerómetro y del giroscopio para obtener la estimación de los ángulos roll y pitch. El ángulo yaw solo se estima con el giroscopio debido a que con el acelerómetro no es posible estimarlo.

1.4.1 VEHÍCULOS AÉREOS NO TRIPULADOS - HEXAROTOR

1.4.1.1 Historia del UAV

La humanidad ha deseado poder conquistar el cielo, y una de tantas investigaciones se realizó durante la primera Guerra mundial, donde los militares estadounidenses comenzaron a desarrollar el avión no tripulado [1], [6]. En la segunda guerra mundial los vehículos no tripulados se usaron para reconocimiento e identificación de los bunkers alemanes [1].

Durante los años 80 y 90 se profundizó el desarrollo computacional y de sistemas de control electrónico donde los drones actuales fueron tomando forma. La tecnología UAV se fortalece como tecnológicamente fiable durante los años 80 [7].

En la actualidad los UAV se encuentran mayormente en el campo de la estrategia e información en conflictos bélicos, sin embargo, se puede ver drones de varios tamaños y formas debido a que sus características dependen de la aplicación [8].

1.4.1.2 Hexarotor

Un hexarotor es un tipo de UAV clasificado como multirotor. Un multirotor es una aeronave de ala rotatoria que posee cuatro o más rotores. Las principales características de esta aeronave son: su despegue y aterrizaje se realizan de forma vertical, tiene capacidad de quedarse suspendido en el aire sin desplazamiento (es decir que el valor de los ángulos de orientación del dron estén en cero), es bastante ágil para maniobrar y tiene una estructura simple [5].

La estructura de este tipo de UAV puede ser descrita como un cuerpo central que a su alrededor posee seis propulsores dispuestos de forma hexagonal, ver Figura 1.1. La dinámica de un hexarotor es altamente no lineal y acoplada, y es un sistema multivariable porque posee seis grados de libertad (tres grados traslacionales: X , Y y Z y tres grados de orientación: roll, pitch y yaw) [5].

El hexarotor consta de cuatro entradas de control que son empuje y los torques en roll, pitch y yaw; y posee seis salidas que son posición en x , y y z , y orientación en roll, pitch y yaw (ϕ, θ , y ψ).

La dificultad en el control de este tipo de sistemas radica en el acoplamiento no lineal existente entre los actuadores y los grados de libertad. El acoplamiento se refiere a que un cambio en una entrada de control afecta a varias salidas. La no linealidad se refiere a que las expresiones matemáticas que rigen como las variables de control afectan los grados de libertad del hexarotor (posición y orientación), no son lineales [5].

Las técnicas de control comúnmente utilizadas como el control PID solo permiten un buen desempeño cuando la orientación del hexarotor no está muy alejada de la posición de suspensión en el aire sin desplazamiento, esto debido a que es una técnica de control lineal y a que no es robusto ante las perturbaciones [5].

En la Figura 1.1 se muestran las diferentes configuraciones que podría tener el hexarotor:

- **Hexarotor I:** al poseer dos rotores más que un quadrotor en comparación con éstos, los hexarotores: permiten una mayor capacidad de carga y son más estables debido a que pueden aterrizar de forma segura si un propulsor falla. Por su estabilidad se los usa en aplicaciones de fotografía y video. En esta configuración el frente de la aeronave coincide con uno de los rotores.

- **Hexarotor V:** es similar al hexarotor I pero en esta configuración el frente de la aeronave está entre dos rotores.
- **Hexarotor IY:** este tipo de configuración permite obtener la misma capacidad de carga que un hexarotor I o V pero con un menor tamaño de la aeronave. Con esta configuración la maniobrabilidad se reduce en comparación con el hexarotor I o V. El frente de la aeronave coincide con uno de los rotores.
- **Hexarotor Y:** es similar a la configuración hexarotor IY con la diferencia de que el frente de la aeronave está entre dos rotores.

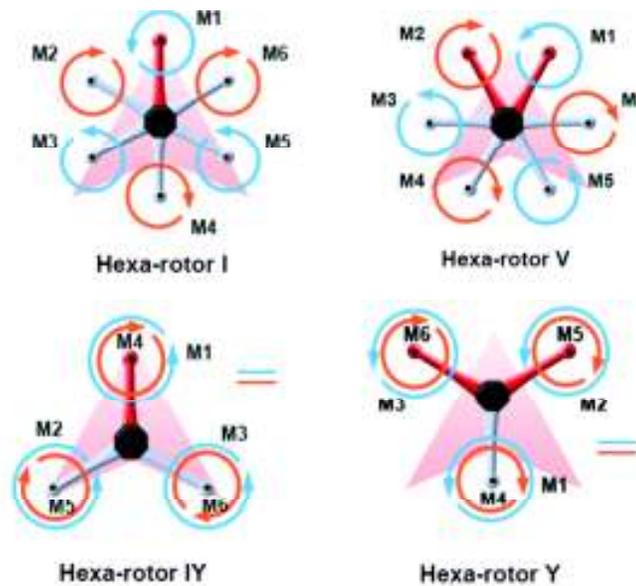


Figura 1.1. Tipo de configuraciones [9].

El tipo de configuración del hexarotor implementado es tipo V, en la Figura 1.2 se muestra la ubicación de los ejes de orientación del dron.

1.4.1.3 Modelo matemático del hexarotor

Se utilizó el modeló dinámico del hexarotor para el diseño de los controladores SMC. El modelo dinámico considera la evolución de la posición, velocidad y aceleración del hexarotor en respuesta a los torques sobre los ejes de rotación y al empuje generado por los motores. La posición de este tipo de UAV varía en función de la rotación de la estructura en torno a los siguientes tres ejes: rotación alrededor del eje x o roll, rotación alrededor del eje y o pitch y rotación alrededor del eje z o yaw (ver Figura 1.2) [5].

La posición de los ejes de roll, pitch y yaw sobre la estructura del hexarotor y el sentido de giro de las hélices se puede observar en la Figura 1.2. La rotación alrededor de cada uno de los ejes se logra mediante la variación de las velocidades de los propulsores de la siguiente manera [10]:

- Un giro positivo en roll (ϕ) requiere incrementar la velocidad de los motores 3 y 4 a la vez que se decrementan las velocidades de los motores 1 y 6. Para un giro negativo se invierte el procedimiento anterior.
- Un giro positivo en pitch (θ) requiere incrementar la velocidad de los motores 1, 2 y 3 a la vez que se decrementan las velocidades de los motores 4, 5 y 6. Para un giro negativo se invierte el procedimiento anterior.
- Un giro positivo en yaw (ψ) requiere incrementar la velocidad de los motores 2, 4 y 6 a la vez que se decrementan las velocidades de los motores 1, 3 y 5. Para un giro negativo se invierte el procedimiento anterior.

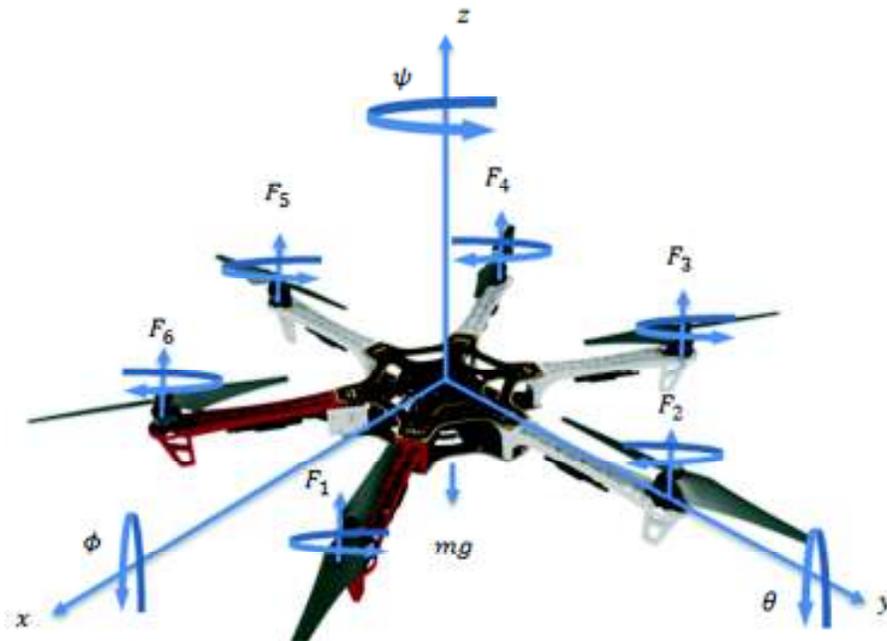


Figura 1.2. Ejes de rotación de un hexarotor [5].

Las expresiones matemáticas correspondientes al modelo dinámico que se muestran a continuación están basadas en las siguientes suposiciones [10]:

- El hexarotor es un cuerpo rígido.
- El dron tiene una estructura simétrica.
- Las tensiones en todas las direcciones son proporcionales al cuadrado de la velocidad de los propulsores.

En la Tabla 1.1 se encuentran las constantes y variables usadas en el modelo así como una breve descripción de las mismas [10], [11].

Tabla 1.1. Parámetros del modelo dinámico del hexarotor.

Parámetro	Valor	Descripción
m	2.247 [kg]	Masa del hexarotor.
b	2.986e-6 [$N \cdot s^2$]	Factor de empuje.
l	0.29 [m]	Longitud desde el centro de masa hasta el propulsor.
g	9.8 [m/s^2]	Gravedad.
J_r	3.357e-5 [$kg \cdot m^2$]	Inercia rotacional del propulsor.
I_x	0.0216 [$kg \cdot m^2$]	Inercia en el eje X.
I_y	0.0216 [$kg \cdot m^2$]	Inercia en el eje Y.
I_z	0.0432 [$kg \cdot m^2$]	Inercia en el eje Z.
$x \ y \ z$	- [m]	Posición relativa del hexarotor con respecto a un punto de referencia.
$\dot{x} \ \dot{y} \ \dot{z}$	- [m/s]	Velocidad de desplazamiento del hexarotor.
$\phi \ \theta \ \psi$	- [rad]	Ángulos de inclinación del hexarotor con respecto a los ejes X, Y y Z.
$\dot{\phi} \ \dot{\theta} \ \dot{\psi}$	- [rad/s]	Velocidad de giro del hexarotor con respecto a los ejes X, Y y Z.
U_1	- [N]	Empuje total de los rotores.
U_2	- [$N \cdot m$]	Torque en roll.
U_3	- [$N \cdot m$]	Torque en pitch.
U_4	- [$N \cdot m$]	Torque en yaw.
ω_i	- [rad/s]	Velocidad angular del rotor i , donde i toma valores del 1 al 6.
ω	- [rad/s]	Contra torque de yaw.
I	$\begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}$ [$N \cdot m \cdot s^2$]	Aproximación del tensor de inercia.
V	$\begin{bmatrix} u \\ v \\ w \end{bmatrix}$ [m/s]	Velocidad lineal en el sistema fijo del cuerpo.
w	$\begin{bmatrix} p \\ q \\ r \end{bmatrix}$ [rad/s]	Velocidad angular en el sistema fijo del cuerpo.

F	- [N]	Fuerzas que afectan al hexarotor en el marco fijo del cuerpo.
τ	- [N · m]	Torques que afectan al hexarotor en el marco fijo del cuerpo.
μ	- [kg/s]	Constante de la fuerza de arrastre.
$A_x A_y A_z$	- [m ²]	Área de la sección transversal del hexarotor.
ρ	- [kg/m ³]	Densidad del aire.
d	- [N · m · s]	Factor de arrastre.

Para describir el movimiento de un hexarotor se requieren dos sistemas de referencia: el sistema inercial de tierra R_T y el sistema fijo del cuerpo R_C . Ambos sistemas se relacionan entre sí por medio de una serie de rotaciones en el siguiente orden: rotación en roll (Ecuación 1.1), rotación en pitch (Ecuación 1.2) y finalmente rotación en yaw (Ecuación 1.3), en dónde $s = \text{seno}$ y $c = \text{coseno}$ [12].

$$R_C^T(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & s\phi \\ 0 & -s\phi & c\phi \end{bmatrix}$$

Ecuación 1.1. Matriz de rotación en roll.

$$R_C^T(\theta) = \begin{bmatrix} c\theta & 0 & -s\theta \\ 0 & 1 & 0 \\ s\theta & 0 & c\theta \end{bmatrix}$$

Ecuación 1.2. Matriz de rotación en pitch.

$$R_C^T(\psi) = \begin{bmatrix} c\psi & s\psi & 0 \\ -s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Ecuación 1.3. Matriz de rotación en yaw.

Al realizar las rotaciones en el orden antes mencionado se obtiene la matriz de rotación del cuerpo a tierra como se ve en Ecuación 1.4. Esta matriz es ortogonal lo cual significa que su inversa es igual a su transpuesta; por esta razón la transformación de la tierra al cuerpo se define con la Ecuación 1.5 [12].

$$R_C^T = R_C^T(\psi) \cdot R_C^T(\theta) \cdot R_C^T(\phi) = \begin{bmatrix} c\psi \cdot c\theta & c\psi \cdot s\theta \cdot s\phi - s\psi \cdot c\phi & c\psi \cdot s\theta \cdot c\phi + s\psi \cdot s\phi \\ s\psi \cdot c\theta & s\psi \cdot s\theta \cdot s\phi + c\psi \cdot c\theta & s\psi \cdot s\theta \cdot c\phi - c\psi \cdot s\phi \\ -s\theta & c\theta \cdot s\phi & c\theta \cdot c\phi \end{bmatrix}$$

Ecuación 1.4. Matriz de rotación de cuerpo a tierra.

$$(R_C^T)^{-1} = (R_C^T)^T = R_T^C$$

Ecuación 1.5. Matriz de rotación de tierra a cuerpo.

1.4.1.3.1 Ecuaciones de Movimiento

El modelo de Newton-Euler describe cómo afectan a las dinámicas de un cuerpo rígido las fuerzas y torques externos, como se ve en la Ecuación 1.6 [12].

$$\begin{bmatrix} m \cdot I_{3x3} & 0_{3x3} \\ 0_{3x3} & I_{3x3} \end{bmatrix} \cdot \begin{bmatrix} \dot{V} \\ \dot{W} \end{bmatrix} + \begin{bmatrix} W \times m \cdot V \\ W \times I \cdot W \end{bmatrix} = \begin{bmatrix} F \\ \tau \end{bmatrix}$$

Ecuación 1.6. Modelo de Newton - Euler

1.4.1.3.2 Fuerzas

La fuerza de gravedad en el sistema fijo del cuerpo está dada por la Ecuación 1.7 [12].

$$F_{gravedad} = R_T^C \cdot \begin{bmatrix} 0 \\ 0 \\ -m \cdot g \end{bmatrix} = \begin{bmatrix} m \cdot g \cdot s\theta \\ -m \cdot g \cdot c\theta \cdot s\phi \\ -m \cdot g \cdot c\theta \cdot c\phi \end{bmatrix}$$

Ecuación 1.7. Fuerza de gravedad.

La fuerza de empuje (ver Ecuación 1.8) es la potencia de levantamiento que hace volar al hexarotor y depende de la suma de las velocidades de las hélices. Para mantener el empuje siempre que se decrementa la velocidad de un motor se incrementa en la misma medida la velocidad de otro motor cuando se realizan maniobras de roll, pitch y yaw [12].

$$F_{empuje} = b \cdot (\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2 + \omega_5^2 + \omega_6^2)$$

Ecuación 1.8. Fuerza de Empuje.

Una fuerza de arrastre (ver Ecuación 1.9) aparece en el multirrotor cuando está volando y afecta a las aceleraciones lineales en el eje x y y [12].

$$F_{arrastrre} = \begin{bmatrix} -\mu \cdot u \\ -\mu \cdot v \\ 0 \end{bmatrix}$$

Ecuación 1.9. Fuerza de arrastre.

La fuerza producida por la resistencia del aire es proporcional al cuadrado de la velocidad de desplazamiento del multirotor y depende del tamaño y la forma del mismo. Esta fuerza está dado por la Ecuación 1.10 en donde: C es una constante adimensional [12].

$$F_{aire} = \begin{bmatrix} -0.5 \cdot C \cdot A_x \cdot \rho \cdot u \cdot |u| \\ -0.5 \cdot C \cdot A_y \cdot \rho \cdot v \cdot |v| \\ -0.5 \cdot C \cdot A_z \cdot \rho \cdot w \cdot |w| \end{bmatrix}$$

Ecuación 1.10. Fuerza de la resistencia del aire.

1.4.1.3.3 Torques

Los torques afectan la rotación total alrededor de un eje dependiendo de la distancia a la que se encuentre la fuerza del centro de gravedad del multirotor como se ve en la Figura 1.3. De acuerdo a este principio los torques generados por los motores en los ejes de rotación del hexarotor de roll, pitch y yaw están dados por Ecuación 1.11, Ecuación 1.12 y Ecuación 1.13 respectivamente. Un propulsor girando en el sentido de las manecillas del reloj produce un torque en sentido contrario a las manecillas del reloj en el eje de yaw y viceversa [12].

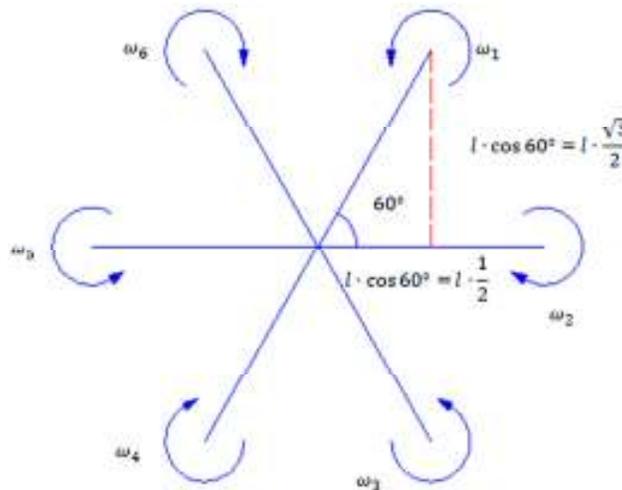


Figura 1.3. Distancias de los rotores al centro de gravedad.

$$\tau_\phi = b \cdot l \cdot (-0.5 \cdot \omega_1^2 - \omega_2^2 - 0.5 \cdot \omega_3^2 + 0.5 \cdot \omega_4^2 + \omega_5^2 + 0.5 \cdot \omega_6^2)$$

Ecuación 1.11. Torque en roll.

$$\tau_{\theta} = b \cdot l \cdot \frac{\sqrt{3}}{2} \cdot (-\omega_1^2 + \omega_3^2 + \omega_4^2 - \omega_6^2)$$

Ecuación 1.12. Torque en pitch.

$$\tau_{\psi} = d \cdot (-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2 - \omega_5^2 + \omega_6^2)$$

Ecuación 1.13. Torque en yaw.

El efecto giroscópico se produce debido a la rotación de los propulsores, este efecto produce un torque (Ecuación 1.14) en el sistema, el cual depende de la inercia rotacional de los propulsores y del contra torque del yaw. El contra torque del yaw está dado por la Ecuación 1.15 [12].

$$\tau_{giroscópico} = \begin{bmatrix} -J_r \cdot \dot{\theta} \cdot \omega \\ -J_r \cdot \dot{\phi} \cdot \omega \\ 0 \end{bmatrix}$$

Ecuación 1.14. Torque del efecto giroscópico.

$$\omega = -\omega_1 + \omega_2 - \omega_3 + \omega_4 - \omega_5 + \omega_6$$

Ecuación 1.15. Contra torque del yaw.

El contra torque del yaw produce un torque en el multirotor debido a la diferencia en la aceleración rotacional de los propulsores, este torque está dado por la Ecuación 1.16 [12].

$$\tau_{contra} = \begin{bmatrix} 0 \\ 0 \\ -J_r \cdot \omega \end{bmatrix}$$

Ecuación 1.16. Torque del contra torque del yaw.

Las velocidades de los rotores en función del empuje y torques se expresan como se muestra en la Ecuación 1.17 [10].

Las variables de empuje y de torques se expresan en función de la velocidad de los rotores como se muestra en la Ecuación 1.18 a la Ecuación 1.21 [10].

$$\begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \\ \omega_5^2 \\ \omega_6^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{6 \cdot b} & 0 & \frac{1}{3 \cdot b} & -\frac{1}{6 \cdot b} \\ \frac{1}{6 \cdot b} & \frac{\sqrt{3}}{6 \cdot b} & \frac{1}{6 \cdot b} & \frac{1}{6 \cdot b} \\ \frac{1}{6 \cdot b} & \frac{\sqrt{3}}{6 \cdot b} & \frac{1}{6 \cdot b} & \frac{1}{6 \cdot b} \\ \frac{1}{6 \cdot b} & 0 & -\frac{1}{6 \cdot b} & -\frac{1}{6 \cdot b} \\ \frac{1}{6 \cdot b} & 0 & -\frac{1}{6 \cdot b} & -\frac{1}{6 \cdot b} \\ \frac{1}{6 \cdot b} & \frac{\sqrt{3}}{6 \cdot b} & \frac{1}{6 \cdot b} & \frac{1}{6 \cdot b} \\ \frac{1}{6 \cdot b} & -\frac{\sqrt{3}}{6 \cdot b} & -\frac{1}{6 \cdot b} & -\frac{1}{6 \cdot b} \\ \frac{1}{6 \cdot b} & -\frac{\sqrt{3}}{6 \cdot b} & \frac{1}{6 \cdot b} & \frac{1}{6 \cdot b} \end{bmatrix} \cdot \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 2 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & 0 & -2 & 1 \\ 1 & -1 & -1 & -1 \\ 1 & -1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{6 \cdot b} \cdot U_1 \\ \frac{\sqrt{3}}{6 \cdot b} \cdot U_2 \\ \frac{1}{6 \cdot b} \cdot U_3 \\ \frac{1}{6 \cdot b} \cdot U_4 \end{bmatrix}$$

Ecuación 1.17. Velocidad de los rotores en función del empuje y torques requeridos.

$$U_1 = b \cdot (\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2 + \omega_5^2 + \omega_6^2)$$

Ecuación 1.18. Empuje generado por los motores.

$$U_2 = \frac{\sqrt{3}}{2} \cdot (\omega_2^2 + \omega_3^2 - \omega_5^2 - \omega_6^2)$$

Ecuación 1.19. Torque en roll generado por los motores.

$$U_3 = \frac{b}{2} \cdot (2 \cdot \omega_1^2 + \omega_2^2 - \omega_3^2 - 2 \cdot \omega_4^2 - \omega_5^2 + \omega_6^2)$$

Ecuación 1.20. Torque en pitch generado por los motores.

$$U_4 = b \cdot (-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2 - \omega_5^2 + \omega_6^2)$$

Ecuación 1.21. Torque en yaw generado por los motores.

Usando la Ecuación 1.1 a la Ecuación 1.21 se pueden obtener ecuaciones diferenciales de segundo orden que describan la posición y la orientación en el espacio del hexarotor, las cuales se muestran de la Ecuación 1.22 a la Ecuación 1.27. Para simplificar el modelo y el diseño de las ecuaciones del control SMC se eliminaron del modelo la fuerza de arrastre y la fuerza ocasionada por la resistencia del aire [5], [10].

$$\ddot{x} = (\cos \phi \cdot \sin \theta \cdot \cos \psi + \sin \phi \cdot \sin \psi) \cdot \frac{1}{m} \cdot U_1$$

Ecuación 1.22. Aceleración del hexarotor en la dirección x .

$$\ddot{y} = (\cos \phi \cdot \sin \theta \cdot \sin \psi - \sin \phi \cdot \cos \psi) \cdot \frac{1}{m} \cdot U_1$$

Ecuación 1.23. Aceleración del hexarotor en la dirección y .

$$\ddot{z} = -g + (\cos \phi \cdot \cos \theta) \cdot \frac{1}{m} \cdot U_1$$

Ecuación 1.24. Aceleración del hexarotor en la dirección z .

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

Ecuación 1.25. Aceleración del ángulo roll.

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

Ecuación 1.26. Aceleración del ángulo pitch.

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

Ecuación 1.27. Aceleración del ángulo yaw.

1.4.2 COMPONENTES

En la Figura 1.4 se muestra el diagrama de hardware del hexarotor, la parte de potencia es alimentada con la batería LiPo de 11.1 V, mientras que los dispositivos electrónicos de control son alimentados con la batería LiPo de 7.4 V. También se puede apreciar la comunicación con el computador y la obtención de datos a través del USB / UART. En esta sección se detalla los componentes de hardware presentes en el hexarotor.

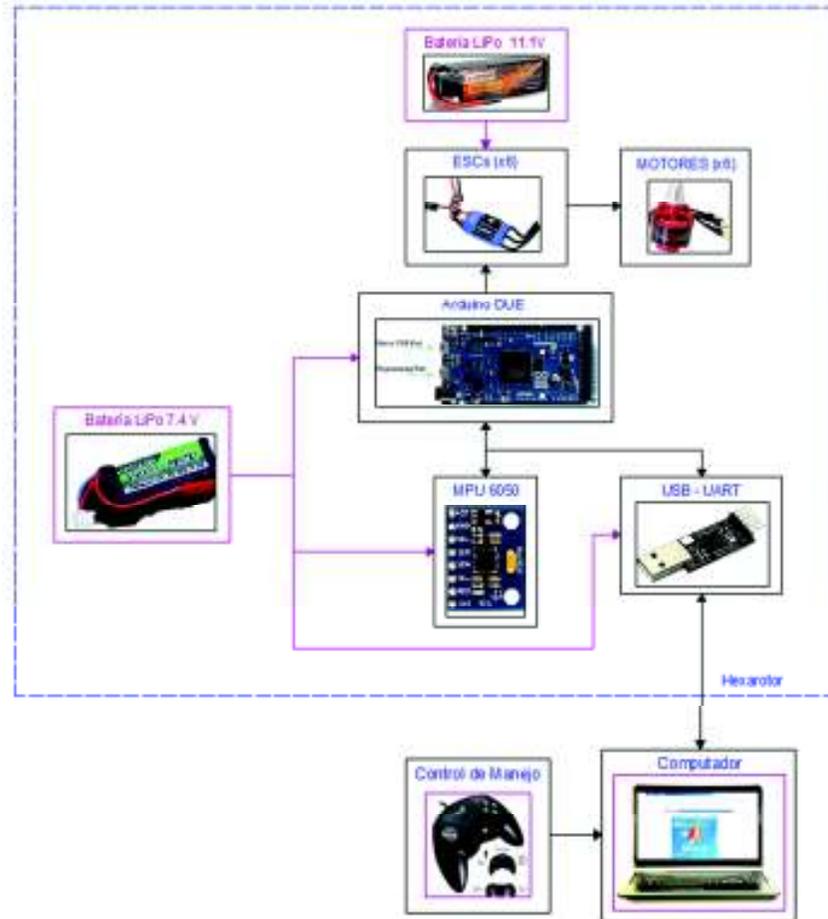


Figura 1.4. Diagrama de hardware.

1.4.2.1 Estructura del Hexarotor

El modelo utilizado para la estructura es el DJI F550 (ver Figura 1.5), el cual está elaborado con un material resistente y posee un circuito impreso de cobre para la alimentación de los motores. El peso aproximado de la estructura sin ningún componente adicional es de 400 [g] [11].



Figura 1.5. Estructura del hexarotor DJI F550.

1.4.2.2 Controlador Electrónico de Velocidad (ESC)

Los ESCs (ver Figura 1.6) son dispositivos que a su entrada reciben voltaje DC para a su salida entregar un voltaje AC trifásico. Para variar la velocidad de giro del motor el ESC varía la potencia entregada al motor, en función del ancho del pulso PWM enviado desde el controlador [11], [13].

El principio de control de velocidad de un ESC se basa en que la fuerza contra electro motriz es directamente proporcional a la velocidad del motor y que el torque producido es casi directamente proporcional a la corriente de fase. En la Figura 1.6 se muestra un esquema del control que lleva a cabo un ESC para variar la velocidad de un motor *brushless* [14].

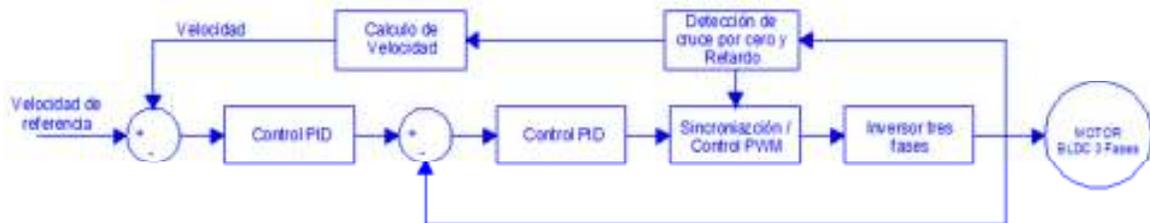


Figura 1.6. Esquema del control del ESC.

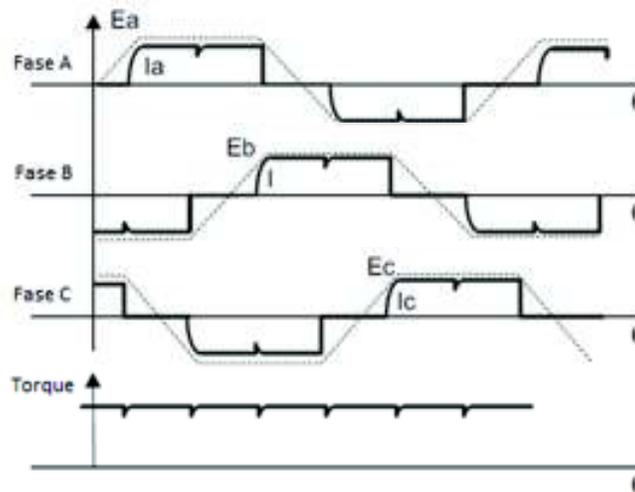


Figura 1.7. Torque resultante, voltajes y corrientes de fase.

El esquema de control de la Figura 1.6 permite que solo se deba controlar la corriente de una sola fase por vez, por lo que solo se requiere de un sensor de corriente. El controlador energiza solo la fase que produce mayor torque en cada instante; debido a

que la corriente no puede variar instantáneamente el torque presenta un pequeño rizado cada 60° de giro como se ve en la Figura 1.7 [14].

El inversor trifásico (ver Figura 1.8) no solo varía la potencia entregada al motor, también se encarga de sincronizar el flujo magnético del estator y del rotor y del control de la corriente de fase. El voltaje de alimentación DC es troceado por el inversor trifásico a una frecuencia fija. El resistor *shunt* se usa como un sensor de corriente, esta medición es filtrada para eliminar la frecuencia de troceado del voltaje DC [14].

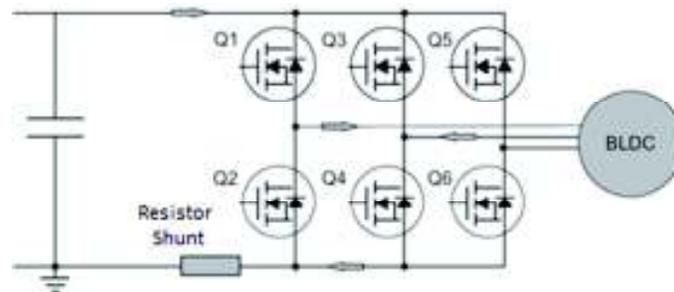


Figura 1.8. Inversor trifásico del ESC.

Para escoger un ESC se debe tener en cuenta el tipo de motor que van a manejar, el número de polos del motor, el consumo de corriente del motor y el tipo de batería a utilizar para alimentar los motores. El modelo de los ESC presentes en el hexarotor es el DYS Simok 30 A, estos tienen un peso aproximado de 180 [g] (ver Figura 1.9) [10].

Es importante configurar los ESCs antes de su uso. Se configuró para todos los ESC el pulso mínimo y máximo con el que van a trabajar para la variación de velocidad de los motores [10]. Todos los ESC fueron configurados para trabajar con un pulso de control de entre 600 y 2000 microsegundos.



Figura 1.9. ESC DYS Simonk 30 A [11].

1.4.2.3 Motores *Brushless*

Los motores *brushless* no tienen escobillas, el rotor cuenta con imanes permanentes y en el estator o carcasa se encuentra un bobinado trifásico. Por este motivo requieren de un variador electrónico (ESC) que entregue a cada bobinado el valor de voltaje necesario para producir el movimiento del rotor [10]. Considerar el consumo de corriente de los motores es importante, ya que de esto depende el dimensionamiento de los ESC y también de la batería a utilizar. Se debe verificar también el peso máximo que pueden levantar los motores con el tipo de hélices a utilizar y tener esto en cuenta al momento de montar todo lo necesario en el hexarotor.

Los motores brushless presentes en el hexarotor son DJI 2212 920 Kv (ver Figura 1.10), su peso es de aproximadamente 50 [g] y su constante de voltaje de 920 [rpm/V] [11], [13].



Figura 1.10. Motor DJI 2212 920 Kv *brushless* [11].

1.4.2.4 Hélices

Existen varios tipos de hélices, por lo que el modelo a utilizar depende del motor a utilizar, del montaje sobre el rotor del motor y de las limitaciones de espacio en la estructura del dron. Las hélices recomendadas para los motores disponibles en el dron son las 1045L y 1045R (ver Figura 1.11), las cuales son de 10" de diámetro y 4.5" de inclinación.

Estas hélices en conjunto con los motores antes mencionados permiten levantar un peso de hasta 2.4 [Kg] si la configuración seleccionada para el dron es la de un hexarotor [11]. El peso aproximado del hexarotor con todos sus componentes es de 2.2 [Kg].



Figura 1.11. Hélices 1045L y 1045R [11].

La Figura 1.12 muestra el empuje resultante de los motores en función del pulso de control que se envía al ESC y del modelo de hélice utilizada, donde 1045L/R y 8045L/R son modelos de hélices [13].

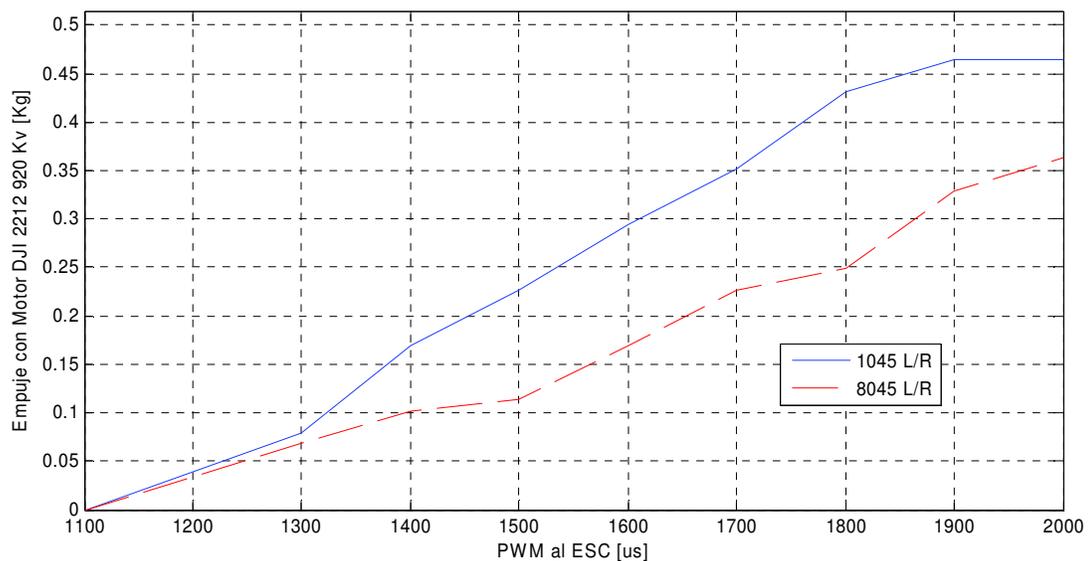


Figura 1.12. Empuje en función del pulso de control [13].

1.4.2.5 Arduino DUE

El hexarotor cuenta con un Arduino Due como tarjeta de control principal (ver Figura 1.13). En él se encuentra el algoritmo de control para el manejo del dron, además éste recopila datos de los sensores para de esa manera realizar la estimación de orientación y posteriormente usar estos datos en el control. Las características principales de esta tarjeta se muestran en la Tabla 1.2 [15].

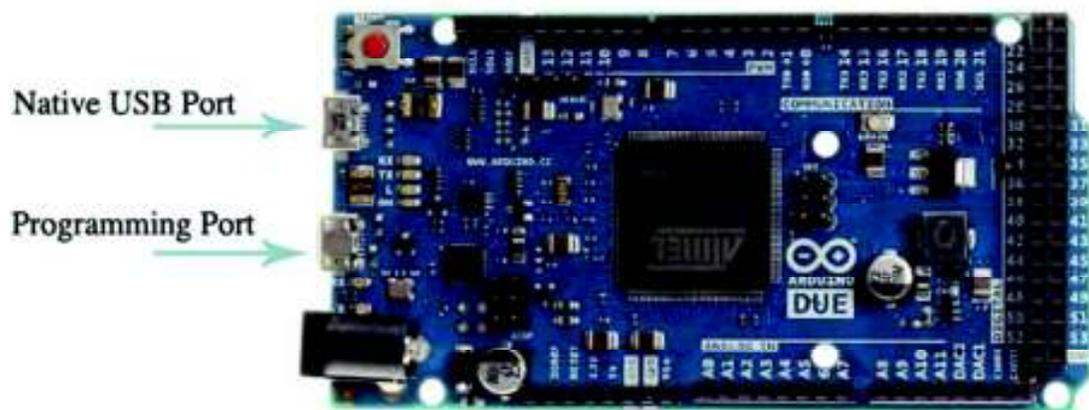


Figura 1.13. Tarjeta embebida Arduino DUE [11].

Tabla 1.2. Características de Arduino DUE [15].

Microcontrolador	AT91SAM3X8E
Entradas / Salidas digitales	54 (12 pueden usarse como salidas PWM)
Entradas analógicas	12
Salidas analógicas	2 (DAC)
Voltaje de alimentación	7 – 12 V
Corriente total soportada en todas las líneas DC I/O (Entradas/Salidas)	130 mA
Voltaje de operación	3.3 V
Corriente DC máxima en 3.3 V	800 mA
Corriente DC máxima en 5 V	800 mA
Flash Memory	512 KB
SRAM	96 KB
Cristal oscilador	84 MHz
Conectores	SPI y JTAG
Peso	36 g
Controlador DMA	Alivia a la CPU de realizar tareas que requieren mucha memoria.

1.4.2.5.1 Comparación Ardupilot Mega y tarjeta embebida Arduino DUE

La tarjeta embebida Arduino DUE es de propósito general, mientras que la tarjeta embebida Ardupilot Mega es un autopiloto comercial de código abierto dedicado para el control de vehículos aéreos de ala fija, multirrotores, barcos y coches [11].

Para el autopiloto Ardupilot Mega existe software dedicado para el control de vehículos aéreos; al contrario, para el Arduino Due no se encuentra disponible este tipo de software.

El software dedicado del autopiloto es común para todos los tipos de aeronaves por lo que el usuario debe configurar en el software el tipo de aeronave y el tipo de sensores que utilizará. Además, el software dedicado del autopiloto usa control PID [11].

Para realizar cambios en el software dedicado del autopiloto se debe conocer bien la estructura del programa y las librerías que utiliza.

El autopiloto tiene sensores embebidos como: giroscopio de tres ejes, acelerómetro de tres ejes y un barómetro. En la tarjeta embebida Arduino Due no hay sensores presentes.

1.4.2.6 Tarjeta IMU MPU6050

La IMU MPU6050 (ver Figura 1.14) es un dispositivo electrónico de seguimiento de movimiento, tiene 6 grados de libertad de los cuales 3 ejes corresponden al giroscopio (ADC de 16 - bits) y los otros 3 al acelerómetro (ADC de 16 bits). Se puede comunicar a través del protocolo I2C serial, y usa un voltaje de alimentación de 2.375 a 3.46 [V] [16].

Esta IMU no realiza la estimación de la orientación en ángulos de Euler por sí misma, tan solo ofrece la posibilidad de filtrar los datos de los sensores que posee usando un filtro pasa bajo. Debido a esto la estimación de la orientación debe ser realizada por el usuario. Además es necesario escoger la escala de medición tanto para el acelerómetro como para el giroscopio, el valor del rango de medición es dependiente de la aplicación.

Esta es la IMU que se instaló en el hexarotor y con cuyas mediciones se estima la orientación del hexarotor en ángulos de Euler. La IMU se montó sobre la estructura del

hexarotor utilizando cinta de doble faz y asegurándose de colocar el circuito integrado que contiene los sensores en el centro de toda la estructura.

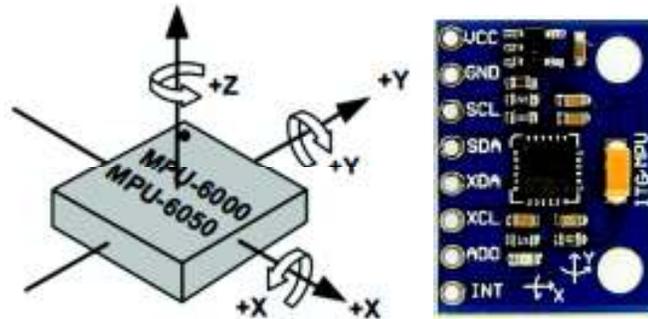


Figura 1.14. IMU MPU6050 [16].

1.4.2.7 Batería LIPO 11.1V 9000mAh 3S1P CELL 60C-120C

Estas baterías constan de celdas de 3.7 voltios que se conectan en serie y/o paralelo para lograr la capacidad deseada. El peso de esta batería es de aproximadamente 689 gramos, ver Figura 1.15 [17].

La nomenclatura 3S1P indica que esta batería consta de 3 celdas conectadas en serie (3S) y que este conjunto de celdas en serie solo se repite una vez (1P) por lo que no hay más bloques de 3S conectados en paralelo entre sí [11].



Figura 1.15. Batería LiPo 11.1V 9000mAh 3S1P Cell 60C-120C [17].

Los términos 9000 mAh y 60C indican que la batería puede entregar 9000 mA durante una hora y la tasa de descarga puede llegar a ser de hasta 60 veces la corriente nominal de 9000 mAh; es decir, 540 Ah será la corriente máxima que puede entregar la batería. Las celdas pueden cargarse hasta 4.2 voltios, por lo que la batería tendrá 12.6 voltios al estar completamente cargada y 11.1 voltios al estar descargada al máximo [11].

Una forma sencilla de dimensionar la capacidad de la batería es considerando el consumo de los motores, porque son los elementos que mayor energía demandan [11]. El consumo de los motores influirá directamente en el tiempo de vuelo que nos permitirá la batería escogida. Para empezar con el cálculo de tiempo de vuelo proporcionado con esta batería se considera que desde el inicio hasta el final del vuelo los motores estarán en promedio a un 50% de su capacidad, esto es 15 amperios por motor; porque este valor es suficiente para mantener al dron a una altura aproximada de 1 metro [11].

Con la Ecuación 1.28 se calcula el flujo de corriente demandado por los motores y usando la Ecuación 1.29 se obtiene el tiempo de vuelo que permite esta batería [11].

$$\text{Flujo de corriente} = \frac{I_{total}}{\text{Tiempo de descarga}} = \frac{15 [A] \cdot 6 [\# \text{ de motores}]}{60 [min]} = 1.5 \left[\frac{A}{min} \right]$$

Ecuación 1.28. Flujo de corriente requerido por los motores [11].

La corriente máxima entregada por esta batería es de 540 A, por lo que si soporta la demanda de los motores de 180 A ($6 \times 30 A$) cuando estos están a su máxima capacidad [11].

$$\text{Tiempo de vuelo} = \frac{\text{Capacidad de la batería}}{\text{Flujo de corriente}} = \frac{9000 [mA]}{1.5 \left[\frac{A}{min} \right]} = 6 [min]$$

Ecuación 1.29. Tiempo de vuelo alcanzado con la batería [11].

1.4.2.8 Batería LIPO TURNIGY NANO-TECH 300mAh 2S 35~70C



Figura 1.16. Batería LiPo Turnigy nano-tech 300mah 2S 35~70C [18].

La batería LiPo Nano se usa para alimentar a todos los dispositivos electrónicos del hexarotor tales como el Arduino DUE y la IMU MPU6050, esta batería se añadió para

separar la parte de potencia de la de control. Su peso es de aproximadamente 28 gramos (ver Figura 1.16) [18].

1.4.2.9 Conversor USB A UART

El conversor USB a UART CP2102 (ver Figura 1.17) Permite la conexión del puerto serial del Arduino DUE con el puerto USB del computador para recibir datos y enviar comandos. Permitiendo de esta manera el manejo del hexarotor y el almacenamiento de los datos enviados por el Arduino DUE durante el vuelo [19].

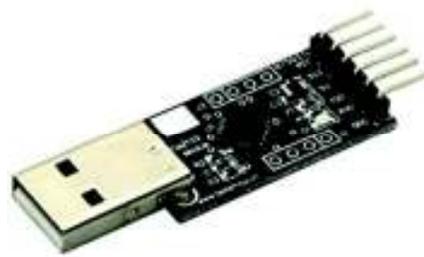


Figura 1.17. Conversor USB a UART CP2102 [19].

1.4.2.10 Control de manejo GENIUS G-08XU

El control de manejo se conecta a un computador. El computador se usa para recibir los datos provenientes del controlador y para enviar los comandos del usuario al controlador. Para realizar esto se desarrolló un algoritmo en MATLAB el cual permite el uso de un control de mano para el manejo del hexarotor y también almacena las tramas de datos del controlador al mismo tiempo, posteriormente el programa permite separar los datos de las tramas almacenadas y presentarlos de forma gráfica.

Este control cuenta con 4 botones de dirección y 8 botones, usa conexión USB (ver Figura 1.18) y no requiere de la instalación de un driver debido a que cuenta con tecnología *Plug&Play* [20]. El control se usa para manejar al dron (variar *setpoints* de ángulos de orientación y el empuje para variar la altura), de esta manera el piloto puede visualizar todo el tiempo al hexarotor sin desviar su atención del mismo. El uso de un control mejora considerablemente el tiempo de reacción que tendrá el piloto.



Figura 1.18. Control Genius G-08XU [20].

1.4.3 CONTROL POR MODOS DESLIZANTES (SMC)

El control SMC (*Sliding Mode Control*) es una técnica de control no lineal y requiere del modelo del sistema para generar la ley de control. Las principales ventajas de este controlador son: su robustez, presenta insensibilidad a la variación paramétrica y tiene buen seguimiento [5].

El control SMC consiste en definir una ley de control la cual consta de dos partes: fase de alcanzabilidad y fase deslizante, las cuales se muestran en la Ecuación 1.30 [10].

$$U_{SMC} = U_{eq} + U_d$$

Ecuación 1.30. Ley de control SMC.

Fase de alcanzabilidad (Control discontinuo - U_d): se realiza mediante una acción de control discontinua y es la parte del control responsable de que el sistema alcance la superficie deslizante [21], [22].

Fase deslizante (Control equivalente - U_{eq}): es un valor continuo que representa el valor medio del control discontinuo y es la parte del control que lleva al sistema a deslizarse sobre la superficie deslizante [21], [22].

Para diseñar un control SMC se debe determinar una superficie deslizante (s) que cumpla la siguiente condición $\dot{s} = 0$ y la condición de estabilidad por Lyapunov que viene dada por $\dot{s} \cdot s < 0$ [21].

Los autores Jean – Jacques E. Slotiene y Weiping Li proponen las superficies de deslizamiento de la Ecuación 1.31 y de la Ecuación 1.32, donde n indica el orden del sistema a controlar [23].

$$s = \left(\frac{d}{dt} + \lambda \right)^{n-1} \cdot e(t)$$

Ecuación 1.31. Superficie diferencial de orden $n-1$.

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

Ecuación 1.32. Superficie integral – diferencial de orden n .

El control SMC presenta el fenómeno del *chattering*. Se conoce como *chattering* a las oscilaciones de la variable controlada alrededor del *setpoint*, estas oscilaciones se caracterizan por ser de amplitud y frecuencia finita. El *chattering* aparece debido a que la acción de control es discontinua y también a causa de las dinámicas no modeladas. El *chattering* puede reducir la exactitud del control pero este fenómeno puede atenuarse usando métodos de suavizamiento de *chattering* [5], [24].

A continuación se presentan los métodos de suavizamiento que se pueden usar para disminuir el fenómeno *chattering*.

1.4.3.1 Métodos de suavizamiento del *chattering*

La señal de control para la fase de alcanzabilidad está dada por la Ecuación 1.33. Donde K_d es el parámetro sintonización de la fase de alcanzabilidad del control SMC (ver Figura 1.19) [21].

$$U_D = K_d \cdot \text{sign}(s) \quad \text{sign}(s) = \begin{cases} 1 & s > 0 \\ 0 & s = 0 \\ -1 & s < 0 \end{cases}$$

Ecuación 1.33. Función de alcanzabilidad con la función signo.



Figura 1.19. Función signo.

La función signo produce oscilaciones de alta frecuencia que conllevan al fenómeno del *chattering*, para suavizar este fenómeno se puede implementar los siguientes métodos de suavizamiento del *chattering*:

- **Método de suavizamiento mediante sigmoide:**

La función sigmoide (ver Figura 1.20) se utiliza para reemplazar la función signo y suavizar el fenómeno del *chattering* y está dada por la Ecuación 1.34. El parámetro delta (δ) permite suavizar la acción de control [21].

$$U_D = K_d \cdot \text{sigmoide}(s) \quad \text{sigmoide}(s) = \frac{s}{|s| + \delta}$$

Ecuación 1.34. Función sigmoide.

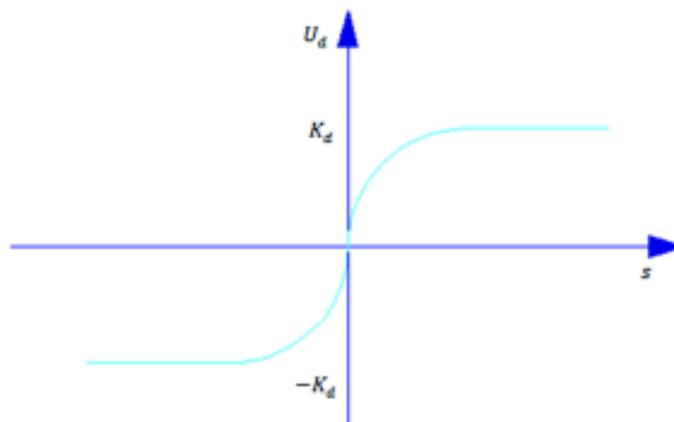


Figura 1.20. Función sigmoide.

- **Método de suavizamiento mediante zona muerta:**

La zona muerta (ver Figura 1.21) es la función signo con zona muerta y está dada por la Ecuación 1.35 [21].

$$U_D = K_d \cdot \text{zonamuerta}(s) \quad \text{zonamuerta}(s) = \begin{cases} K_d & s \geq v \\ 0 & -v < s < v \\ -K_d & s \leq -v \end{cases}$$

Ecuación 1.35. Zona muerta

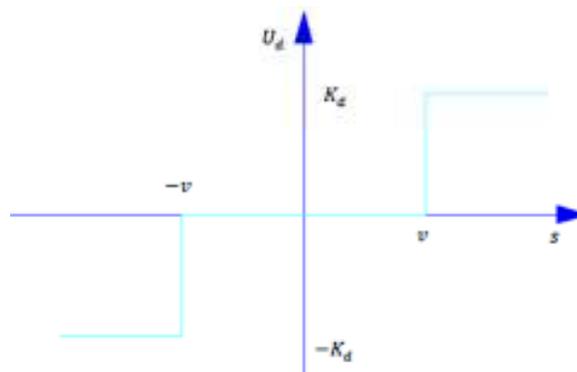


Figura 1.21. Zona muerta.

- **Método de suavizamiento mediante capa de frontera:**

La capa de frontera sustituye la función signo por la función $\text{sat}(s)$, donde delta (δ) permite suavizar la acción de control y es una constante positiva correspondiente al ancho de la capa de frontera y s es la superficie deslizante. La función sat está dada por la Ecuación 1.36, la zona $|s| < \delta$ indica que se está dentro de la superficie deslizante (ver Figura 1.22) [22].

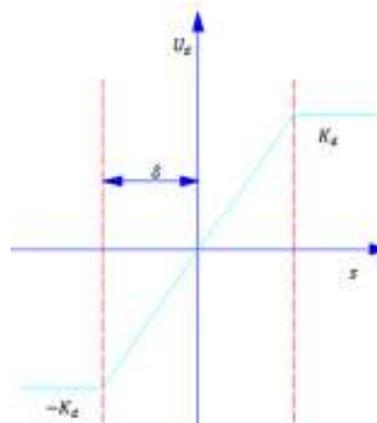


Figura 1.22. Suavizamiento utilizando método de capa de frontera

$$U_D = K_d \cdot sat(s) \quad sat(s) = \begin{cases} \frac{s}{\delta} & |s| < \delta \\ sign(s) & |s| \geq \delta \end{cases}$$

Ecuación 1.36. Función sat.

Existe un rango de valores de δ y K_d donde se puede garantizar el funcionamiento correcto del controlador [22].

1.4.3.2 Sintonización de controladores SMC

A continuación se detallan los procedimientos existentes para la sintonización de controladores SMC. Se puede simular el modelo del sistema a controlar y encontrar los parámetros adecuados para los controladores usando ecuaciones de sintonización, metodología heurística o usando algoritmos genéticos [25].

1.4.3.2.1 Por curva de reacción

Este método requiere que la respuesta del sistema ante una entrada paso tenga la forma de una “s” como se ve en la Figura 1.23, en donde $m(t)$ es la señal de control ingresada al sistema y $c(t)$ es la respuesta de la variable a controlar. Los parámetros que se deben obtener de la curva de reacción son K , t_0 y τ (ganancia en estado estacionario del proceso, tiempo muerto y constante de tiempo respectivamente), debido a que se necesita de ellos para usar las ecuaciones de sintonización de un control SMC [26].

De la curva de reacción del proceso se deben obtener los valores de t_0 y τ , para ello se debe trazar una recta tangente a la curva de reacción en el punto de máxima tasa de cambio como se ve en la Figura 1.24. Para encontrar K se usa la Ecuación 1.37 [26].

$$K = \frac{\Delta C_s}{\Delta m}$$

Ecuación 1.37. Ganancia en estado estacionario del proceso.

Para sintonizar un control SMC con los parámetros de la curva de reacción se usan las ecuaciones propuestas en [27], donde la superficie deslizante para el control SMC es de tipo PID.

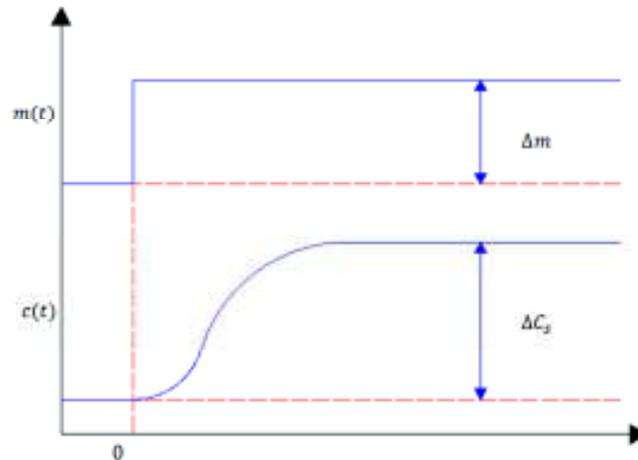


Figura 1.23. Curva de reacción del proceso [26].

Las ecuaciones de sintonización se muestran de la Ecuación 1.38 a la Ecuación 1.43, donde $e(t)$ es el error de la variable a controlar y K_d , δ , λ_0 y λ_1 son los parámetros de sintonización del control SMC [27].

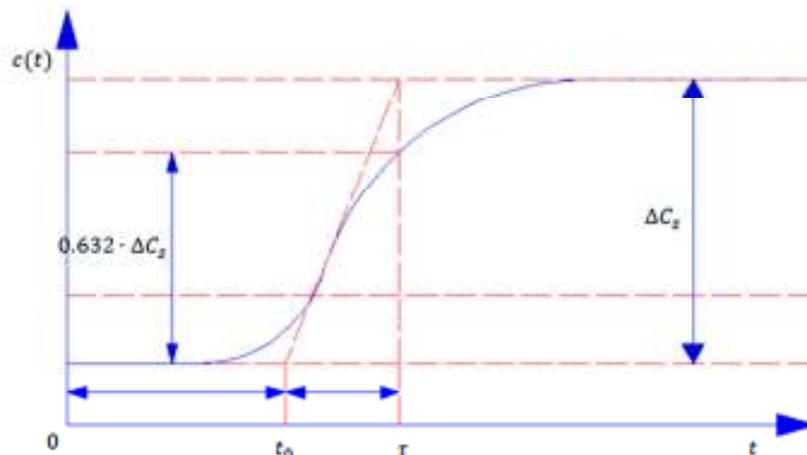


Figura 1.24. Parámetros de la curva de reacción [26].

$$U_{SMC}(t) = \left(\frac{t_0 + \tau}{K} \right) \cdot \left[\left(\frac{t_0 + \tau}{t_0 \cdot \tau} - \lambda_1 \right) \cdot \frac{d c(t)}{dt} + \frac{c(t)}{t_0 \cdot \tau} + \lambda_0 \cdot e(t) \right] + K_d \cdot \frac{s(t)}{|s(t)| + \delta}$$

Ecuación 1.38. Ley de control SMC.

$$s(t) = \text{signo}(K) \cdot \left[\frac{d e(t)}{dt} + \lambda_1 \cdot e(t) + \lambda_0 \cdot \int_0^t e(t) dt \right]$$

Ecuación 1.39. Superficie de deslizamiento.

$$\lambda_1 = \left[\frac{0.167}{K^{0.00952006} \cdot \tau^{0.23411834}} + 0.8558 \cdot \left(\frac{\tau}{t_0} \right)^{0.0163} \right]^{27.8127}$$

Ecuación 1.40. Ecuación de sintonización para λ_1 .

$$\lambda_0 = 0.7072 \cdot \left(\frac{1}{\tau} \right)^{1.7331} \cdot \left(\frac{\tau}{t_0} \right)^{1.0898}$$

Ecuación 1.41. Ecuación de sintonización para λ_0 .

$$K_d = \frac{35.5364 \cdot \tau^{0.5924}}{K^{0.7997}}$$

Ecuación 1.42. Ecuación de sintonización para K_d .

$$\delta = \frac{27.9585}{K^{2.589} \cdot \tau^{6.6041}} \cdot \left(\frac{\tau}{t_0} \right)^{7.45662} + 72.4675$$

Ecuación 1.43. Ecuación de sintonización para δ .

Este método no es aplicable para el hexarotor debido a que su respuesta ante una entrada tipo escalón no tiene forma de “s”.

1.4.3.2.2 Por algoritmos genéticos

Los algoritmos genéticos son un método aleatorio usado para encontrar los valores más adecuados para los parámetros de los controladores usando como método de evaluación una función de desempeño o coste definida por el usuario [25].

El sistema a controlar se simula para poder usar los algoritmos genéticos y encontrar los parámetros adecuados para los controladores. Para la sintonización es necesario tener el modelo de la planta a controlar para poder realizar la sintonización utilizando los métodos mencionados anteriormente [25].

En [28] se usó algoritmos genéticos para sintonizar los parámetros de los controladores SMC de orientación de un quadrotor, en una simulación. La función de desempeño utilizada, consideraba el tiempo de establecimiento y los sobre picos. El control SMC sintonizado mostró un buen desempeño aún fuera de las condiciones de suspensión [28]

1.4.3.2.3 Por metodología heurística

La práctica más común para la sintonización de un control SMC es usar la metodología heurística para encontrar los valores adecuados para los parámetros de los controladores.

Este método se utilizó en [10] para la sintonización de los parámetros de los controladores SMC de altura y orientación de un hexarotor en una simulación. Los resultados mostraron un buen seguimiento de los *setpoints*.

La sintonización de los controladores en este trabajo de titulación se realizó por medio de la metodología heurística, y a partir de esta se propone un procedimiento de sintonización del control SMC como se explica en la sección 2.3.

1.4.4 CONTROL PID

El control PID (Proporcional Integral Derivativo) es una técnica de control realimentado con una estructura simple. Es un controlador para sistemas SISO (una entrada y una salida), aunque también es usado en sistemas multivariable. Además, algunas plantas inestables no pueden ser estabilizadas con controladores PID [29].

El Control PID (ver Ecuación 1.44) incluye tres acciones [29]:

- **Proporcional (P):** es un control cuya salida es proporcional al error de la variable controlada.
- **Integral (I):** la salida de este controlador es igual al error acumulado de la variable controlada a lo largo del tiempo. Es un controlador lento.
- **Derivativo (D):** es un controlador de carácter predictivo y hace más rápida la acción de control. Su desventaja es que puede amplificar el ruido presente en la señal de error.

$$PID = P \cdot e(t) + I \cdot \int e(t) dt + D \cdot \frac{de(t)}{dt}$$

Ecuación 1.44. Expresión matemática del control PID.

Los drones comerciales tienen tarjetas dedicadas conocidas como autopilotos, estas constan de controladores PID en cascada para los ángulos de orientación. En la siguiente sección se explica cómo funciona el control en cascada, debido a que es el tipo de controlador que se implementará para compararlo con el control SMC.

1.4.4.1 Control en cascada

El control en cascada es un control realimentado en el cual la salida de un controlador realimentado es la referencia para otro controlador realimentado. Al controlador cuya salida es la referencia del segundo controlador se lo conoce como control primario, externo o maestro. De igual manera, al controlador que recibe como referencia la salida de un controlador se lo conoce como control secundario, interno o esclavo. La estructura del control en cascada se muestra en la Figura 1.25 [29].

Las ventajas de usar este esquema de control son [29]:

- Se elimina el efecto de algunas perturbaciones por lo que la respuesta del sistema se vuelve más estable.
- Se mejora la dinámica de la respuesta del lazo de control.
- Las perturbaciones que ocurren en el lazo interno son compensadas por el controlador interno antes de que afecten al lazo externo.
- La constante de tiempo del lazo secundario es baja, por lo que la compensación del control es más rápida.
- El control en cascada es menos sensible a los errores de modelado.

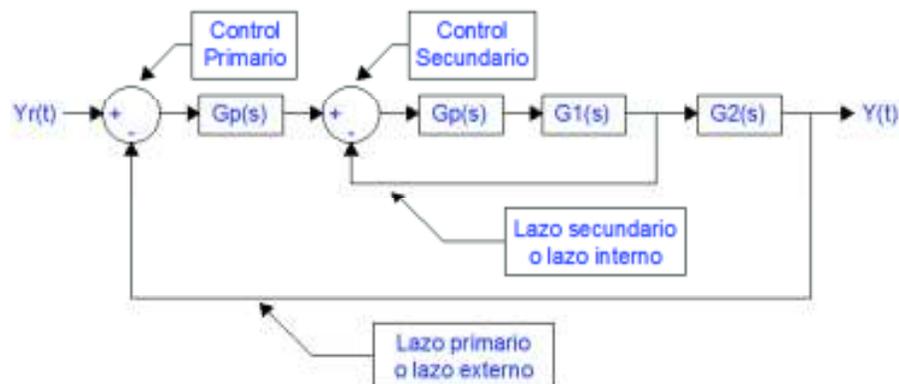


Figura 1.25. Estructura del control en cascada.

1.4.4.2 Sintonización para el control PID

1.4.4.2.1 Zeigler y Nichols

Para este método se requiere al sistema a controlar un control tipo P en lazo cerrado. El objetivo es lograr que la variable controlada oscile con amplitud y frecuencia constante, como se ve en la Figura 1.26 [30].

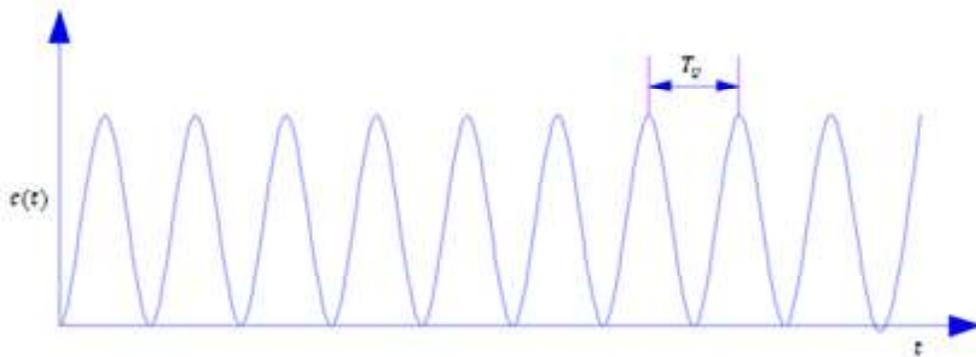


Figura 1.26. Variable controlada en oscilación constante.

El control se sintoniza como se ve en la Tabla 1.3 usando la ganancia (K_{CU}) que llevo a la variable controlada a la oscilación y el periodo de oscilación (T_U) al que oscila la variable controlada [30].

Tabla 1.3 Sintonización de PID por Zeigler Nichols.

Control	Ganancia Proporcional K_C	Tiempo de integración T_I	Tiempo de derivación T_D
P	$\frac{K_{CU}}{2}$	-	-
PI	$\frac{K_{CU}}{2.2}$	$\frac{T_U}{1.2}$	-
PID	$\frac{K_{CU}}{1.7}$	$\frac{T_U}{2}$	$\frac{T_U}{8}$

La relación entre el tiempo integral (T_I) y el tiempo derivativo (T_D) con los parámetros del control PID se muestra en la Ecuación 1.45.

$$PID = K_C \cdot \left(1 + \frac{1}{T_I} \cdot \int e(t) dt + T_D \cdot \frac{de(t)}{dt} \right) = P \cdot e(t) + I \cdot \int e(t) dt + D \cdot \frac{de(t)}{dt}$$

Ecuación 1.45. PID con tiempo integral y derivativo.

1.4.4.2.2 Por curva de reacción

Este método requiere que la respuesta del sistema ante una entrada paso tenga la forma de una “s” como se ve en la Figura 1.23, en donde $m(t)$ es la señal de control ingresada al sistema y $c(t)$ es la respuesta de la variable a controlar. Los parámetros que se deben obtener de la curva de reacción son K , t_0 y τ [30].

La obtención de los parámetros de la curva de reacción se describió en la sección 1.4.3.2.1. El control PID se sintoniza como se muestra en la Tabla 1.4 [30].

Tabla 1.4 Sintonización de PID por curva de reacción

Control	Ganancia Proporcional K_C	Tiempo de integración T_I	Tiempo de derivación T_D
P	$\frac{1}{K} \cdot \left(\frac{\tau}{t_0} \right)$	-	-
PI	$\frac{0.9}{K} \cdot \left(\frac{\tau}{t_0} \right)$	$3.33 \cdot t_0$	-
PID	$\frac{1.2}{K} \cdot \left(\frac{\tau}{t_0} \right)$	$2 \cdot t_0$	$\frac{1}{2} \cdot t_0$

1.4.4.2.3 Por metodología heurística

El control PID se puede sintonizar de forma heurística teniendo en consideración como afecta cada parte del control a la dinámica de la variable controlada [30].

1.4.4.2.4 Para PID en cascada

Para la sintonización, se empieza con el control secundario y posteriormente con el primario. La sintonización se puede realizar con cualquiera de los métodos mencionados anteriormente [29].

1.4.5 ESTIMACIÓN DE ORIENTACIÓN EN UAVs

En la actualidad los sensores más utilizados para la estimación de la orientación de UAVs durante el vuelo son: magnetómetros, acelerómetros y giroscopios. Con la ayuda de estos sensores se puede obtener una estimación de la orientación del dron. Entre las formas en las que se puede estimar la orientación de un vehículo aéreo se encuentran: ángulos de Euler, orientación relativa o absoluta en cuaterniones o por matriz de rotación [11].

Existen tarjetas embebidas que combinan uno o más sensores de los mencionados anteriormente, a estas se las conoce como unidades de medición inercial (IMU – *Inertial Measurement Unit*) [11].

Cada tipo de sensor posee ventajas y desventajas y por este motivo se suele usar las medidas de varios tipos de sensores para estimar la orientación de un vehículo en el espacio. La fusión de sensores es un procedimiento que permite aprovechar las ventajas de cada sensor al momento de obtener una estimación de la orientación en el espacio de un objeto [31], [32].

1.4.5.1 Magnetómetro

Un magnetómetro detecta el campo magnético generado por la tierra y se usa como brújula para sistemas de navegación. Es un sensor muy sensible a los campos magnéticos que se encuentren a su alrededor [33].

1.4.5.2 Giroscopio

El giroscopio es un dispositivo que mide la velocidad angular, se usa para medir la rotación. Las unidades de medición son grados por segundo ($^{\circ}/s$) o revoluciones por segundo (*rps*). Su principal aplicación se encuentra en sistemas de navegación autónomos, donde son usados para determinar la orientación de los vehículos [33].

1.4.5.3 Acelerómetro

El acelerómetro es un dispositivo electromecánico que mide las fuerzas de aceleración que representan la tasa de cambio de la velocidad de un objeto. Las unidades de medición son metros por segundo al cuadrado (m/s^2) o fuerzas gravitacionales (g), las cuales se reciben mediante una señal eléctrica que varía según la aceleración física a la que es sometido el sensor. Se utiliza en aplicaciones de orientación o para detectar las vibraciones en los sistemas. Los principales modelos de acelerómetro son: acelerómetros piezoresistivos, acelerómetros piezoeléctricos y acelerómetros capacitivos [34], [35].

1.4.5.4 Unidad de medición inercial (IMU)

La IMU es un dispositivo electrónico que combina acelerómetro, giroscopio y/o magnetómetro; mide velocidad, orientación y fuerzas gravitacionales. Se utiliza principalmente para sistemas de navegación de barcos, aviones (como los UAV), misiles, naves espaciales, satélites y aterrizadores. La IMU se usa para determinar la orientación de un objeto en el espacio [31], [32].

1.4.5.5 Fusión de sensores

En esta sección se indica la fusión de sensores que se usa para estimar los ángulos roll y pitch utilizando las mediciones del giroscopio y del acelerómetro. El ángulo yaw solo se estima con el giroscopio ya que no es posible estimarlo con el acelerómetro.

Los datos entregados por el giroscopio y por el acelerómetro no pueden ser usados directamente para determinar la orientación del hexarotor en el espacio debido a que presentan inconvenientes; los cuales se describen posteriormente en esta sección. Sin embargo, al combinar las mediciones de ambos sensores se puede obtener una estimación bastante precisa de la orientación en el espacio de un objeto [31].

El problema con el acelerómetro es que al momento de encender los motores y empezar a maniobrar el hexarotor, no solo se estará obteniendo medidas del vector de gravedad sino que también aparecerán las fuerzas que sirven para la sustentación y el desplazamiento del drone. Este tipo de sensor al ser muy sensible ante fuerzas externas

presentará un alto grado de perturbación en su medición debido al funcionamiento de los motores. Sin embargo, la medición entregada por el acelerómetro aún es confiable a largo plazo por lo que se recomienda filtrar su medición [31].

Los giroscopios no son fácilmente influenciados por fuerzas externas, pero su medida tiende a presentar una desviación a lo largo del tiempo. Al usar sólo los datos del giroscopio para estimar una posición angular se corre el riesgo de que la operación de integración utilizada para estimar la posición angular vaya acumulando este error de deriva con el tiempo. Esto provoca que la medición del giroscopio solo sea confiable a corto plazo [31].

Una forma sencilla de combinar las ventajas de ambos sensores y al mismo tiempo disminuir al máximo las desventajas es usar una fusión de sensores. En resumen, en la fusión de sensores se usa las mediciones del giroscopio solo en el corto plazo, mientras que las mediciones del acelerómetro se consideran en el largo plazo. La forma más sencilla de implementar la fusión de sensores se muestra en la Ecuación 1.46, donde: [31]

- *porc*: constante de confiabilidad, puede variar entre 0 y 1.
- $\hat{\phi}_k$: estimación del ángulo en el instante k .
- *giro*: medida del giroscopio en radianes por segundo.
- Δt : intervalo de tiempo usado en la integración.
- $\hat{\phi}_a$: ángulo estimado a partir de las medidas del acelerómetro.

$$\hat{\phi}_k = [porc \cdot (\hat{\phi}_{k-1} + giro \cdot \Delta t)] + [(1 - porc) \cdot \hat{\phi}_a]$$

Ecuación 1.46. Fusión de sensores.

Esta fusión usa los datos del giroscopio obtenidos en cada periodo de muestreo y los integra para obtener una estimación del ángulo con el giroscopio. Luego, combina el ángulo estimado a partir de los datos del giroscopio con el ángulo estimado a partir de los datos del acelerómetro. El parámetro “porc” en la Ecuación 1.46 permite establecer el porcentaje de influencia que tendrá el giroscopio y el acelerómetro en la estimación final de la orientación del hexarotor [31].

1.4.6 FILTRO DE KALMAN

El Filtro de Kalman es un algoritmo óptimo recursivo que permite estimar los estados de una manera óptima para minimizar el índice de error cuadrático medio. El algoritmo de filtro de Kalman únicamente requiere las mediciones actuales, el estado calculado previamente y la matriz de covarianza para estimar de forma óptima el estado actual de un sistema. En la Figura 1.27 se muestran los pasos que se requieren para implementar un filtro de Kalman [36].

La estimación es óptima debido a que el filtro usa el conocimiento previo del comportamiento del sistema para obtener una estimación del estado actual minimizando el error estadísticamente.

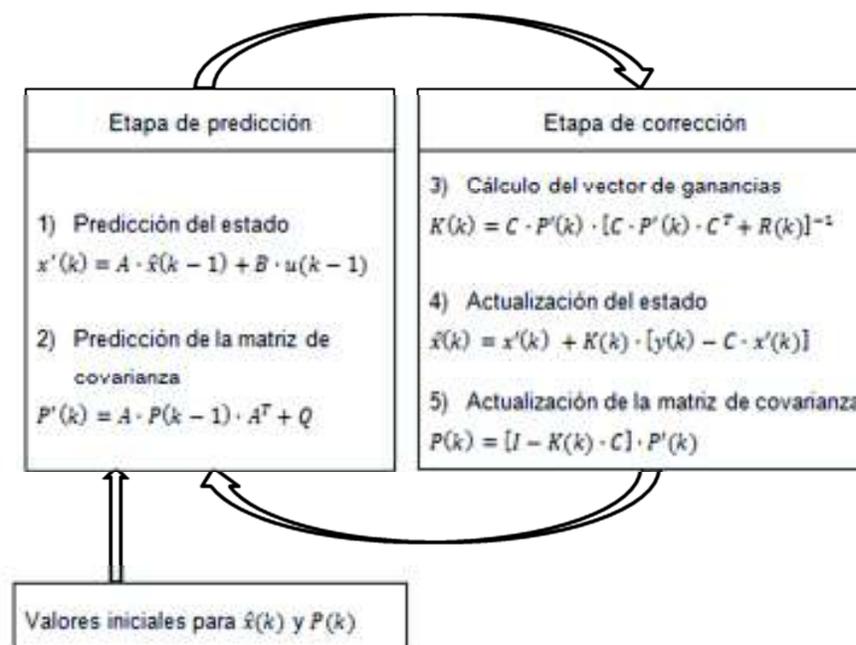


Figura 1.27. Etapas del Filtro de Kalman [36].

El filtro de Kalman consta de dos etapas que son [36], [37]:

- **La etapa de predicción, extra polarización o estimación:** para la primera ejecución del filtro se necesitan los valores iniciales de la estimación de estado y de la matriz de covarianza inicial. En esta etapa se calculan: una predicción de estado $x'(k)$ (el valor de la predicción es calculado a partir del valor más actualizado del estado) y una predicción de la matriz de covarianza del estado ($P(k)$).

La matriz A relaciona el estado en el periodo anterior ($k - 1$) con el estado actual predicho del sistema $x'(k)$ y de forma similar la matriz B relaciona la entrada del sistema en el periodo anterior $u(k - 1)$ con el estado actual predicho del sistema $x'(k)$.

La matriz de covarianza Q representa el error o perturbación del proceso y la matriz de covarianza R representa el error o perturbación de la medida. Se asume que estas matrices son independientes entre ellas y que los errores son ruido blanco con distribución de probabilidad normal.

En la ecuación de predicción del estado se predice el estado del sistema a partir del estado actual. La predicción de la matriz de covarianza, predice el valor de la matriz de covarianza del error.

- **La etapa de corrección o actualización:** en esta etapa se realiza el cálculo del vector de ganancia, la actualización o estimación del estado y la actualización de la matriz de covarianza.

La ganancia de Kalman (K) es un valor que minimiza el error de la actualización o estimación del estado, permite encontrar un valor óptimo del estado del sistema. La matriz C relaciona el estado con la medición de la salida del sistema $y(k)$.

En esta etapa como primer paso se calcula el vector de ganancias $K(k)$, el cual permite minimizar el error en la estimación del estado. Posteriormente, se realiza la actualización o estimación del estado tomando en cuenta la predicción del estado, el vector de ganancias y la salida del sistema.

Finalmente, se realiza la actualización de la matriz de covarianza. Lo cual permite obtener una nueva estimación de la covarianza del error.

Si se implementa el filtro de Kalman en el instante k ; como se muestra en la Figura 1.27, el algoritmo se comporta como un filtro de alisado. En un Filtro de alisado las matrices A y C son matrices identidad, mientras que la matriz B es una matriz nula [36], [37].

Por otra parte, al implementar las ecuaciones del filtro de Kalman en el instante $k + 1$, el filtro de Kalman se comporta como un filtro de predicción [36], [37].

2 METODOLOGÍA

Este capítulo inicia con el diseño de los controladores SMC, se detalla el procedimiento seguido para la obtención de las ecuaciones de los controladores SMC para roll, pitch y yaw. De igual manera, en la sección de diseño de los controladores PID se explica el esquema de control a utilizar para los mismos.

Posteriormente, se llega a la fase de diseño e implementación del software de control. En esta sección se detallan: el programa realizado en la tarjeta embebida Arduino DUE, los filtros utilizados para las mediciones de la IMU y se explica el algoritmo de estimación de los ángulos de orientación. Además se explica el funcionamiento del software desarrollado en MATLAB para la adquisición de datos y el manejo del hexarotor.

En la sección 2.3 y 2.4, se detalla el procedimiento seguido para lograr la sintonización de los parámetros de los controladores SMC y de los parámetros de los controladores PID respectivamente.

Finalmente, en la sección 2.5 (calibración del control de manejo) se describe como se varían los *setpoints* de orientación y la altura para el manejo del hexarotor.

En todas las figuras a mostrarse a continuación se usará la siguiente nomenclatura para las leyendas de las gráficas:

- ϕ, θ, ψ : menciona el ángulo al cual corresponden los datos de la imagen.
- ϕ_d, θ_d, ψ_d : es el *setpoint* del ángulo a controlar.
- U_d : son los valores que toma la fase de alcanzabilidad del control SMC de la variable a controlar.
- U_{eq} : son los valores que toma la fase deslizante del control SMC de la variable a controlar.
- $\dot{\phi}, \dot{\theta}, \dot{\psi}$: es la velocidad de cambio del ángulo a controlar.
- $\dot{\phi}_d, \dot{\theta}_d, \dot{\psi}_d$: es el *setpoint* de velocidad del ángulo a controlar.
- Pulso base de los ESC: es el ancho de pulso en microsegundos de la señal PWM (Pulse width modulation – modulación por ancho de pulso) que se envía a los ESC (*Electronic speed controler*) para variar la velocidad de los motores. El funcionamiento de los ESCs se describió en la sección 1.4.2.2.

2.1 CONTROLADORES DE ORIENTACIÓN

2.1.1 DISEÑO DE LOS CONTROLADORES SMC

La estructura del control SCM implementada se encuentra en la Figura 2.1. El control por modos deslizantes busca que los estados del sistema converjan a la superficie deslizante y que al alcanzarla los estados se deslicen por la superficie. Por lo tanto el primer paso para encontrar las ecuaciones de los controladores SMC es la selección de la superficie a utilizar [10], [21].

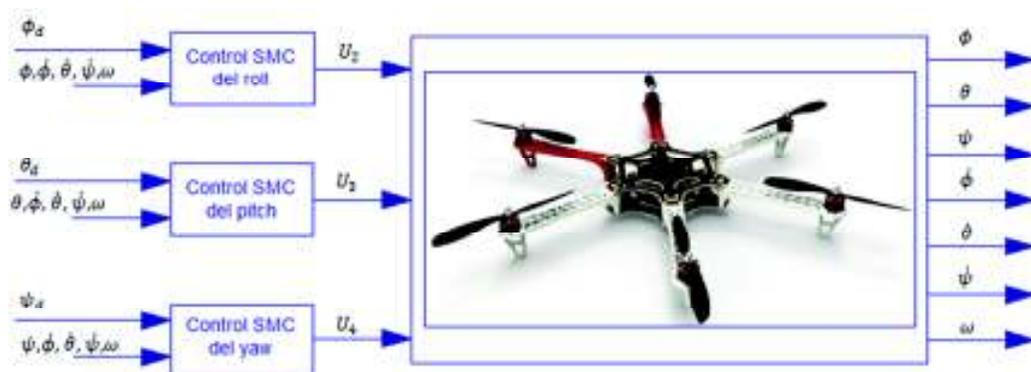


Figura 2.1. Estructura del control SMC implementado.

Se seleccionó una superficie integral - diferencial (ver Ecuación 1.32) porque la parte integral ayuda a corregir perturbaciones y dinámicas no modeladas como es el caso de: corrientes de aire, turbulencias del aire impulsado por los motores, descarga de la batería, el tiempo de aceleración de los motores, el tiempo de frenado de los motores y el control de lazo abierto de los ESC que no miden la velocidad de los motores de forma directa debido a que miden solo la corriente entregada al motor en una fase [5], [10], [21].

Se empezará el diseño con el ángulo roll. Al observar la Ecuación 2.1 se puede ver que la ecuación del modelo del sistema es de segundo orden, por lo tanto el orden usado para generar la superficie a partir de la expresión de la Ecuación 2.2 será de $n = 2$ [10], [21], [23]

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

Ecuación 2.1. Aceleración del ángulo roll.

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

Ecuación 2.2. Superficie deslizante integral - diferencial.

La superficie deslizante a utilizar se muestra en la Ecuación 2.3. Para determinar la acción de control equivalente, la superficie deslizante debe cumplir la condición $\dot{s} = 0$, como se explicó en la sección 1.4.3. La acción de control equivalente es la responsable de mantener al sistema en la condición $\dot{s} = 0$ [10], [21], [23].

$$s = \dot{e} + \lambda_1 \cdot e + \lambda_2 \cdot \int e(t) dt$$

Ecuación 2.3. Superficie deslizante a utilizar.

Para ello, derivamos la Ecuación 2.3 y en ella reemplazamos la definición de la doble derivada del error mostrada en la Ecuación 2.4, donde ϕ_d es el *setpoint* del ángulo roll. Posteriormente igualamos la expresión resultante a cero y obtenemos la expresión de la Ecuación 2.5 [10], [21].

$$e_\phi = \phi_d - \phi$$

$$\dot{e}_\phi = \dot{\phi}_d - \dot{\phi}$$

$$\ddot{e}_\phi = \ddot{\phi}_d - \ddot{\phi}$$

Ecuación 2.4. Definición de los errores.

$$\dot{s} = \ddot{e}_\phi + \lambda_1 \cdot \dot{e}_\phi + \lambda_2 \cdot e_\phi = (\ddot{\phi}_d - \ddot{\phi}) + \lambda_1 \cdot \dot{e}_\phi + \lambda_2 \cdot e_\phi = 0$$

Ecuación 2.5. Expresión resultante al aplicar la condición $\dot{s} = 0$.

$$\ddot{\phi}_d + \lambda_1 \cdot \dot{e}_\phi + \lambda_2 \cdot e_\phi = \dot{\theta} \cdot \psi \cdot \left(\frac{I_y - I_z}{I_x} \right) - \frac{J_r}{I_x} \cdot \dot{\theta} \cdot \omega + \frac{l}{I_x} \cdot U_2$$

Ecuación 2.6. Expresión de la aceleración del roll relacionada con la superficie deslizante.

$$U_{eq} = \frac{I_x}{l} \cdot \left(\ddot{\phi}_d + \lambda_1 \cdot \dot{e}_\phi + \lambda_2 \cdot e_\phi - \dot{\theta} \cdot \psi \cdot \left(\frac{I_y - I_z}{I_x} \right) + \frac{J_r}{I_x} \cdot \dot{\theta} \cdot \omega \right)$$

Ecuación 2.7. Acción de control equivalente para el ángulo roll.

Se puede observar que de la Ecuación 2.5 se puede despejar $\ddot{\phi}$ y reemplazar su valor por la Ecuación 1.25, obteniendo la expresión de la Ecuación 2.6. La acción de control equivalente se obtiene al reemplazar la entrada de control U_2 por la acción de control equivalente U_{eq} en la Ecuación 2.6 y despejarla como se ve en la Ecuación 2.7, la cual representa la acción de control equivalente para el ángulo roll [10], [21].

La ley de control SMC para el ángulo roll puede ser escrita usando la Ecuación 1.30, la Ecuación 1.36 y la Ecuación 2.7 como se muestra en la Ecuación 2.8. Se decidió usar para la acción de control discontinuo el método de suavizamiento del *chattering* mediante capa de frontera [10], [21].

$$U_{SMC\phi} = \frac{I_x}{l} \cdot \left(\ddot{\phi}_d + \lambda_1 \cdot \dot{e}_\phi + \lambda_2 \cdot e_\phi - \dot{\theta} \cdot \psi \cdot \left(\frac{I_y - I_z}{I_x} \right) + \frac{J_r}{I_x} \cdot \dot{\theta} \cdot \omega \right) + K_d \cdot \text{sat}(s)$$

Ecuación 2.8. Ley de control SMC para el ángulo roll.

$$V = \frac{1}{2} \cdot s^2 > 0$$

Ecuación 2.9. Función candidata de Lyapunov definida positiva.

$$\dot{V} = s \cdot \dot{s} < 0$$

Ecuación 2.10. Condición de estabilidad.

Posteriormente, se debe asegurar que la acción de control discontinuo U_d permita alcanzar la superficie deslizante. Para ello se define una función candidata de Lyapunov, (Ecuación 2.9) la cual debe ser definida positiva y además la derivada de esta función debe ser definida negativa (Ecuación 2.10) para que se cumpla la condición de alcanzabilidad [10], [21].

Reemplazando la Ecuación 2.5 en la Ecuación 2.10 se obtiene la Ecuación 2.11, con la cual se puede encontrar la condición de alcanzabilidad. En la Ecuación 2.11 reemplazamos la Ecuación 2.6 habiendo realizado previamente el cambio de U_2 por la ley de control SMC del roll $U_{SMC\phi}$ para obtener la expresión de la Ecuación 2.12 [10], [21].

$$(\ddot{\phi}_d - \ddot{\phi} + \lambda_1 \cdot \dot{e}_\phi + \lambda_2 \cdot e_\phi) \cdot s < 0$$

Ecuación 2.11. Expresión para encontrar la condición de alcanzabilidad.

$$\left(\ddot{\phi}_d - \left(\dot{\theta} \cdot \psi \cdot \left(\frac{I_y - I_z}{I_x} \right) - \frac{J_r}{I_x} \cdot \dot{\theta} \cdot \omega + \frac{l}{I_x} \cdot U_{SMC\phi} \right) + \lambda_1 \cdot \dot{e}_\phi + \lambda_2 \cdot e_\phi \right) \cdot s < 0$$

Ecuación 2.12. Expresión obtenida al reemplazar la ley de control SMC del roll.

Se debe reemplazar $U_{SMC\phi}$ de la Ecuación 2.12 por la expresión de la Ecuación 2.8, luego se procede a expandir la expresión resultante y a eliminar los términos semejantes hasta que se obtiene la expresión mostrada en la Ecuación 2.13 [10], [21].

$$-\frac{l}{I_x} \cdot K_d \cdot \text{sat}(s) \cdot s < 0$$

Ecuación 2.13. Expresión para usada para verificar la condición de alcanzabilidad.

Con la desigualdad obtenida en la Ecuación 2.13 se debe analizar las condiciones que deben cumplirse para que se mantenga la desigualdad, teniendo en cuenta para esto que l e I_x son constantes y que K_d es un parámetro de ajuste del control. Las condiciones necesarias para mantener la desigualdad se muestran en la Ecuación 2.14, de las condiciones se puede concluir que K_d debe ser positiva para que se cumpla la condición de alcanzabilidad [10], [21].

$$\text{si } s > 0 \rightarrow \text{sat}(s) > 0 \rightarrow K_d > 0$$

$$\text{si } s < 0 \rightarrow \text{sat}(s) < 0 \rightarrow K_d > 0$$

Ecuación 2.14. Condiciones para mantener la desigualdad.

Para encontrar las leyes de control de los ángulos pitch y yaw se sigue el mismo procedimiento utilizado anteriormente para el ángulo roll, las leyes de control resultantes se muestran en la Ecuación 2.15 y en la Ecuación 2.16 respectivamente [10], [21].

$$U_{SMC\theta} = \frac{I_y}{l} \cdot \left(\ddot{\theta}_d - \dot{\phi} \cdot \psi \cdot \left(\frac{I_z - I_x}{I_y} \right) + \frac{J_r}{I_y} \cdot \dot{\phi} \cdot \omega + \lambda_1 \cdot \dot{e}_\theta + \lambda_2 \cdot e_\theta \right) + K_d \cdot \text{sat}(s)$$

Ecuación 2.15. Ley de control SMC para el ángulo pitch.

$$U_{SMC\psi} = \frac{I_z}{l} \cdot \left(\ddot{\psi}_d - \dot{\phi} \cdot \dot{\theta} \cdot \left(\frac{I_x - I_y}{I_z} \right) + \lambda_1 \cdot \dot{e}_\psi + \lambda_2 \cdot e_\psi \right) + K_d \cdot \text{sat}(s)$$

Ecuación 2.16. Ley de control SMC para el ángulo yaw.

Como se puede observar en la Ecuación 2.8, Ecuación 2.15 y Ecuación 2.16, aparecen los términos de la derivada del error y el *setpoint* de aceleración de la variable a controlar. A continuación se mostrara cómo se obtienen estos términos.

Para la derivada del error \dot{e}_ϕ se puede ver en la Ecuación 2.4 que este depende de la diferencia entre el *setpoint* de velocidad y el valor actual de la velocidad de roll. El valor actual de la velocidad del roll es entregado por el giroscopio luego de ser alisado por un filtro de Kalman y el *setpoint* de velocidad se obtiene usando el control proporcional mostrado en la Ecuación 2.17.

$$\begin{aligned}\dot{\phi}_d &= P_\phi \cdot e_\phi \\ \dot{\theta}_d &= P_\theta \cdot e_\theta \\ \dot{\psi}_d &= P_\psi \cdot e_\psi\end{aligned}$$

Ecuación 2.17. Cálculo del *setpoint* de velocidad.

El *setpoint* de aceleración de la variable a controlar $\ddot{\phi}_d$ de igual manera que en el caso anterior se calculó usando un control proporcional mostrado en la Ecuación 2.18, donde:

- **k** : valor de la variable en el ciclo actual del programa.
- **$k - 1$** : valor que tuvo la variable en el ciclo anterior del programa.
- **Δt** : tiempo de muestreo del programa (alrededor de 3 milisegundos ver la sección 2.2.3.5).

El objetivo de calcular el *setpoint* de aceleración con la Ecuación 2.18 es que actúe como un término derivativo que evite que la velocidad sobre pase su *setpoint* y alcance valores muy altos y a la vez es una ayuda para alcanzar el *setpoint* de velocidad cuando la velocidad se aleja del mismo.

$$\begin{aligned}\ddot{\phi}_d &= P_{\dot{\phi}} \cdot \frac{\dot{e}_\phi(k) - \dot{e}_\phi(k-1)}{\Delta t} \\ \ddot{\theta}_d &= P_{\dot{\theta}} \cdot \frac{\dot{e}_\theta(k) - \dot{e}_\theta(k-1)}{\Delta t} \\ \ddot{\psi}_d &= P_{\dot{\psi}} \cdot \frac{\dot{e}_\psi(k) - \dot{e}_\psi(k-1)}{\Delta t}\end{aligned}$$

Ecuación 2.18. Cálculo del *setpoint* de aceleración de la variable a controlar.

2.1.2 CONTROLADORES PID

Con el fin de comparar los resultados obtenidos con el control SMC se implementó para los ángulos de orientación un control PID similar al implementado en los autopilotos comerciales [11].

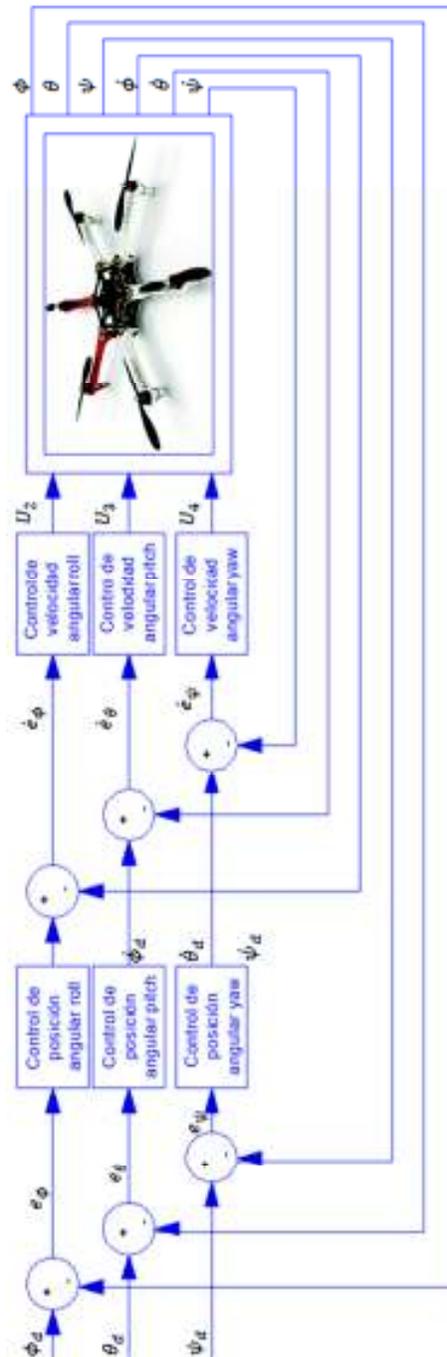


Figura 2.2. Control PID en cascada.

La estructura del control PID en cascada se muestra en la Figura 2.2 en donde el controlador de posición angular (control externo) es un control P y el controlador de velocidad angular (control interno) es un control PID. La salida de control del control de posición angular representa el *setpoint* de velocidad para el control de velocidad angular [11].

Cuando se realiza un vuelo autónomo se requiere de mayor precisión por lo que los autopilotos agregan un control externo más el cual usa datos del GPS y de un barómetro para controlar la posición del dron [11].

2.2 DISEÑO E IMPLEMENTACIÓN DEL SOFTWARE DE CONTROL

A continuación se describirán las funcionalidades del programa realizado para el Arduino DUE en el mismo orden de ejecución del programa. Un diagrama de flujo general del programa implementado se puede observar en la Figura 2.3.



Figura 2.3. Flujograma de software de control.

2.2.1 DECLARACIÓN DE LIBRERÍAS Y VARIABLES

En la primera parte del programa se declaran las variables y las librerías que se usarán, las cuales son:

- **Servo.h:** permite generar señales con ancho de pulso variable para manipular el movimiento de hasta 12 motores de vehículos de radio control.
- **Wire.h** permite realizar la comunicación I²C entre el Arduino DUE y la IMU MPU 6050.

Entre las variables declaradas se encuentran:

- La velocidad de transmisión de datos a través del puerto serial definida en 460800 baudios.
- Se declara los pines a los que se conectará la señal de control de velocidad de los seis motores y se asocia estos pines a un objeto de la librería *Servo.h* para permitir la generación de los pulsos de control.
- El valor máximo y mínimo del ancho del pulso de control que puede ser enviado a los ESCs de los motores durante el vuelo entre 1950 y 1000 microsegundos respectivamente.
- El incremento del pulso de control de velocidad de los motores durante el arranque, definido en 15 microsegundos (este valor se seleccionó por medio de pruebas para lograr un arranque no brusco).
- El valor al que se llevará el pulso base enviado a los ESC al terminar el arranque.
- Las variables que indican cuando: se inició el proceso de arranque, se terminó el arranque, se recibió el comando de arranque, se recibió el comando de parada, se terminó el proceso de arranque y cuando se recibió el comando de descenso.
- Las variables que se usan para almacenar los datos recibidos desde la IMU.
- Las variables que almacenan la estimación de los ángulos de orientación usando el giroscopio, el acelerómetro y el filtro complementario.
- Las variables para los filtros de alisado de Kalman de las medidas del giroscopio.
- Las variables para los filtros de alisado de Kalman de los ángulos de orientación estimados con el acelerómetro, estos filtros sólo se usan previo al arranque para determinar la orientación inicial del hexarotor.
- Las constantes y variables que definen los parámetros de los controladores SMC y las variables usadas en el cálculo de las señales de control.
- Las variables que pueden usarse para agregar *offsets* (desviación constante de una medida del valor real) a las señales de control y a la estimación de roll y pitch usando las medidas del acelerómetro en caso de que se requieran.
- Las variables donde se almacena el valor del ancho del pulso a enviar a los ESC y la variable donde se almacena el cálculo del contra torque del yaw.

- Las variables que permiten: detectar una falla en la comunicación entre el hexarotor y el computador, elegir los datos a enviarse al computador, enviar los datos seleccionados cada cinco ciclos del programa (esto se hizo de forma experimental para evitar que la pérdida de datos enviados al computador), poner el *setpoint* de aceleración presente en los controladores a cero cuando se realizan cambios de *setpoint* o cuando se arranca el hexarotor y limitar el pulso base enviado a los motores durante el arranque.
- Las variables que permiten calcular el tiempo de ejecución del programa (el tiempo de muestreo es de 3 milisegundos, ver la sección 2.2.3.5).
- Las constantes que almacenan el valor de los registros para la comunicación con la IMU y los valores necesarios para el escalamiento de las medidas del giroscopio y el acelerómetro.

2.2.2 FUNCIÓN VOID SETUP

La función *void setup* se ejecuta una sola vez al encender la tarjeta Arduino, por lo que se usa para configuraciones iniciales [38]. En la Figura 2.4 se muestra un flujograma de la función *void setup*.



Figura 2.4. Flujograma de la función *void setup()*.

En esta función se ejecutan las siguientes acciones:

- Se configura el puerto serial a utilizar para la comunicación con el computador usando la velocidad de transmisión previamente definida.
- Se configura la velocidad de la comunicación I²C en 400 kHz.
- Se llama a la función `setupMPU`, la cual se encarga de configurar la IMU con los siguientes parámetros: se desactiva la función de *sleep*, se configura el uso del reloj interno de 8MHz, se define la escala de medición del giroscopio en $\pm 1000^\circ/\text{s}$, la escala de medición del acelerómetro en $\pm 16\text{g}$ y se configura el uso de un filtro pasa bajo con banda de paso de 100Hz para las mediciones del giroscopio y el acelerómetro. Luego de cada configuración se verifica que dicha configuración haya sido almacenada correctamente.
- Se envía a todos los ESCs un pulso de 600 microsegundos para que estén listos para su funcionamiento.
- Se coloca en el *setpoint* de roll, pitch y yaw un valor de cero.
- Se obtiene el valor del *offset* para el giroscopio en sus tres ejes (roll, pitch y yaw) cómo se explicará a continuación.
- Se lee el tiempo en microsegundos que lleva el programa ejecutándose, este tiempo se usa en la estimación de los ángulos de Euler.

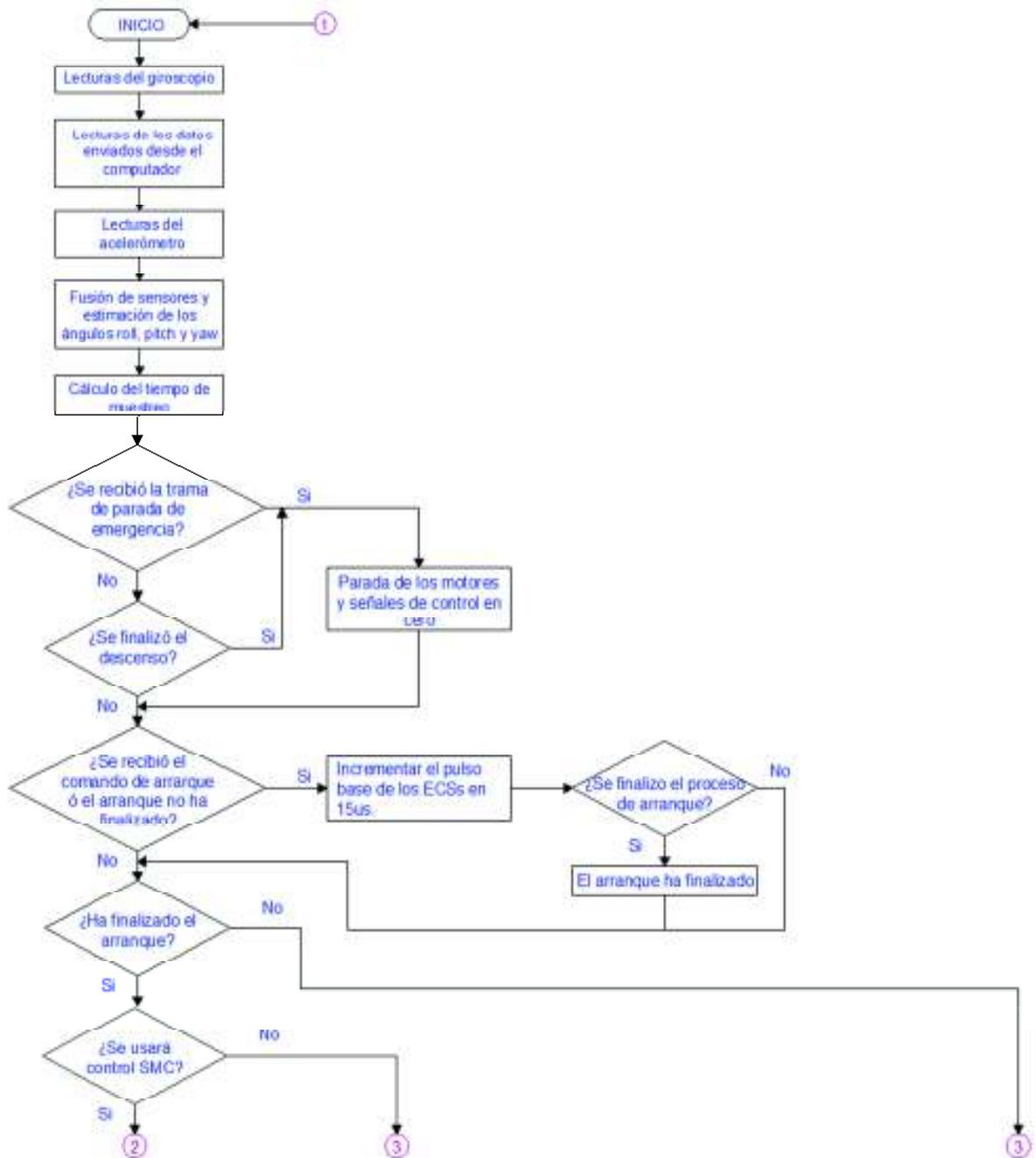
2.2.2.1 **Offset del giroscopio**

Obtener el *offset* del giroscopio es necesario debido a que las mediciones del giroscopio están desplazadas del cero; es decir que, aún en ausencia de movimientos circulares la medida entregada por el giroscopio es diferente de cero (presenta un *offset*).

Por este motivo, antes de poder arrancar el hexarotor se toman 2000 mediciones del giroscopio en intervalos de 5 milisegundos con el hexarotor en reposo. Estas mediciones se promedian para obtener el *offset* del giroscopio para cada eje (roll, pitch y yaw). El *offset* obtenido se resta de todas las futuras mediciones del giroscopio. En la sección 3.1.1.1 se muestra una prueba relacionada con el *offset* del giroscopio.

2.2.3 FUNCIÓN LOOP

A continuación se describen las funciones implementadas en la función *loop* del programa. Todo código que se escriba dentro de esta función se repetirá de forma continua [38]. En la Figura 2.5 se muestra el flujograma de la función *loop*.



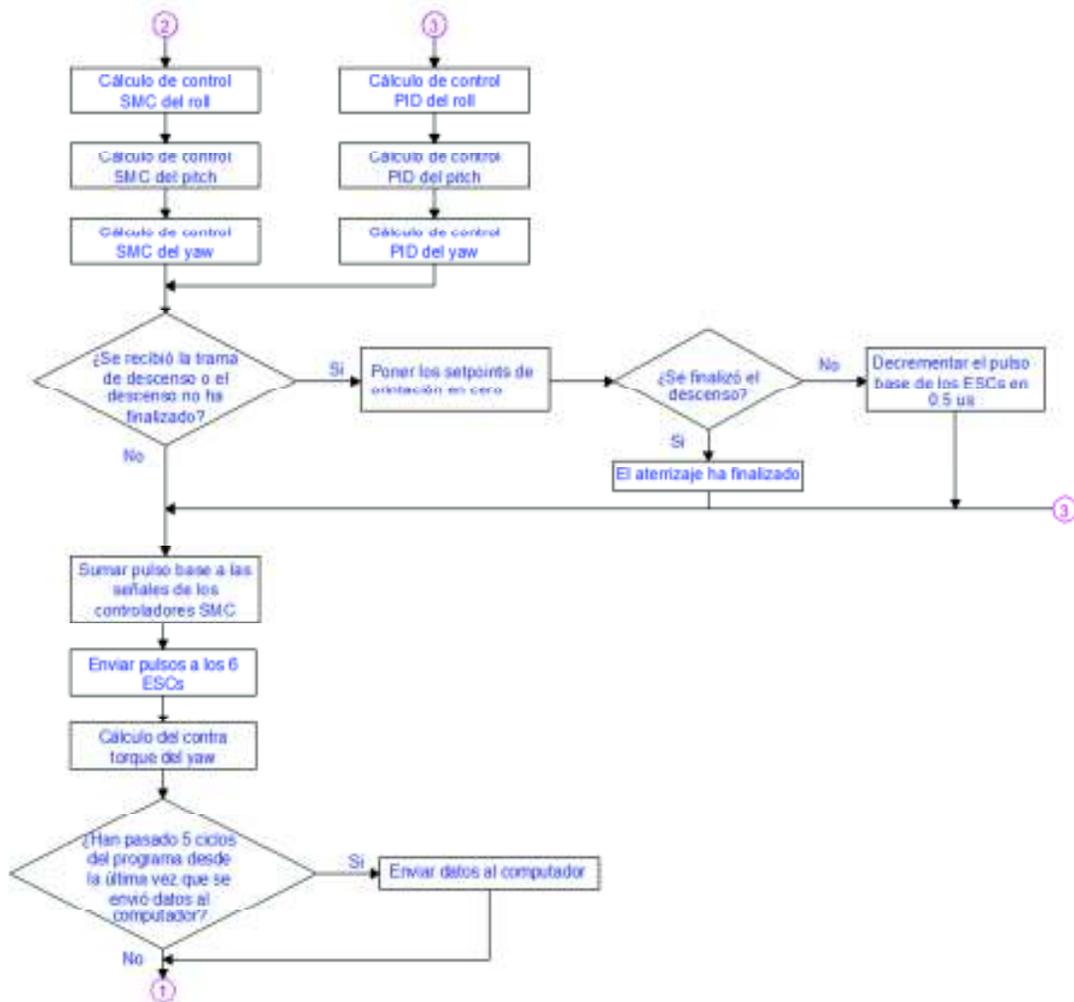


Figura 2.5. Flujoograma de la función loop().

2.2.3.1 Lectura del giroscopio

Se llama a la función `lectura_imu`, la cual solicita y recibe las mediciones del giroscopio para los tres ejes. Antes de usar la medida, ésta es validada con el fin de evitar el uso de datos erróneos, se verifica que la medida recibida no esté fuera del rango de medición configurado y además que ésta sea un dato numérico.

Posteriormente, se resta el *offset* del giroscopio (ver sección 2.2.2.1) obtenido previamente de las mediciones y se aplica el factor de conversión requerido para transformar la medida de grados por segundo a radianes por segundo. Seguidamente la medida en radianes por segundo pasa por un filtro de alisado de Kalman, cuyo propósito es eliminar lo más posible el ruido presente en las mediciones.

Los valores de los filtros de Kalman implementados para los ángulos roll, pitch y yaw son $P = 1$, $Q = 0.01$, $R = 1$ y $\hat{x} = 0$ (ver sección 1.4.6). Las medidas del giroscopio ya filtradas se usarán posteriormente para estimar los ángulos de Euler.

2.2.3.2 Lectura de datos

Se llama a la función `lectura_datos_hmi`, la cual verifica si hay datos en el *buffer* del puerto serial usado para la comunicación con el computador. Si los hay, primero realiza una validación de los datos y si éstos llegaron sin errores.

Todo dato enviado hacia el controlador tiene el formato `# + dato + @`. Los caracteres `#` y `@` se usan para validar la trama de datos recibida y la palabra `dato` hace referencia a un carácter. Cada carácter tiene asignada una función, las cuales se muestran en la Tabla 2.1.

Dentro de la función de lectura de datos se incrementa una variable de control de la comunicación en 1 cada vez que se llama a la función. Cuando esta variable supera el valor de 300 (alrededor de 1 segundo) se inicia la secuencia de descenso debido a la pérdida de la comunicación con el computador, esto es importante debido a que el computador envía los comandos de maniobra que ejecuta el usuario y por tanto es una medida de seguridad para evitar accidentes. La variable de control de la comunicación se pone en cero cada vez que se recibe el carácter `l` (ver Tabla 2.1).

Tabla 2.1. Caracteres usados en la comunicación.

Letra	Función
a	Es la orden de arrancar el hexarotor y pone el valor adecuado en la variable que provoca que el <i>setpoint</i> de aceleración presente en los controladores SMC de roll y pitch tome un valor de cero sólo en ese ciclo del programa para evitar un comportamiento brusco de esta parte derivativa debido al cambio brusco del error inicial.
b	Es la orden de parada del hexarotor.
c	Es la orden de descenso del dron.
d	Incrementa el pulso base enviado a los ESC en 5 microsegundos, se usa para variar la altura del hexarotor durante el vuelo.

e	Decrementa el pulso base enviado a los ESC en 3 microsegundos, se usa para variar la altura del hexarotor durante el vuelo.
f	Incrementa el <i>setpoint</i> del ángulo roll en 1 grado, el valor máximo del <i>setpoint</i> se limita a 2 grados y pone el valor adecuado en la variable que provoca que el <i>setpoint</i> de aceleración presente en el controlador SMC del ángulo roll tome un valor de cero sólo en ese ciclo del programa.
g	Decrementa el <i>setpoint</i> del ángulo roll en 1 grado, el valor mínimo del <i>setpoint</i> se limita a -2 grados y pone el valor adecuado en la variable que provoca que el <i>setpoint</i> de aceleración presente en el controlador SMC del ángulo roll tome un valor de cero sólo en ese ciclo del programa.
h	Incrementa el <i>setpoint</i> del ángulo pitch en 1 grado, el valor máximo del <i>setpoint</i> se limita a 2 grados y pone el valor adecuado en la variable que provoca que el <i>setpoint</i> de aceleración presente en el controlador SMC del ángulo pitch tome un valor de cero sólo en ese ciclo del programa.
i	Decrementa el <i>setpoint</i> del ángulo pitch en 1 grado, el valor máximo del <i>setpoint</i> se limita a -2 grados y pone el valor adecuado en la variable que provoca que el <i>setpoint</i> de aceleración presente en el controlador SMC del ángulo pitch tome un valor de cero sólo en ese ciclo del programa.
j	Coloca en cero el valor del <i>setpoint</i> del ángulo roll y pone el valor adecuado en la variable que provoca que el <i>setpoint</i> de aceleración presente en el controlador SMC del ángulo roll tome un valor de cero sólo en ese ciclo del programa.
k	Coloca en cero el valor del <i>setpoint</i> del ángulo pitch y pone el valor adecuado en la variable que provoca que el <i>setpoint</i> de aceleración presente en el controlador SMC del ángulo pitch tome un valor de cero sólo en ese ciclo del programa.
l	Este carácter se envía cada 0.233 segundos aproximadamente desde el computador y se usa para comprobar que la comunicación siga hábil.
m	Decrementa el <i>setpoint</i> del ángulo yaw en 22.5 grados, el valor máximo del <i>setpoint</i> se limita a 172 grados y pone el valor adecuado en la variable que provoca que el <i>setpoint</i> de aceleración presente en el controlador SMC del ángulo yaw tome un valor de cero sólo en ese ciclo del programa.
n	Incrementa el <i>setpoint</i> del ángulo yaw en 22.5 grados, el valor máximo del <i>setpoint</i> se limita a -172 grados y pone el valor adecuado en la variable que provoca que el <i>setpoint</i> de aceleración presente en el controlador SMC del ángulo yaw tome un valor de cero sólo en ese ciclo del programa.
o	Coloca en cero el valor del <i>setpoint</i> del ángulo yaw y pone el valor adecuado en la

variable que provoca que el *setpoint* de aceleración presente en el controlador SMC del ángulo yaw tome un valor de cero sólo en ese ciclo del programa.

2.2.3.3 Lectura del acelerómetro

Se llama a la función `lectura_euler`, la cual solicita y recibe las mediciones del acelerómetro para los tres ejes. Antes de usar la medida ésta es validada con el fin de evitar el uso de datos erróneos, se verifica que la medida recibida no esté fuera del rango de medición configurado y que ésta sea un dato numérico.

Posteriormente, se estiman los ángulos del roll y pitch con los datos recibidos del acelerómetro. Para ello se calcula el módulo de las mediciones con la Ecuación 2.19, seguidamente para encontrar el ángulo roll estimado con el acelerómetro se usa la Ecuación 2.20 y para el ángulo pitch estimado con el acelerómetro se usa la Ecuación 2.21.

En las ecuaciones antes mencionadas los términos a usar son los siguientes [31]:

- A_{cx} : Medida del acelerómetro en el eje x .
- A_{cy} : Medida del acelerómetro en el eje y .
- A_{cz} : Medida del acelerómetro en el eje z .
- $\hat{\phi}_a$: Ángulo roll estimado con el acelerómetro.
- $\hat{\theta}_a$: Ángulo pitch estimado con el acelerómetro.

Luego de calcular el módulo y previo a estimar los ángulos de roll y pitch el programa verifica que el valor del módulo sea superior a las medidas del acelerómetro para los ejes x y y con el fin de evitar errores en el cálculo de la función arcoseno.

$$\text{módulo} = \sqrt{A_{cx}^2 + A_{cy}^2 + A_{cz}^2}$$

Ecuación 2.19. Calculo del módulo de las mediciones del acelerómetro.

$$\hat{\phi}_a = \text{arcoseno}\left(\frac{A_{cy}}{\text{módulo}}\right)$$

Ecuación 2.20. Estimación del ángulo roll con el acelerómetro.

$$\hat{\theta}_a = \arccos\left(\frac{Acx}{\text{módulo}}\right)$$

Ecuación 2.21. Estimación del ángulo pitch con el acelerómetro.

Antes del arranque las estimaciones de los ángulos de roll y pitch pasan por un filtro de alisado de Kalman para minimizar el ruido que aparece en las mediciones del acelerómetro. Esto se realiza sólo antes del arranque debido a que en ese intervalo de tiempo los ángulos de Euler del hexarotor se calculan únicamente con las medidas del acelerómetro y esas medidas serán el valor inicial que tendrán los ángulos de roll y pitch antes del arranque. Los valores de los filtros de Kalman implementados se muestran en la Tabla 2.2.

Tabla 2.2. Valores del filtro de Kalman para la medida del acelerómetro.

Acelerómetro	P	Q	R	\hat{x}
Ángulo roll	2	0.1	2	0
Ángulo pitch	2	0.1	5	0

2.2.3.4 Fusión de sensores

Para la fusión de sensores se requiere de los ángulos de Euler estimados con el giroscopio y los estimados con el acelerómetro. El ángulo yaw sólo se estima con el giroscopio (ver Ecuación 2.22) y no con fusión de sensores; porque con el acelerómetro no es posible estimar el ángulo yaw.

La estimación de los ángulos orientación con los datos del giroscopio previamente filtrados con un filtro de alisado de Kalman se realiza mediante la integración discreta de las medidas del giroscopio utilizando una aproximación por áreas rectangulares como se muestra en la Ecuación 2.22, dónde:

- $\hat{\phi}_g, \hat{\theta}_g, \hat{\psi}_g$: ángulo roll, pitch y yaw estimado con el giroscopio.
- $m_{g\phi}, m_{g\theta}, m_{g\psi}$: medida del giroscopio para roll, pitch y yaw respectivamente.
- Δt : tiempo de muestreo del programa (3 milisegundos, ver sección 2.2.3.5).
- k : valor de la variable en la actual ejecución del programa
- $k - 1$: valor de la variable en la ejecución anterior del programa.

$$\hat{\phi}_g(k) = \hat{\phi}_g(k-1) + m_{g\phi} \cdot \Delta t$$

$$\hat{\theta}_g(k) = \hat{\theta}_g(k-1) + m_{g\theta} \cdot \Delta t$$

$$\hat{\psi}_g(k) = \hat{\psi}_g(k-1) + m_{g\psi} \cdot \Delta t$$

Ecuación 2.22. Estimación del ángulo con el giroscopio.

Posteriormente, se compensan las estimaciones con el giroscopio de los ángulos de roll y pitch cuando ocurre un giro en yaw mediante la Ecuación 2.23 y Ecuación 2.24, donde [39]:

- Δt : Tiempo de muestreo del programa (alrededor de 3 milisegundos ver la sección 2.2.3.5).
- $\hat{\phi}_g$: Ángulo roll estimado con el giroscopio.
- $\hat{\theta}_g$: Ángulo pitch estimado con el giroscopio.
- $\hat{\psi}$: Medida del giroscopio del yaw luego de ser filtrada por el filtro de Kalman.
- $\hat{\phi}_{gc}$: Ángulo roll estimado con el giroscopio y compensado el giro del yaw.
- $\hat{\theta}_{gc}$: Ángulo pitch estimado con el giroscopio y compensado el giro del yaw.

$$\hat{\phi}_{gc} = \hat{\phi}_g + \hat{\theta}_g \cdot \sin(\hat{\psi} \cdot \Delta t)$$

Ecuación 2.23. Compensación del giro del yaw en el ángulo roll.

$$\hat{\theta}_{gc} = \hat{\theta}_g - \hat{\phi}_g \cdot \sin(\hat{\psi} \cdot \Delta t)$$

Ecuación 2.24. Compensación del giro del yaw en el ángulo pitch.

El filtro complementario utilizado para la estimación de los ángulos de roll y pitch se muestra en la Ecuación 2.25, donde:

- $\hat{\phi}_{gc}, \hat{\theta}_{gc}$: ángulo estimado con el giroscopio y compensado el giro del yaw.
- f : factor de confiabilidad de las estimaciones, su valor está entre 0 y 1.
- $\hat{\phi}_a, \hat{\theta}_a$: ángulo estimado con el acelerómetro.

$$\hat{\phi} = \hat{\phi}_{gc} \cdot f + \hat{\phi}_a \cdot (1 - f)$$

$$\hat{\theta} = \hat{\theta}_{gc} \cdot f + \hat{\theta}_a \cdot (1 - f)$$

Ecuación 2.25. Fusión de sensores para estimar los ángulos roll y pitch.

El factor de confiabilidad usado para la estimación de los ángulos de orientación roll y pitch fue de 0.9996.

2.2.3.5 Tiempo de muestreo

Se calcula el tiempo total del lazo mediante la diferencia entre el tiempo final de lazo medido luego de leer los datos de la IMU (giroscopio y acelerómetro) y el tiempo inicial del lazo medido luego de la estimación de los ángulos de Euler en el ciclo anterior del programa. De esta manera se tiene el tiempo transcurrido desde la última estimación de los ángulos de orientación, lo que equivale al tiempo de muestreo del programa.

El tiempo de muestreo de un autopiloto comercial es 10 milisegundos o 100Hz para el autopiloto ArduPilot Mega y de 2.5 milisegundos o 400 Hz para el autopiloto Pixhawk [40]. El tiempo de muestreo del programa de control (autopiloto) desarrollado en el Arduino DUE es de 3 milisegundos, lo que equivale a alrededor de 333 Hz o ciclos de ejecución del programa por segundo.

El tiempo de muestreo del programa varía en función de los pulsos de control enviados a los ESCs. Al recibir los datos de vuelo se observó que al enviar el pulso máximo a los ESCs el tiempo de muestreo no superaba los 3 milisegundos; por esta razón se fijó por software el tiempo de muestreo del programa en este valor. El tiempo de ejecución del programa se mide usando un temporizador del Arduino DUE.

2.2.3.6 Parada del hexarotor

Esta parte del programa permite parar al hexarotor en cualquier momento, esto se usa ya sea para apagar los motores luego de un descenso o como una parada de emergencia. Cuando se reciba el comando de parada desde el computador en esta parte del código se pone en cero las señales de control de todos los controladores y se envía a los ESC un pulso de 600 microsegundos para apagar los motores. Luego de una parada el hexarotor puede arrancarse otra vez al reiniciar el programa de MATLAB que permite enviar los comandos al drone para que se pueda volver a enviar un comando de arranque.

2.2.3.7 Orientación inicial

Esta parte del código sólo se usa en el tiempo previo al arranque, porque su función es colocar los valores iniciales de los ángulos de orientación. Para ello los ángulos roll y pitch toman el valor estimado sólo con el acelerómetro, debido a que éste es más confiable cuando no hay vibraciones. El ángulo yaw toma el valor de cero antes del arranque porque éste será siempre su valor inicial.

2.2.3.8 Secuencia de arranque

Cuando se reciba el comando de arranque en esta parte del programa se irá incrementando el pulso base enviado a los ESC en pasos de 15 microsegundos hasta alcanzar el valor máximo definido por el usuario en la sección de declaración de variables. Una vez terminado el proceso de arranque ya no se ejecuta esta parte del programa hasta que el hexarotor haya descendido o parado y se requiera de un nuevo arranque.

2.2.3.9 Controladores SMC

En esta parte del programa se encuentran los controladores SMC para roll, pitch y yaw en ese orden. Los cálculos requeridos para obtener las señales de control son los siguientes:

- Obtener el error del ángulo (ver Ecuación 2.4).
- Calcular la integral del error del ángulo y limitar su valor hasta $\pm\delta$.
- Calcular el *setpoint* de velocidad (ver Ecuación 2.17)
- Obtener el error de velocidad.
- Calcular la superficie deslizante (ver Ecuación 2.3).
- Calcular el *setpoint* de aceleración (ver Ecuación 2.18), si ésta debe ponerse en cero debido a un cambio de *setpoint* o que el hexarotor empezó a arrancar se escribe el valor de cero directamente.
- Calcular la acción de control equivalente y la acción de control discontinuo para obtener la ley de control. Para roll, pitch y yaw se usa la Ecuación 2.8, Ecuación 2.15 y Ecuación 2.16 respectivamente.

2.2.3.10 Descenso del hexarotor

Este código pone en cero los *setpoints* de roll y pitch y va decrementando poco a poco el pulso base enviado a los ESC hasta que éste llega a un valor de 1000 microsegundos. En ese momento las señales de control se ponen en cero y se envía a los ESC un pulso de 600 microsegundos para apagar los motores. Este proceso de descenso puede ser ejecutado por el usuario, pero su uso principal es para descender el dron cuando ocurre una pérdida de comunicación con el computador.

2.2.3.11 Envío de pulsos a los ESC

Esta parte del programa suma el pulso base y las señales de control para obtener el ancho del pulso que se debe enviar a los ESC (ver Ecuación 1.17). Posteriormente se limita este pulso al valor máximo y mínimo ingresado en la sección de declaración de variables pero sólo durante el vuelo y no durante el arranque, descenso y parada del dron.

Además, en esta parte del programa se calcula el contra torque del yaw (ver Ecuación 1.15) usando para ello la Ecuación 2.26, donde ω_i es el ancho del pulso a enviar al motor “i” en microsegundos; esta ecuación la constante; cuyo valor es de 68.8158, la cual permite pasar de microsegundos a *rad/s* las velocidades de los motores.

$$\omega = 68.8158 \cdot (-\omega_1 + \omega_2 - \omega_3 + \omega_4 - \omega_5 + \omega_6) [rad/s]$$

Ecuación 2.26. Cálculo del contra torque del yaw [10].

2.2.3.12 Envío de datos al computador

En la sección de declaración de variables el usuario puede escoger entre 5 opciones diferentes para seleccionar los datos que se enviarán al computador. Los datos son enviados cada 5 ciclos del programa para evitar que el buffer del computador se sature y se pierdan tramas de datos. Las opciones de conjuntos de datos que se pueden escoger son:

- **Datos ángulo roll:** se envían del ángulo roll los datos de: estimación del ángulo, velocidad del ángulo, *setpoint* del ángulo, valor de la superficie deslizante para el

control SMC, valor de la acción de control equivalente, valor de la acción de control discontinuo y el *setpoint* de velocidad.

- **Datos ángulo pitch:** se envían los mismos datos descritos para datos ángulo roll pero para el ángulo pitch.
- **Datos ángulo yaw:** se envían los mismos datos descritos para datos ángulo roll pero para el ángulo yaw.
- **Datos ángulos de orientación:** Se envían los datos de: estimación de los ángulos de Euler (roll, pitch y yaw), *setpoints* angulares de roll y pitch, valor del pulso base enviado a los ESC y el tiempo de muestreo del programa.
- **Datos de la IMU:** Se envían los datos de: estimación de los ángulos de orientación (roll, pitch y yaw), velocidades angular de roll, pitch y yaw, ángulos roll y pitch estimados con el acelerómetro, módulo de las mediciones del acelerómetro, mediciones del acelerómetro en los 3 ejes, tiempo de muestreo del programa y el valor del pulso base enviado a los ESC.

De igual manera que en la lectura de datos los caracteres # y @ se colocan al inicio y al final de la trama respectivamente y se usan para validar la trama de datos recibida. Los datos se separan entre sí usando una coma.

2.2.4 SOFTWARE EN EL COMPUTADOR

En el software MATLAB se desarrolló un algoritmo que permite establecer la comunicación con el Arduino DUE, recibir datos desde el controlador, enviar comandos de vuelo al hexarotor y validar las tramas de datos recibidas. De igual manera se desarrollaron algoritmos para presentar de forma gráfica los distintos grupos de datos que se reciben durante el vuelo. Los programas desarrollados y su funcionalidad se detallan brevemente a continuación.







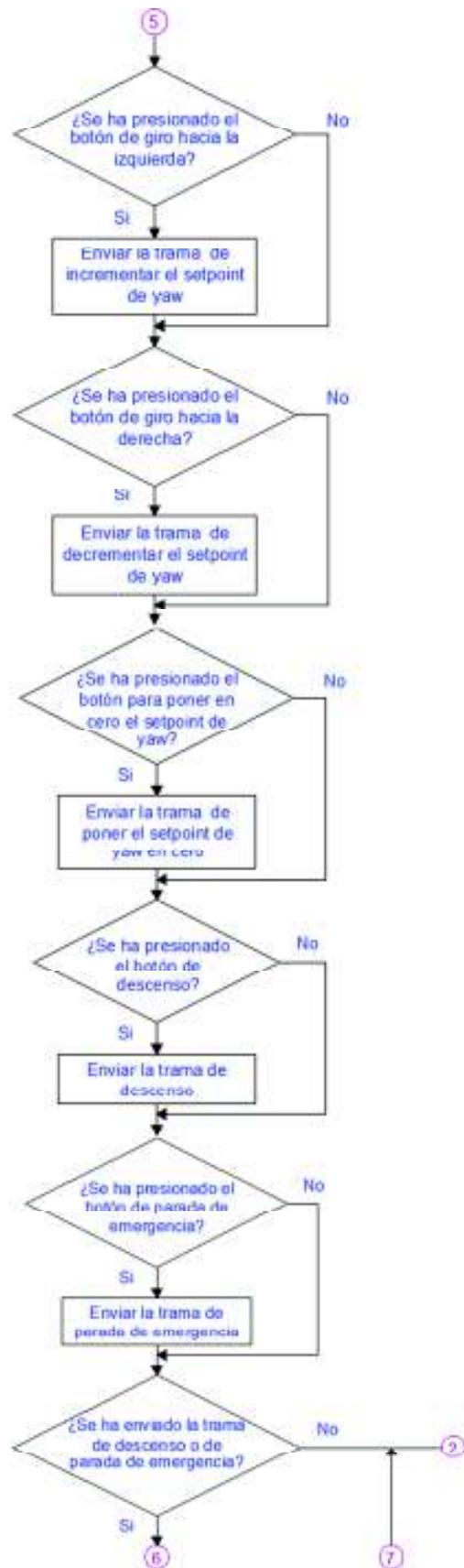




Figura 2.6. Flujograma del programa lee_com.

lee_com: Este programa permite enviar y recibir tramas de datos desde y hacia la tarjeta Arduino Due, sus funcionalidades se describen a continuación. En la Figura 2.6 se detalla un flujograma de este programa. Permite la conexión con el conversor USB a UART creando para ello un objeto comPort en el que se especifican los bits de parada (1 bit), los bits de datos (8 bits), la velocidad de transmisión (460800 baudios) y la paridad (ninguna); de esta forma se pueden enviar y recibir tramas de datos.

Luego de descender o parar al hexarotor es recomendable esperar unos segundos antes de presionar el botón de *stop* para que el programa termine de leer todas las tramas de datos almacenadas en el buffer.

Crea un objeto tipo *joystick* el cual permite conectar el control de manejo a MATLAB y leer el estado de sus botones, de esta manera se envía la trama de datos requerida al Arduino DUE para que se ejecuten las órdenes del usuario. Cada 0.23 segundos se envía la trama correspondiente al aviso de que la comunicación sigue activa (ver Tabla 2.1). Finalmente, almacena todas las tramas de datos recibidas conjuntamente con un vector que contiene el tiempo en el que fueron leídas para poder graficarlas posteriormente.

sep: Separa las tramas de datos recibidas usando el programa “lee_com”, las analiza una por una para descartar las tramas que contengan errores tales como: no se encuentran los caracteres delimitantes de trama # y @, la trama contiene caracteres no numéricos o la trama contiene un número de datos diferente al esperado. Este programa se debe usar si se escogió como opción de conjunto de datos: datos ángulo roll, datos ángulo pitch, datos ángulo yaw o datos ángulos de orientación (ver sección 2.2.3.12), porque todas estas opciones envían un total de 7 datos.

sep_IMU: Separa las tramas de datos recibidas usando el programa “lee_com”, las analiza una por una para descartar las tramas que contengan errores tales como: no se

encuentran los caracteres delimitantes de trama # y @, la trama contiene caracteres no numéricos o la trama contiene un número de datos diferente al esperado. Este programa se debe usar sólo si se escogió como opción de conjunto de datos “datos de la IMU” (ver sección 2.2.3.12), ya que esta opción envía un total de 14 datos.

graf: Grafica los datos recibidos cuando la opción de conjunto de datos escogida fue: datos ángulo roll, datos ángulo pitch o datos ángulo yaw. El usuario tiene la opción de escoger si se grafica el conjunto de datos en su totalidad o si sólo se grafica un tramo.

graf_rpy: Grafica los datos recibidos cuando la opción de conjunto de datos escogida fue datos ángulos de orientación. El usuario tiene la opción de escoger si se grafica el conjunto de datos en su totalidad o si sólo se grafica un tramo.

graf_IMU: Grafica los datos recibidos cuando la opción de conjunto de datos escogida fue datos de la IMU. El usuario tiene la opción de escoger si se grafica el conjunto de datos en su totalidad o si sólo se grafica un tramo.

2.3 PROCEDIMIENTO DE SINTONIZACIÓN DE LOS CONTROLADORES SMC

En la Figura 2.7 se muestra el esquema del control SMC implementado en el hexarotor.

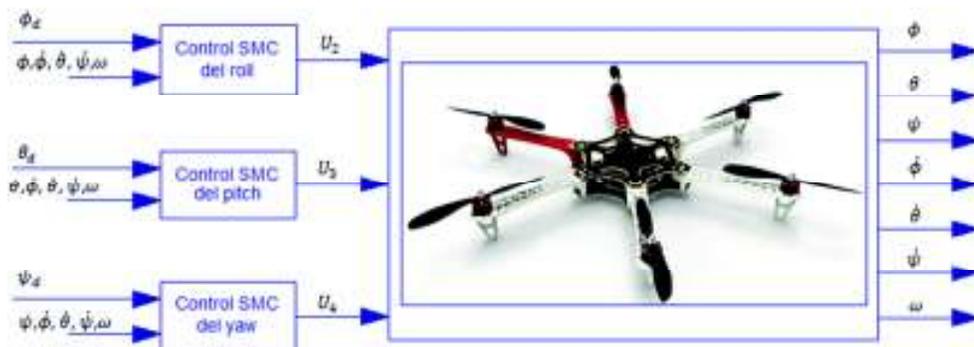


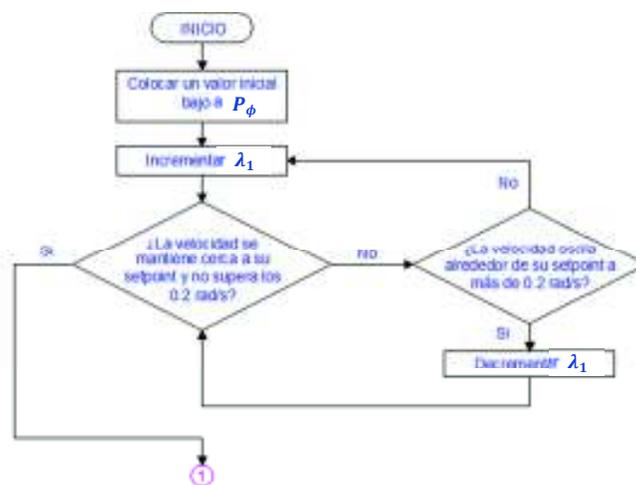
Figura 2.7. Esquema del control SMC implementado.

El hexarotor al presentar un acoplamiento no lineal entre todos sus grados de libertad complica la sintonización de los controladores. La mejor forma de sintonizar los controladores es realizando pruebas de vuelo, pero esto es muy difícil al principio debido

a que no se sabe con certeza el valor de los parámetros de los controladores. En la Figura 2.8 se muestra un flujograma más detallado del procedimiento propuesto para la sintonización de los controladores SMC de orientación usando como ejemplo el ángulo roll.

De forma general el procedimiento de sintonización para los controladores SMC de orientación usando como ejemplo el ángulo roll es el mostrado en el flujograma de la Figura 2.9, cuyos pasos son:

- Se coloca un valor inicial bajo a P_ϕ , para este parámetro se tuvo en cuenta que en [41] se recomienda que como máximo el *setpoint* de velocidad sea de 4.5°/s por cada 1° de error.
- Se encuentra un valor de λ_1 que permita que la velocidad esté cerca de su *setpoint* y que la misma no supere los 0.2 rad/s.
- Encontrar el valor de P_ϕ que permite a la velocidad alcanzar su *setpoint* en menos de 0.3 segundos.
- Colocar un valor alto en K_d , el valor de este parámetro debe permitir a la a fase de alcanzabilidad compensar las perturbaciones.
- Encontrar el valor de δ que permite al ángulo alcanzar su *setpoint* en menos de 1 segundo.
- Se encuentra un valor de λ_2 que permita al ángulo mantener su *setpoint* con velocidades menores a 0.2 rad/s.
- Encontrar el valor de P_ϕ que permita alcanzar el *setpoint* angular en menos de 1 segundo.



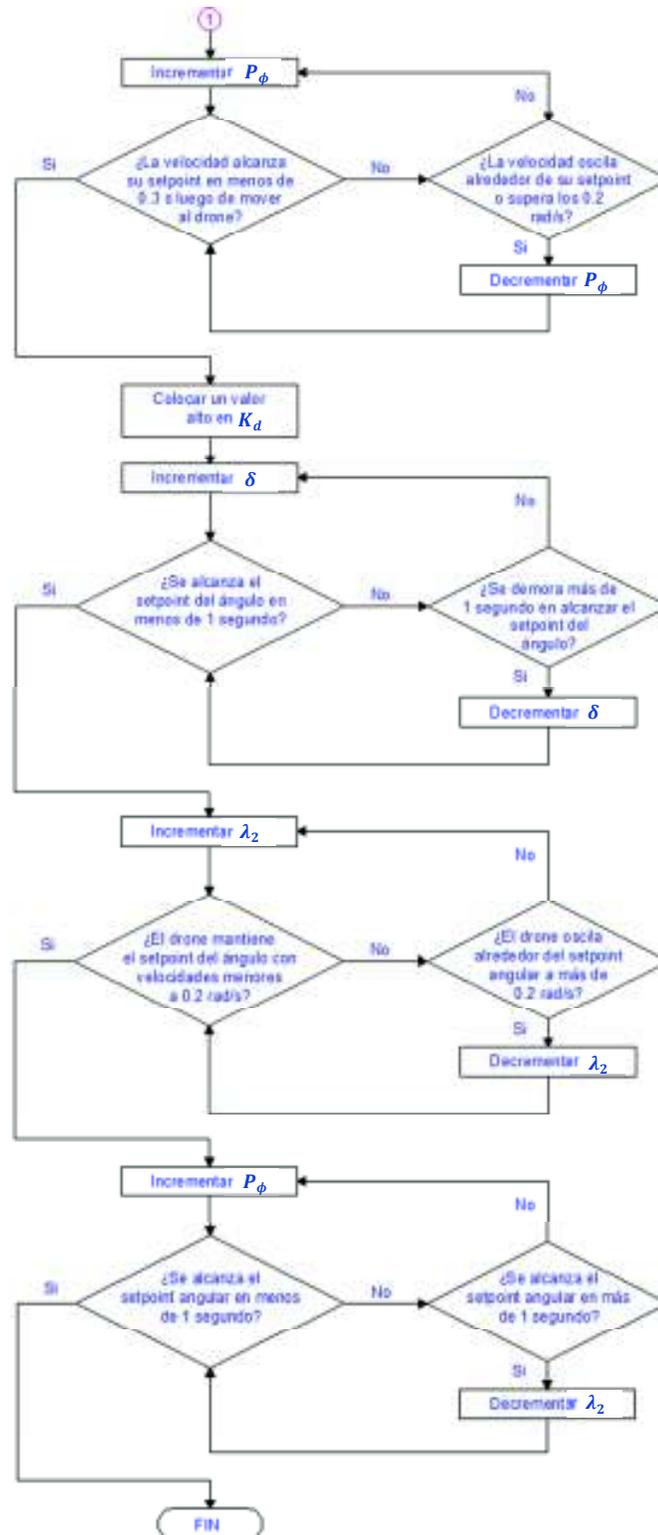


Figura 2.8. Flujo de trabajo del procedimiento de sintonización para el control SMC.

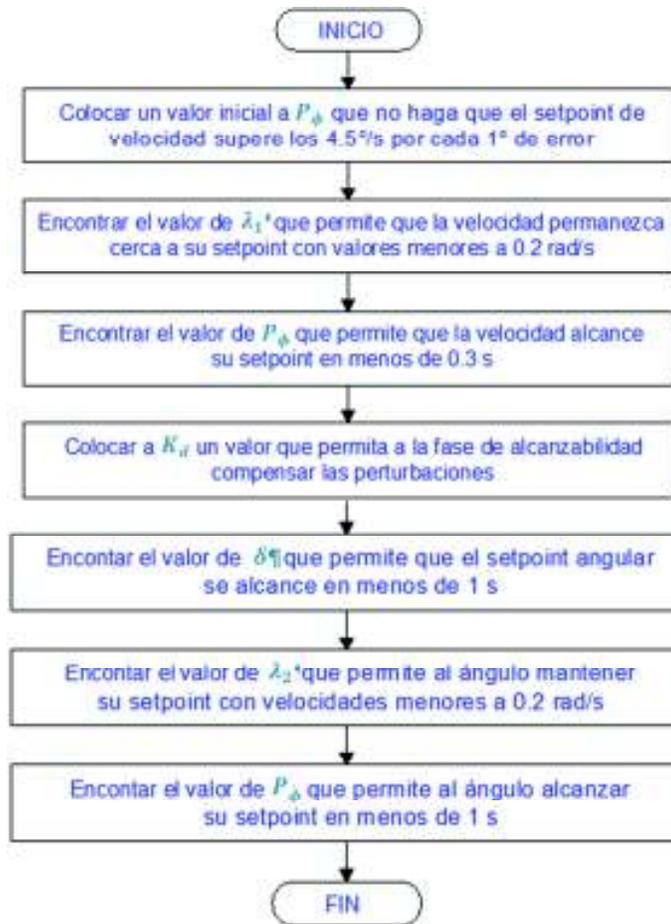


Figura 2.9. Procedimiento de sintonización del control SMC.

2.3.1 SINTONIZACIÓN DE P_ϕ Y λ_1

Para poder sintonizar los controladores primeramente se empezó colocando un valor a P_ϕ (ver Ecuación 2.17) para tener un *setpoint* de velocidad que ayude al controlador a llegar cerca del *setpoint* del ángulo, el cual siempre se mantuvo en cero. El valor inicial escogido para P_ϕ fue de 4 debido a que en [41] se recomienda que como máximo el *setpoint* de velocidad sea de $4.5^\circ/\text{s}$ por cada 1° de error.

También se colocó un valor a λ_1 (ver Ecuación 2.8, Ecuación 2.15 y Ecuación 2.16), debido a que en la acción de control equivalente este parámetro permite corregir el error de velocidad. Todos los demás parámetros de las leyes de control se pusieron en cero, el propósito de esta primera sintonización es encontrar un valor de λ_1 que mantenga a la velocidad lo más cercana posible a su *setpoint*.

Las pruebas para encontrar el valor de P_ϕ y λ_1 se realizan ángulo por ángulo, es decir que si se va a empezar a sintonizar el ángulo roll, todos los parámetros de los controladores de pitch y yaw deben ponerse a cero para que no intervengan en la sintonización.

Además se debe tener siempre un pulso base a enviar a los ESC (U_1 en la Ecuación 1.17), el valor de este pulso base fue de 1600 microsegundos, valor que equivale al pulso que se debe enviar a los ESC para mantener al drone a una altura de alrededor de 1.8 metros.

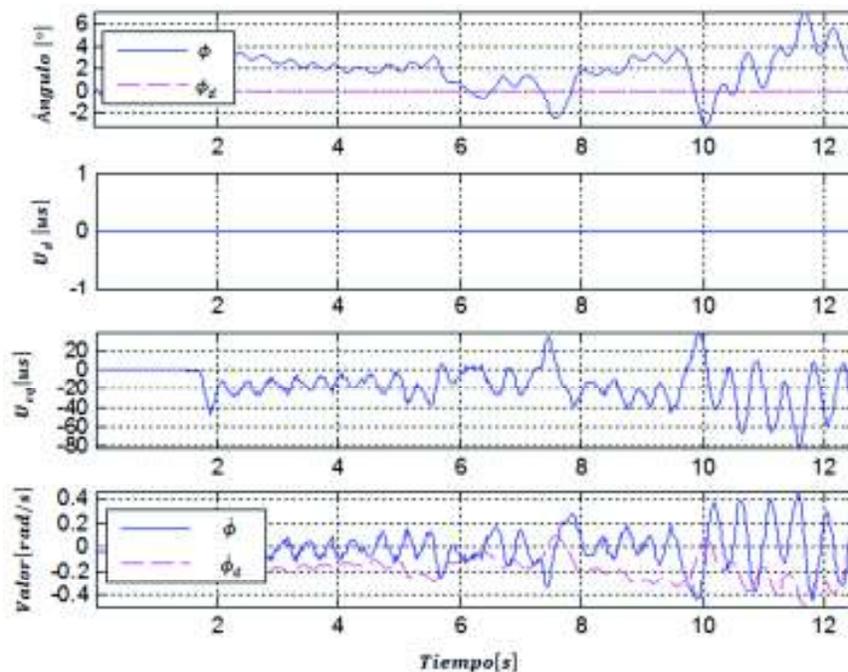


Figura 2.10. Resultados con $P_\phi = 4$ y $\lambda_1 = 50$ (valor alto).

Para la sintonización inicial de los parámetros P_ϕ y λ_1 se debe tener en cuenta las siguientes consideraciones:

- Se debe sostener al drone con las manos cuidando de darle libertad de giro sólo en el eje del ángulo a sintonizar para sentir con las manos si éste presenta una oposición firme a cualquier movimiento.
- Primero se empieza colocando un valor bajo a P_ϕ el cual se mantendrá por el momento constante y se variará el valor de λ_1 . Lo que se desea es que P_ϕ no tenga al principio mucha influencia y que sea sólo una pequeña ayuda para encontrar el valor de λ_1 que logra mantener a la velocidad en su *setpoint*.

- Si el valor de λ_1 es muy bajo se podrá sentir que el drone tiene problemas para mantener una posición y que tiende a girar sin mostrar señales de que el control se opone al movimiento. Un valor alto de λ_1 (ver Figura 2.10) hará que se sientan oscilaciones alrededor de la posición deseada, incluso si es muy alta se presentan oscilaciones demasiado bruscas. Un valor adecuado de λ_1 (ver Figura 2.11) será aquel que presente una firme oposición ante cualquier intento de variar la posición y que al mismo tiempo no presente oscilaciones muy bruscas o visibles para mantener el *setpoint* angular o de velocidad.
- El valor de P_ϕ determina la rapidez con la cual el controlador intentará corregir el error en la posición angular. Por lo tanto, para sintonizarlo se debe realizar pequeñas giros al drone sobre el eje que se está sintonizando con el fin de sentir si al tratar de regresar la velocidad a su *setpoint* el controlador presenta una respuesta rápida y una oposición firme a cualquier intento de rotar el drone.
- Un valor muy pequeño de P_ϕ se sentirá como si el drone tratara de regresar muy lentamente a la posición anterior una vez que se le ha movido. Un valor de P_ϕ demasiado alto provocará la aparición de oscilaciones (ver Figura 2.12) aun cuando no se le gire. Un valor adecuado de P_ϕ (ver Figura 2.13 es aquel que presenta firmeza ante cualquier intento de giro y que a su vez regresa al drone a su posición en un tiempo corto cuando se le ha girado.

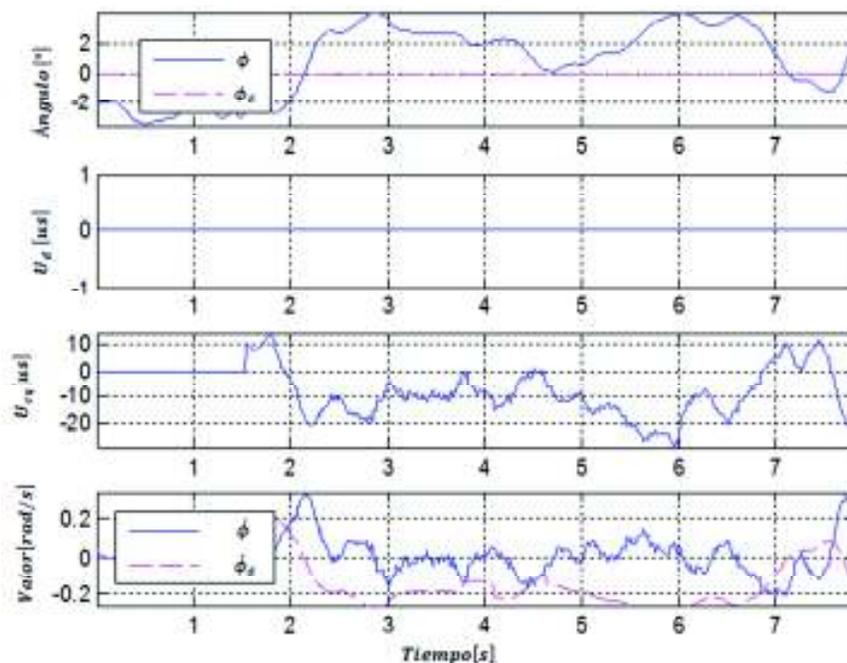


Figura 2.11. Resultados con $P_\phi = 4$ y $\lambda_1 = 30$ (valor adecuado).

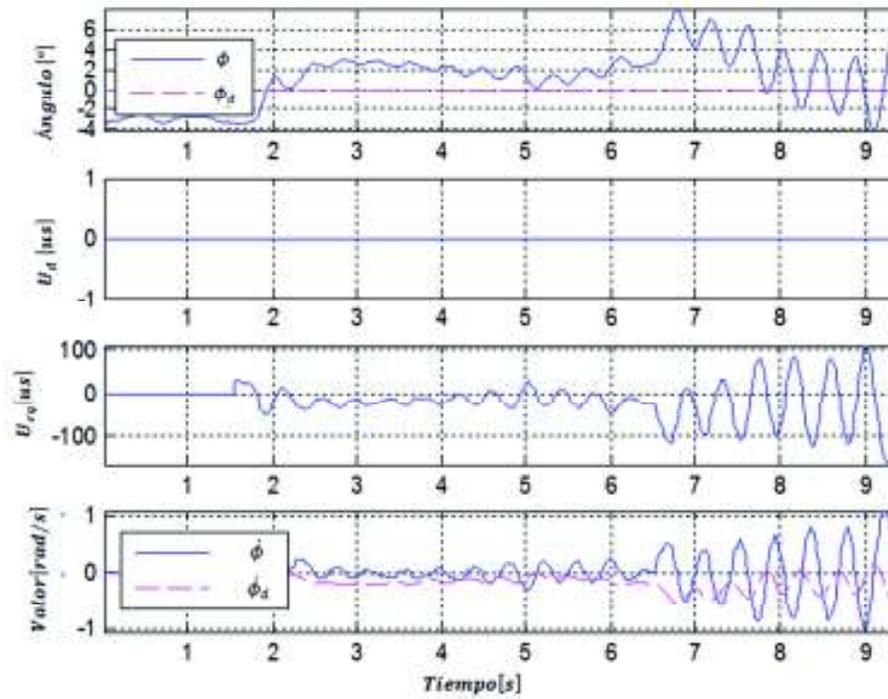


Figura 2.12. Resultados con $\lambda_1 = 30$ y $P_\phi = 4.5$ (valor alto).

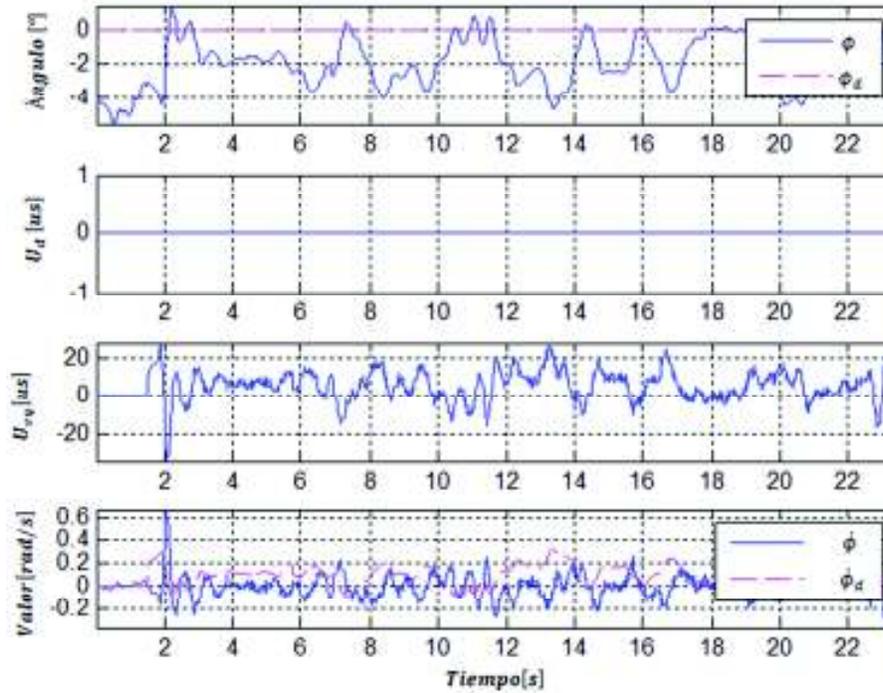


Figura 2.13. Resultados con $\lambda_1 = 30$ y $P_\phi = 4$ (valor adecuado).

2.3.2 SINTONIZACIÓN DE K_d Y δ

Una vez realizado el procedimiento anterior en los ángulos roll, pitch y yaw el hexarotor podrá mantenerse un tiempo en el aire si se pone a funcionar todos los controladores al mismo tiempo. Claro que al no tener todavía los demás parámetros de los controladores sintonizados se observará que dependiendo de su posición inicial presentará problemas para alcanzar el *setpoint* angular con rapidez. En este punto se sintonizará el parámetro K_d (ver Ecuación 2.8, Ecuación 2.15 y Ecuación 2.16) y δ (ver Ecuación 1.36) de la acción de control discontinuo.

Como primer paso se define un valor para K_d , este valor de preferencia tendrá un valor alto y no se modificará durante el proceso de sintonización de K_d y δ . Esto se debe a que el parámetro K_d representa el valor máximo que podrá tomar la acción de control discontinuo y como se puede ver en la Ecuación 1.36 y la Ecuación 2.3 el valor de la integral del error del ángulo presente en la superficie afectará directamente al valor de la acción de control discontinuo. Por lo tanto para que la parte integral pueda compensar cualquier perturbación, especialmente las corrientes de aire, es necesario que K_d tenga un valor elevado.

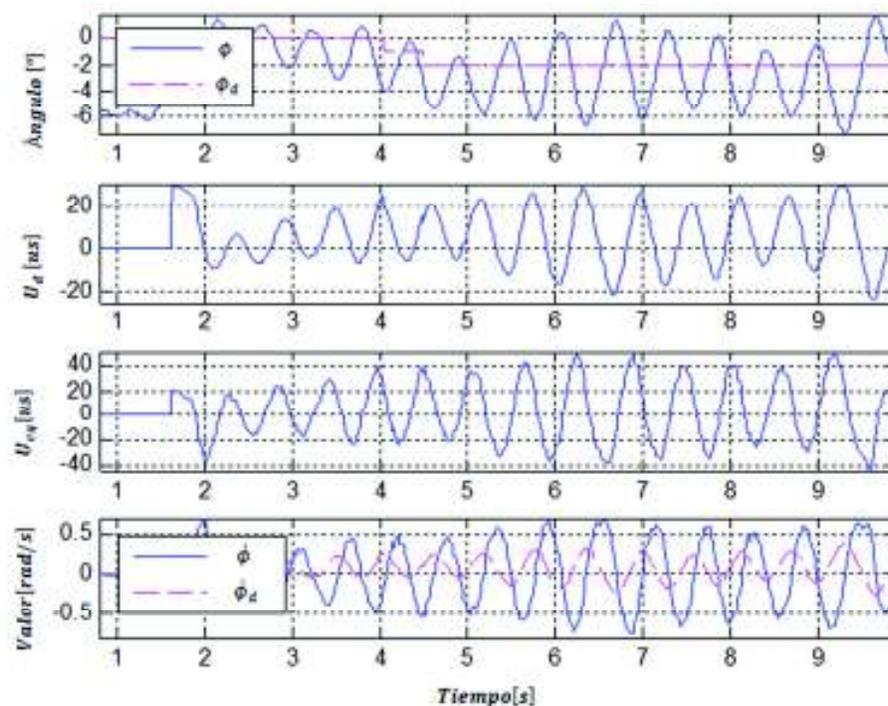


Figura 2.14. Resultados con $K_d = 30$ y $\delta = 3$ (valor bajo).

El parámetro δ se sintoniza realizando pruebas de vuelo, es decir que todos los controladores estarán funcionando y se dejará al dron libre para que se puede observar de mejor manera el comportamiento de los controladores, al principio no será posible dejarlo libre mucho tiempo pero es recomendable no sujetarlo para observar cómo actúan los controladores.

Si el valor de δ es muy bajo el hexarotor oscilará de forma visible alrededor del *setpoint* del ángulo por lo que la velocidad también será alta (ver Figura 2.14). Con un valor demasiado alto el control se demorará mucho en llegar al *setpoint* del ángulo y muy probablemente no pueda llegar y se quede oscilando cerca del mismo. Un valor adecuado de δ es aquel que permite llegar al *setpoint* angular y no permite que el ángulo se aleje mucho de su *setpoint*, además que no debe ser brusco por lo que no deben observarse oscilaciones y se debe mantener la velocidad lo más baja posible (ver Figura 2.15). Nótese que no es necesario que el controlador sea capaz de mantener el *setpoint*, esto se debe a que sin la parte integral esto no es posible [11], [39].

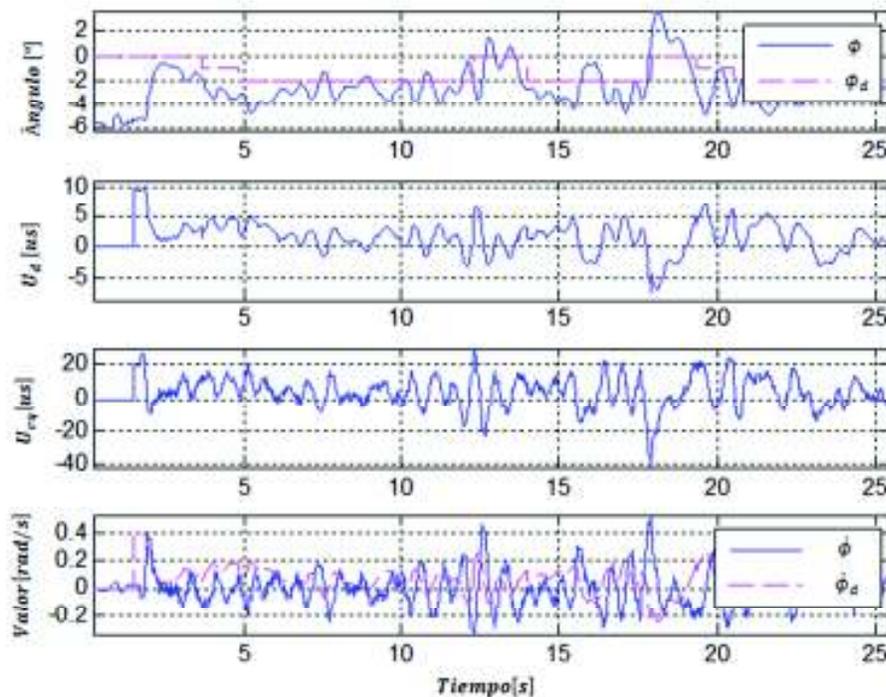


Figura 2.15. Resultados con $K_d = 30$ y $\delta = 10$ (valor adecuado).

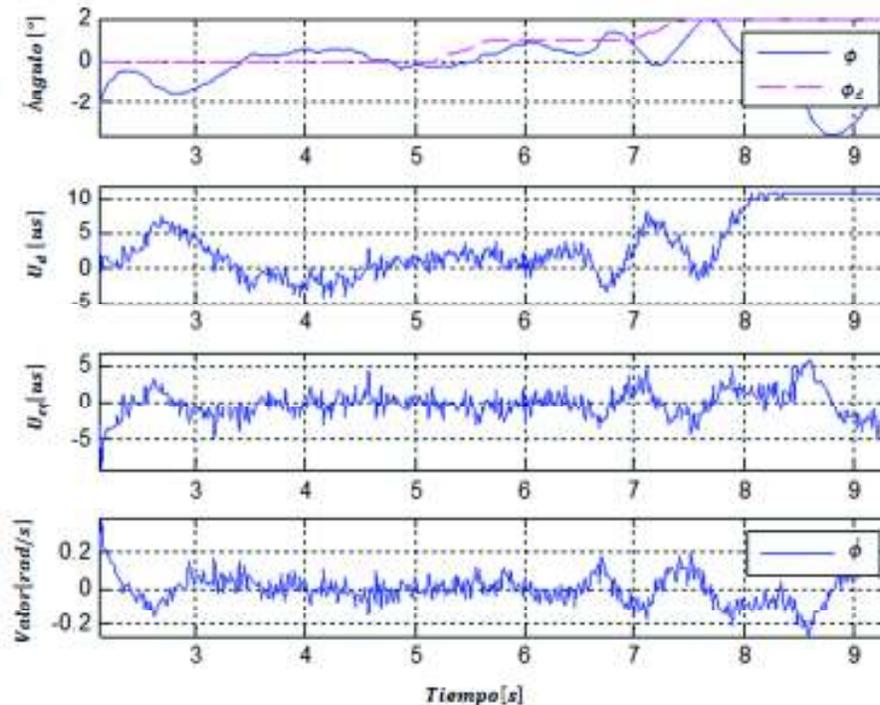


Figura 2.16. Sin el parámetro λ_2 el *setpoint* se alcanza pero no se mantiene.

A medida que el ángulo se acerca a su *setpoint* la acción de control discontinuo y la acción de control equivalente van disminuyendo su valor, y por tanto se pierde la fuerza necesaria para seguir llevando al ángulo hacia su *setpoint* y sobre todo mantenerlo (ver Figura 2.16), motivo por el cual es necesario una parte integral en el controlador. Por este motivo el objetivo durante esta etapa es alcanzar el *setpoint* del ángulo y oscilar cerca del mismo con la velocidad más baja posible.

2.3.3 SINTONIZACIÓN DE λ_2

Llegado a este paso el hexarotor podrá mantener un vuelo, sin embargo al no tener aún un valor en el parámetro de la parte integral de las superficie deslizante λ_2 (ver Ecuación 2.3) realizar desplazamientos será muy difícil debido a que el controlador logrará alcanzar los *setpoints* angulares pero no los mantendrá de una forma adecuada. El término integral de la superficie deslizante se limita al valor de δ para que la parte integral pueda llevar la acción de control discontinua a su valor máximo K_d .

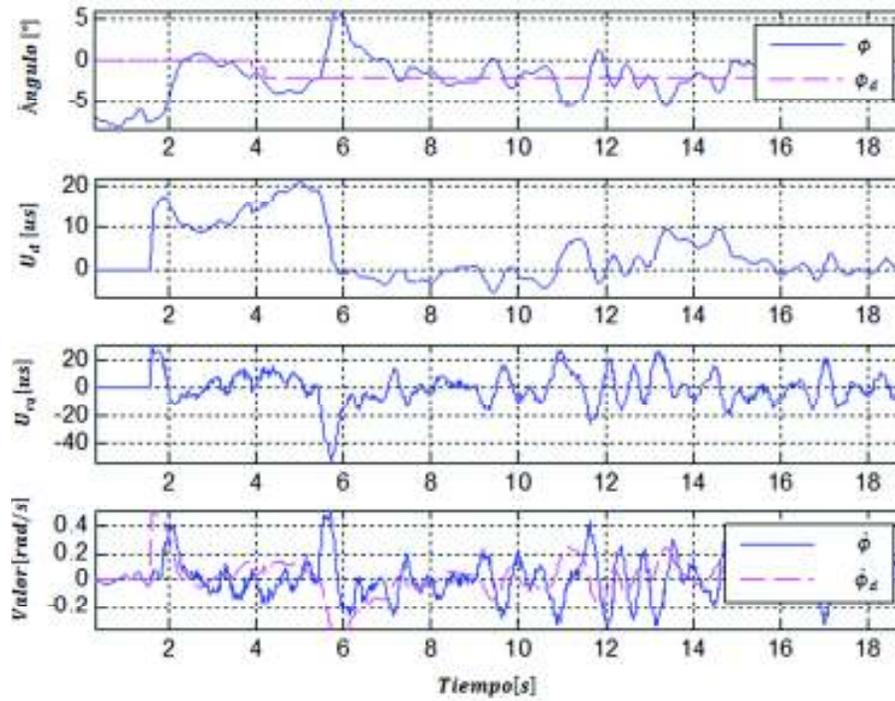


Figura 2.17. Resultados obtenidos con $\lambda_2 = 0.1$ (valor bajo).

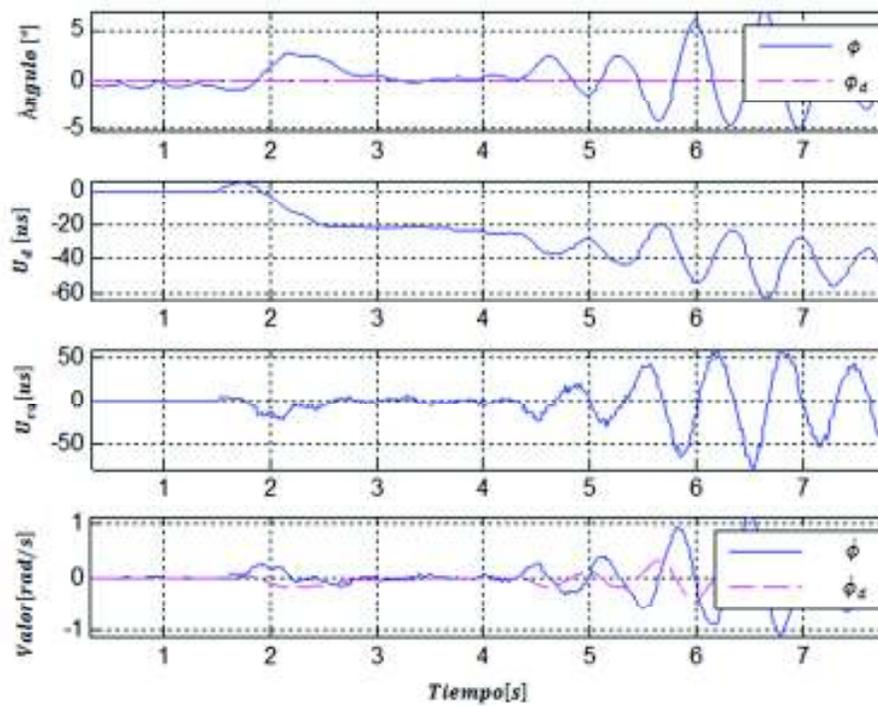


Figura 2.18. Resultados obtenidos con $\lambda_2 = 0.5$ (valor alto).

Para probar el desempeño de la parte integral es necesario que durante el vuelo del dron se realicen cambios del *setpoint* angular (usando el control de manejo) con el fin de verificar si el controlador es capaz de mantener el ángulo alrededor del *setpoint*.

Para sintonizar el valor de λ_2 se empieza aumentando su valor poco a poco. Con un valor bajo de λ_2 el control tardará mucho en llegar al *setpoint* (ver Figura 2.17). Si el valor de λ_2 es muy alto se tendrá oscilaciones bruscas alrededor del *setpoint* que serán visibles durante el vuelo (ver Figura 2.18). Un valor adecuado de λ_2 será aquel que permita: alcanzar los *setpoints* de forma rápida, no deben observarse oscilaciones en el dron y se debe mantener la velocidad lo más baja posible (ver Figura 2.19).

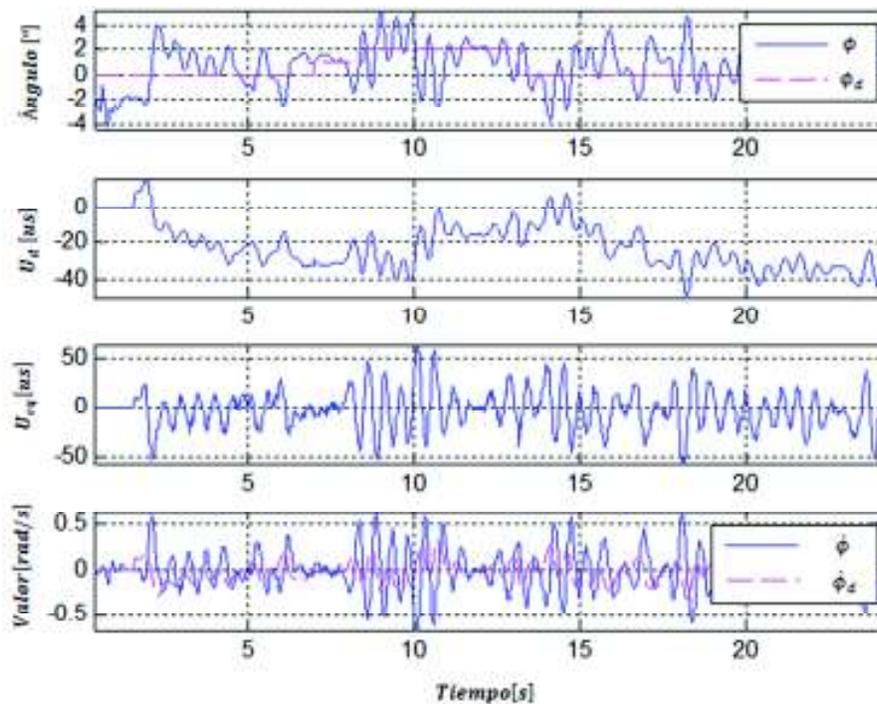


Figura 2.19. Resultados obtenidos con $\lambda_2 = 0.3$ (valor adecuado).

Es probable que se llegue a dar el caso en que el valor de la integral se satura y el valor absoluto de la acción de control discontinua llegue a su máximo (K_d). Este es un síntoma de que el valor escogido para K_d es demasiado bajo por lo que se deberá aumentarlo y por ende es probable que también se deba variar δ .

2.3.4 SINTONIZACIÓN DE $P_{\dot{\phi}}$

Como se mencionó anteriormente el parámetro $P_{\dot{\phi}}$ se usa para determinar el valor del *setpoint* de aceleración, el cual es un término que tiene un comportamiento similar al del término derivativo en un control PID. Por este motivo su principal efecto es el de lograr un mejor seguimiento del *setpoint* de velocidad, mientras mayor sea su valor se obtendrá una menor variación de velocidad con respecto de su *setpoint* a cambio de reducir la velocidad con la que se alcanzan los *setpoints* de ángulo y de velocidad.

Para este parámetro no se propone un procedimiento para sintonizar porque éste afecta la velocidad de reacción del dron ante los cambios de *setpoint* de los ángulos, su valor queda a preferencia del usuario y de cómo éste prefiera el comportamiento del hexarotor durante el vuelo.

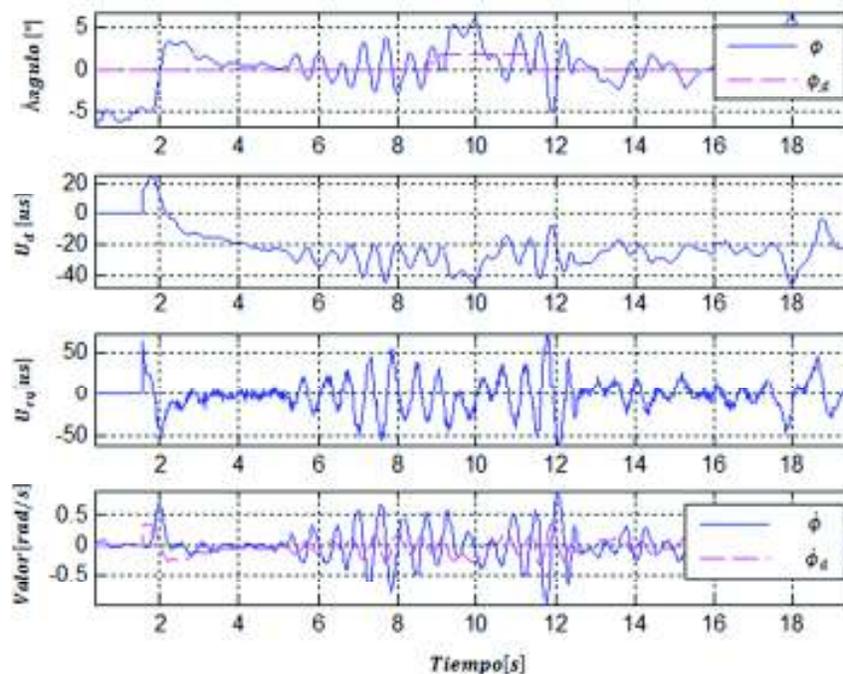


Figura 2.20. Resultados obtenidos con $P_{\dot{\phi}} = 0.3$ (valor bajo).

Si el valor de $P_{\dot{\phi}}$ es muy bajo casi no se podrá notar una diferencia en el vuelo del dron (ver Figura 2.20). Cuando $P_{\dot{\phi}}$ toma valores más altos se nota la reducción en la velocidad de los ángulos y sobre todo un retardo en el seguimiento de los *setpoints* de los ángulos (ver Figura 2.21). Para el caso del ángulo roll que ha sido mostrado como ejemplo en

esta sección se seleccionó un valor de $P_{\dot{\phi}} = 0.7$ ya que con este valor se tuvo una mejora en la velocidad de los ángulos y durante el vuelo las maniobras tenían un tiempo de respuesta que se sentía inmediato y sin retardos (ver Figura 2.22).

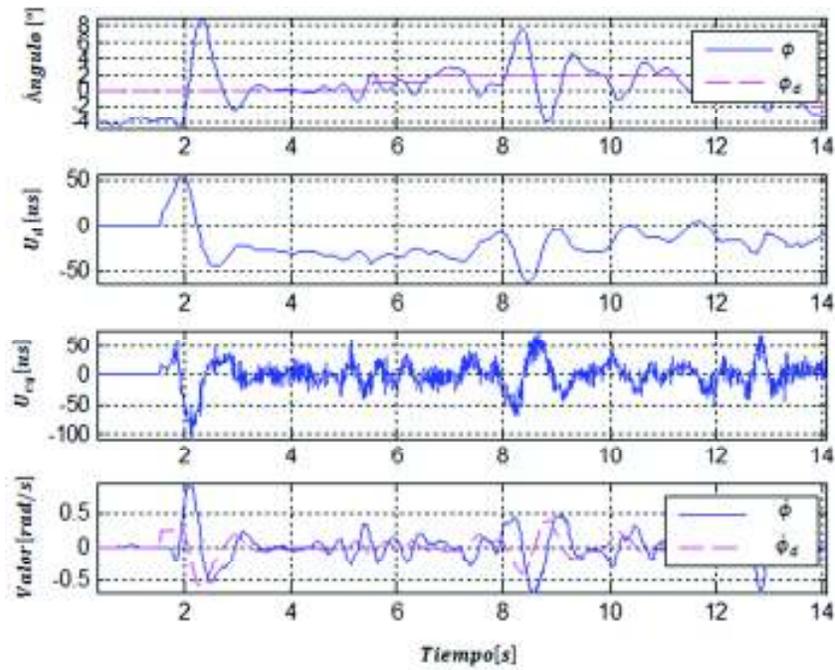


Figura 2.21. Resultados obtenidos con $P_{\dot{\phi}} = 1$ (valor alto).

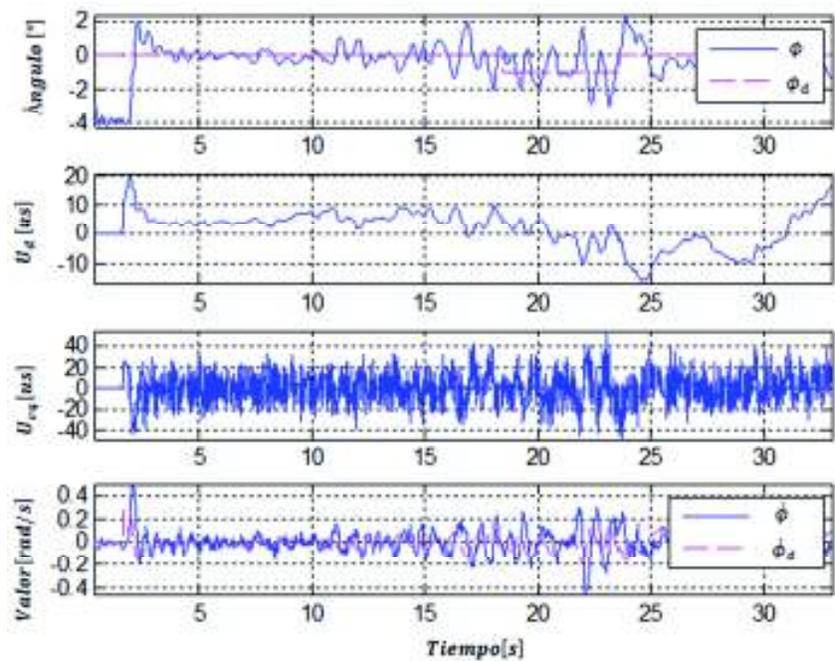


Figura 2.22. Resultados obtenidos con $P_{\dot{\phi}} = 0.7$ (valor adecuado).

Como se puede observar a lo largo del proceso de sintonización los resultados irán variando, esto se debe principalmente al acoplamiento entre variables que existe en el modelo del hexarotor y a que los parámetros λ_1 y λ_2 provocan cambios tanto en la acción de control discontinuo como en la acción de control equivalente. Por este motivo, al llegar a este punto de la sintonización es muy probable que se requiera reajustar los parámetros de los controladores nuevamente hasta lograr el comportamiento deseado.

Luego del proceso de sintonización y de reajuste, los parámetros de los controladores SMC para roll, pitch y yaw a utilizar se muestran en la Tabla 2.3.

Tabla 2.3. Parámetros de los controladores SMC.

Parámetro	P_ϕ	$P_{\dot{\phi}}$	K_d	λ_1	λ_2	δ
SMC de roll	4	0.7	30	40	0.6	10
SMC de pitch	4	0.5	30	30	0.5	20
SMC de yaw	4	0.1	60	1000	10	6

Para ello se puede tomar en consideración lo siguiente:

- Si los valores de velocidad son altos disminuir λ_1 , se debe buscar que sin la acción integral (λ_2) el parámetro λ_1 se acerque al *setpoint* de velocidad sin sobrepasarse mucho.
- Si la velocidad no alcanza el *setpoint* de velocidad puede probarse: aumentar λ_1 o disminuir $P_{\dot{\phi}}$.
- Si el hexarotor tarda en llegar a los *setpoints* de los ángulos o durante el vuelo se siente que el hexarotor tarda en responder a los cambios de dirección enviados, disminuir $P_{\dot{\phi}}$ o aumentar λ_2 .
- Si la acción de control discontinuo se satura por mucho tiempo, aumentar K_d y también aumentar δ .
- Si el *setpoint* del ángulo no es alcanzado rápidamente o no se alcanza, aumentar K_d y también aumentar δ o incrementar λ_2 .
- Si las oscilaciones alrededor del *setpoint* del ángulo son bruscas o con altas amplitudes, aumentar δ , disminuir λ_2 o disminuir λ_1 .

De manera general se pudo observar que:

- Si el problema radica en como la velocidad sigue su *setpoint*, esta dinámica se puede mejorar con λ_1 y $P_{\dot{\phi}}$.

- Si el problema radica en como el ángulo sigue su *setpoint*, esta dinámica se puede mejorar con K_d , δ y λ_2 .
- El parámetro P_ϕ se debe incrementar si se desea que el hexarotor alcance velocidades más altas y decrementar en el caso contrario.
- El parámetro P_ϕ permite reducir las velocidades de los ángulos, pero también hará que el *setpoint* de velocidad y por tanto el *setpoint* de ángulo se alcancen más lento. Por esto al variar P_ϕ es muy probable que se requiera volver a sintonizar los otros parámetros.

2.4 PROCEDIMIENTO DE SINTONIZACIÓN DE LOS CONTROLADORES PID

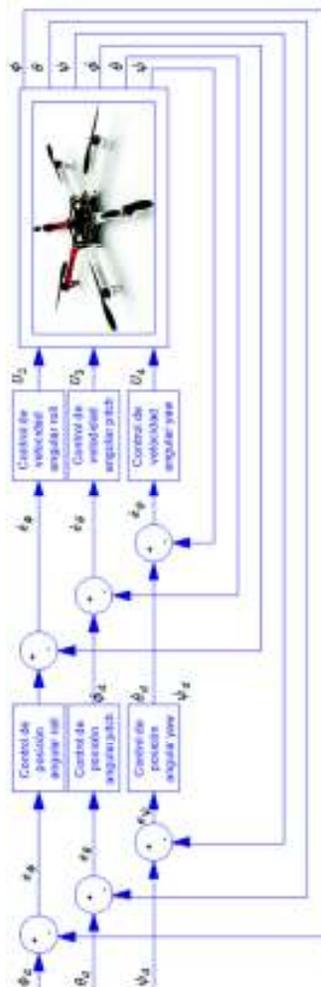


Figura 2.23. Esquema del control PID implementado.

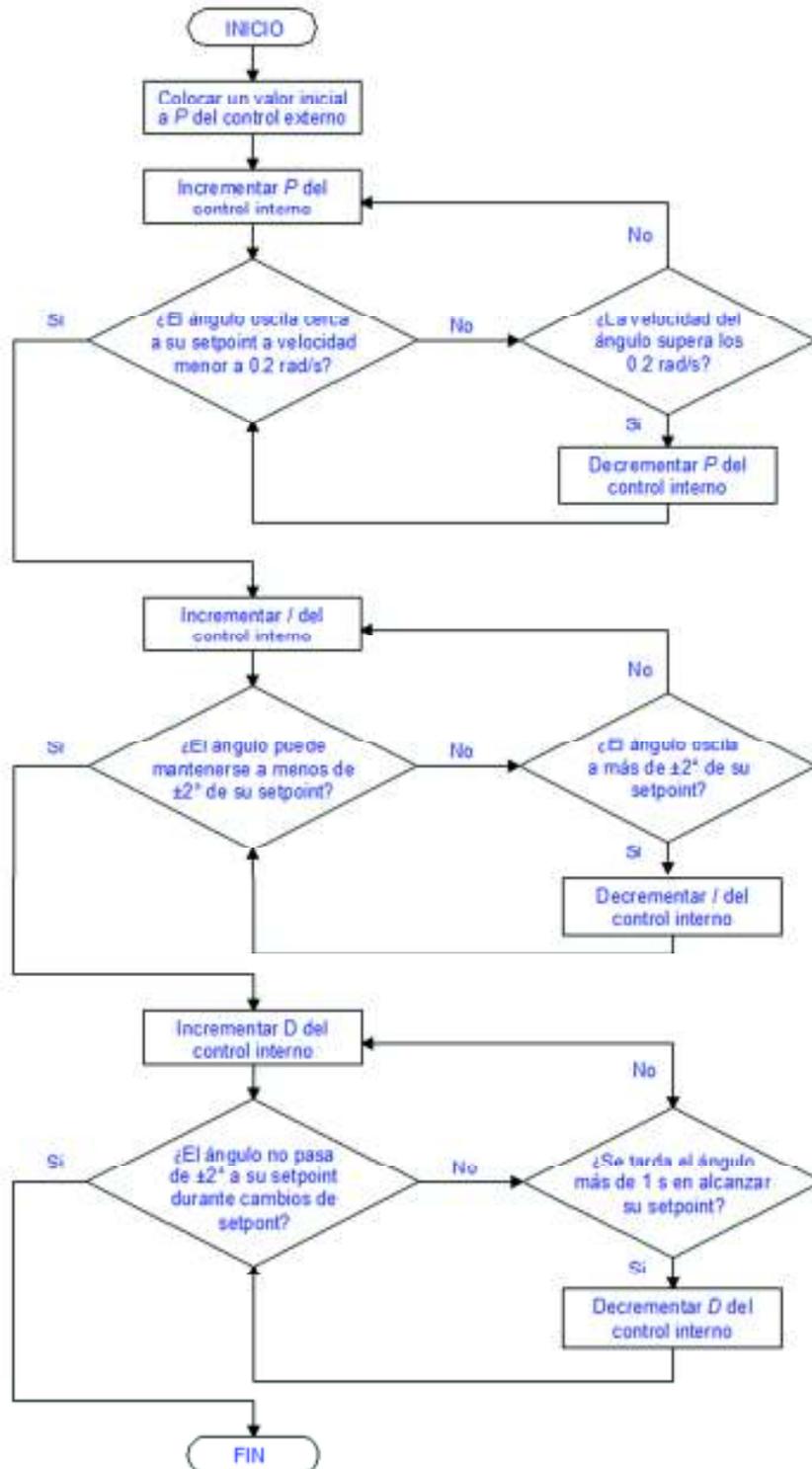


Figura 2.24. Flujograma del procedimiento de sintonización para el control PID.

Los parámetros del control PID se sintonizaron usando el procedimiento propuesto en [41], el cual es un procedimiento heurístico que requiere de pruebas de vuelo. En la Figura 2.23 se muestra el esquema de control PID implementado.

En la Figura 2.24 se muestra un flujograma más detallado del procedimiento propuesto para la sintonización de los controladores PID de orientación.

De forma general el procedimiento de sintonización para los controladores PID de orientación es el mostrado en el flujograma de la Figura 2.25, cuyos pasos son:

- Se coloca un valor en el parámetro P del control externo teniendo en cuenta que en [41] se recomienda que como máximo el *setpoint* de velocidad sea de $4.5^\circ/s$ por cada 1° de error.
- Encontrar el valor de P del control interno que permita al ángulo oscilar alrededor de su *setpoint* con velocidades menores a 0.2 rad/s .
- Se encuentra el valor de I del control interno que permita al ángulo mantenerse a menos de $\pm 2^\circ$ de su *setpoint*.
- Encontrar el valor de D del control interno que permite que durante los cambios de *setpoint* angular el ángulo no sobrepase en $\pm 2^\circ$ a su *setpoint*.



Figura 2.25. Procedimiento de sintonización del control PID.

2.4.1 SINTONIZACIÓN DEL CONTROL EXTERNO

Como se mencionó en la sección 2.1.2, el control externo es un control proporcional, al inicio de la sintonización se le colocó un valor fijo el cual fue de $P = 3$, debido a que en [41] se recomienda que como máximo el *setpoint* de velocidad sea de $4.5^\circ/s$ por cada 1° de error. Al tener el valor del parámetro del controlador externo con su valor fijo se procede a sintonizar el controlador interno.

A mayor valor de P en el control externo el drone intentará corregir los errores angulares más rápidamente, alcanzando así velocidades de giro más altas.

2.4.2 SINTONIZACIÓN DEL PARÁMETRO P DEL CONTROL INTERNO

El procedimiento para la sintonización de este parámetro es el siguiente:

- Se debe aumentar poco a poco el valor de P .
- Si el valor de P es muy bajo el hexarotor oscila lentamente alrededor del *setpoint*, incluso puede virarse al despegar debido a que le falta fuerza al controlador.

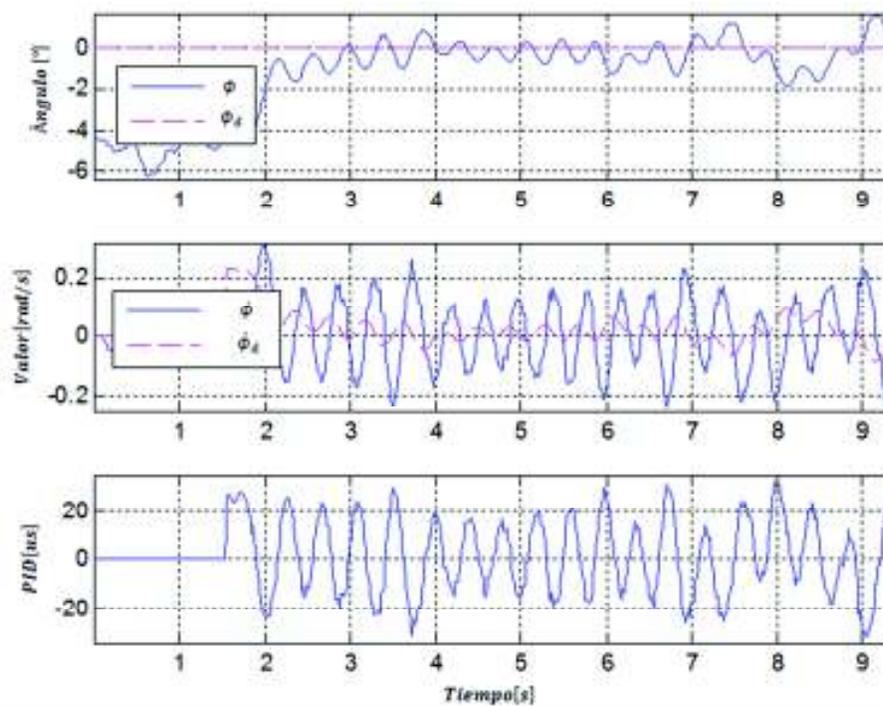


Figura 2.26. Resultados obtenidos con $P = 120$ (valor alto).

- Con un valor muy alto en P se obtendrán oscilaciones rápidas alrededor del *setpoint*, ver Figura 2.26.
- Un valor adecuado de P es el que provoca oscilaciones sutiles alrededor del *setpoint*, ver Figura 2.27.

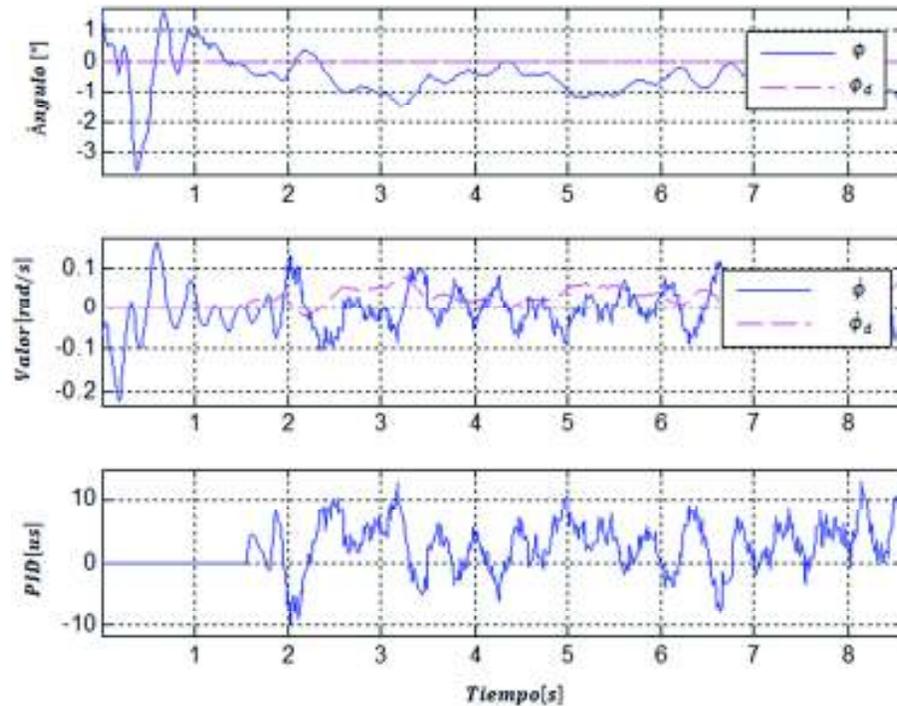


Figura 2.27. Resultados obtenidos con $P = 80$ (valor adecuado).

2.4.3 SINTONIZACIÓN DEL PARÁMETRO I DEL CONTROL INTERNO

El procedimiento para la sintonización de este parámetro es el siguiente:

- Se debe aumentar poco a poco el valor de I .
- Si el valor es muy bajo el hexarotor sigue los *setpoints* con lentitud y corrige de forma lenta las desviaciones del *setpoint*.
- Con un valor muy alto se producen vibraciones al girar y al compensar las desviaciones del *setpoint*, ver Figura 2.28.
- Un valor adecuado de I es aquel que permite giros suaves y mantener la inclinación durante los cambios de *setpoint* angular. Al iniciar los giros puede que oscile un poco, esto se debe a que falta el término derivativo, ver Figura 2.29.

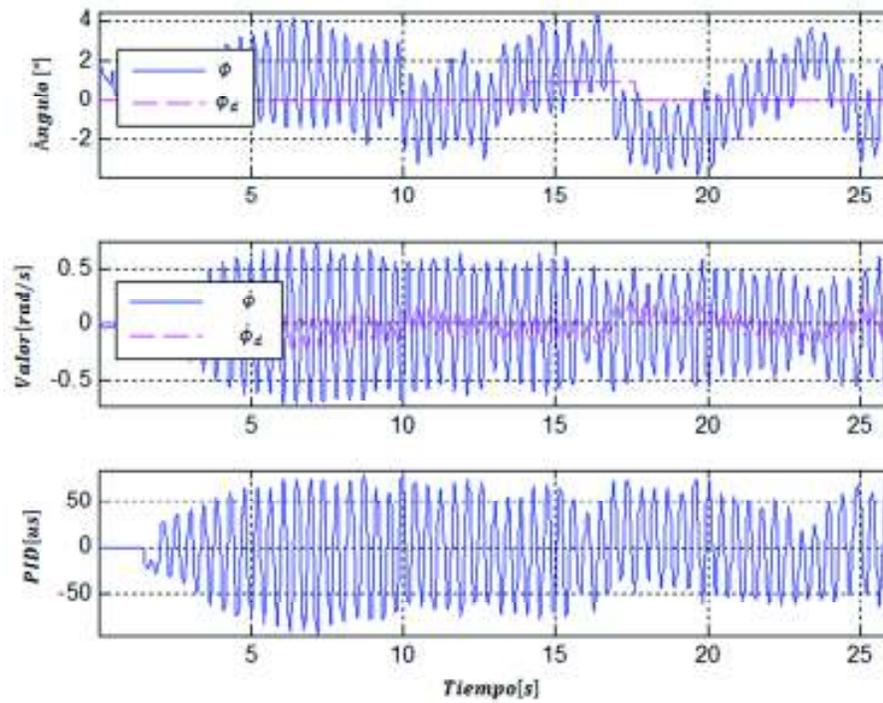


Figura 2.28. Resultados obtenidos con $I = 1$ (valor alto).

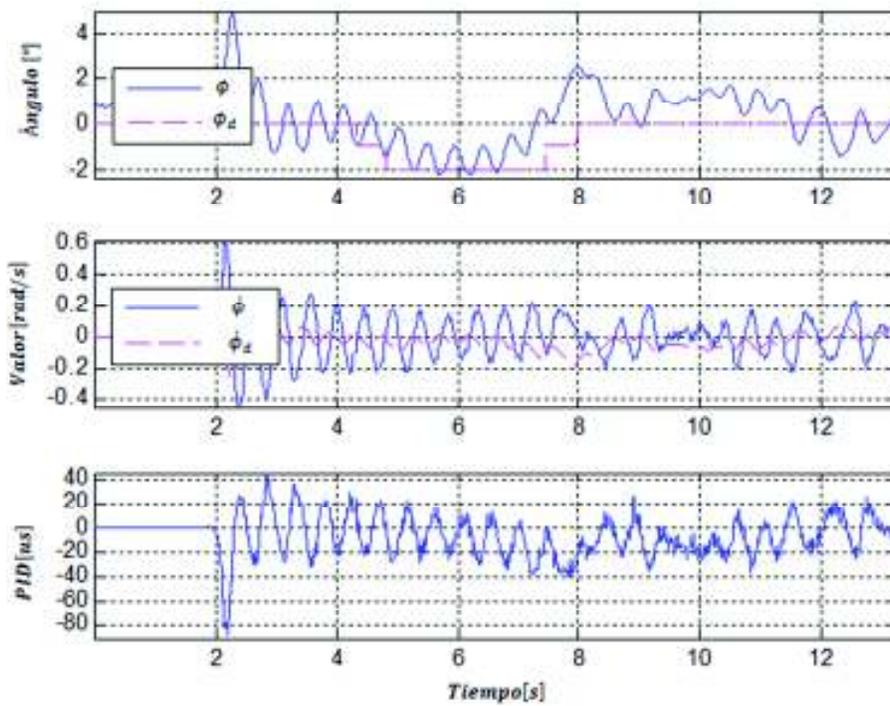


Figura 2.29. Resultados obtenidos con $I = 0.6$ (valor adecuado).

2.4.4 SINTONIZACIÓN DEL PARÁMETRO D DEL CONTROL INTERNO

El procedimiento para la sintonización de este parámetro es el siguiente:

- Se debe aumentar poco a poco el valor de D .
- Si el valor es muy bajo, al girar el hexarotor éste oscila al inicio del giro.
- Con un valor muy alto, al girar el hexarotor el giro se realiza muy lento. Es decir que tarda en alcanzar los *setpoints*.
- Un valor adecuado de D es el que permite realizar giros con suavidad y rapidez, ver Figura 2.30.

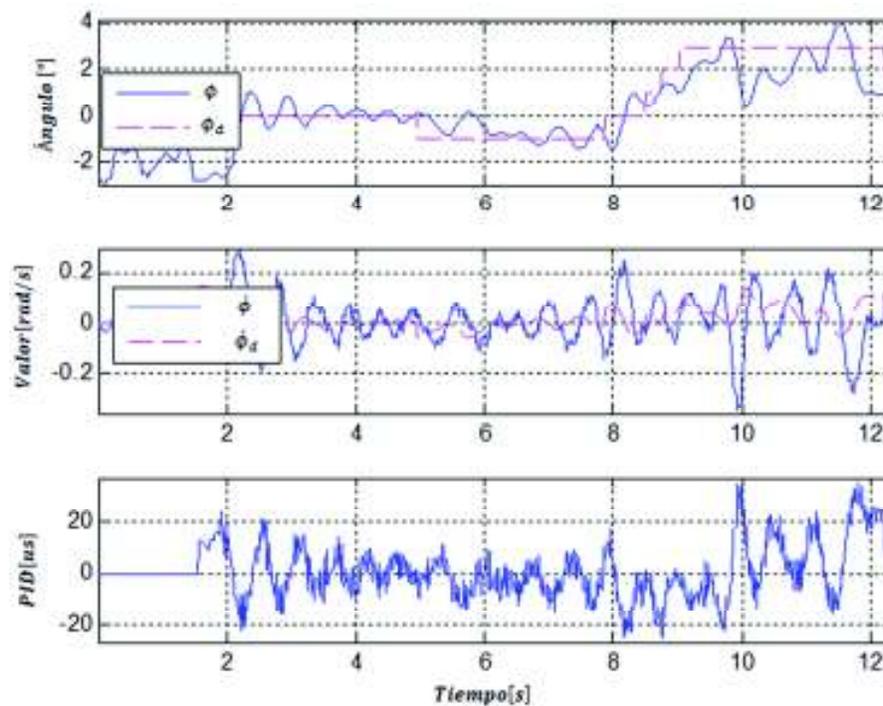


Figura 2.30. Resultados obtenidos con $D = 0.5$ (valor adecuado).

Tabla 2.4. Parámetros del control PID de roll.

Control	Posición	Velocidad			Límite integral
		P	I	D	
Roll	3	80	0.6	0.5	30
Pitch	3	40	0.5	0.4	30
Yaw	3	700	10	0.1	100

Los parámetros del control PID de orientación obtenidos con la sintonización se muestran en la Tabla 2.4.

2.5 CALIBRACIÓN DEL CONTROL DE MANEJO

La configuración de la magnitud en la que se variarán los parámetros de: pulso base de los ESC (control de altura) y los *setpoints* de los ángulos roll, pitch y yaw queda a preferencia del usuario. Las configuraciones presentes en el programa del Arduino DUE para el manejo del hexarotor se resumen en la Tabla 2.5. Se limitó el valor al que pueden llegar los *setpoints* de roll y pitch debido a que mientras más se incline el hexarotor más rápido avanzará y se deseaba tener movimientos lentos para facilitar el manejo.

Tabla 2.5. Configuraciones para las acciones del control de manejo.

Parámetro	Incremento	Decremento	Limitante
Pulso base de los ESC	5 [<i>us</i>]	-3 [<i>us</i>]	-
Setpoint del ángulo roll	1 [°]	-1 [°]	±2 [°]
Setpoint del ángulo pitch	1 [°]	-1 [°]	±2 [°]
Setpoint del ángulo yaw	22.5 [°]	-22.5 [°]	±172 [°]

3 RESULTADOS Y DISCUSIÓN

3.1 RESULTADOS

3.1.1 PRUEBAS REALIZADAS

En la siguiente sección se muestra las pruebas realizadas, comenzando con el *offset* del giroscopio donde se pretende tener un valor promedio como referencia a las mediciones futuras, el filtro de alisado de Kalman en éste se menciona la implementación de este filtro en el giroscopio debido al ruido presente en las medición. Seguidamente se tratará de la compensación de giro en yaw para evitar errores en la estimación de la orientación. Posteriormente se continuará con la fusión de sensores para corregir la deriva del giroscopio y el ruido de la medida del acelerómetro.

3.1.1.1 *Offset* del giroscopio

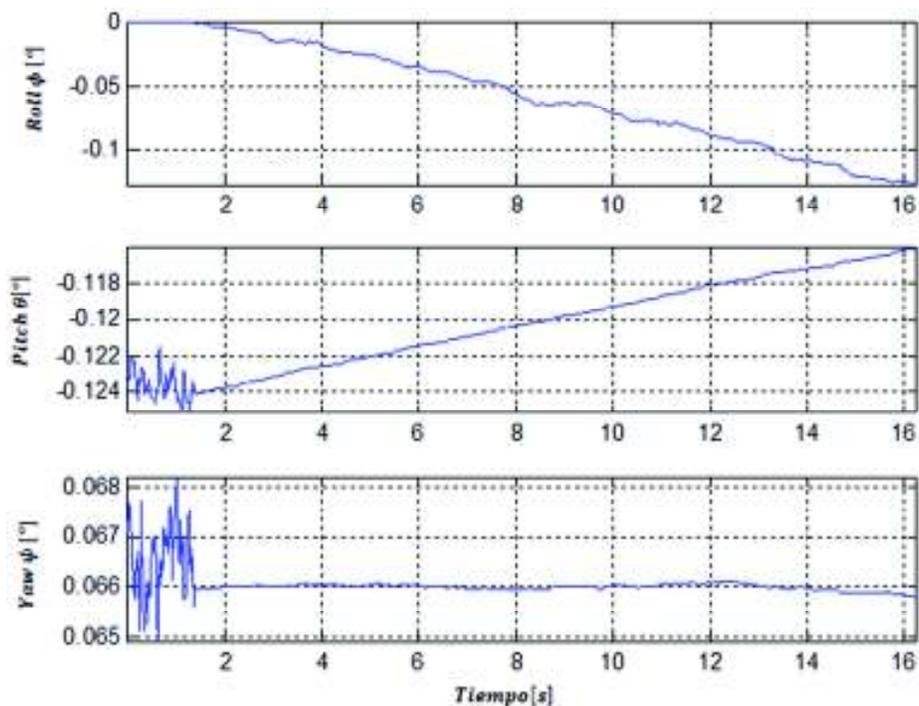


Figura 3.1. Deriva ocasionada por el *offset* presente en el giroscopio.

Si no se realiza el procedimiento del *offset* del giroscopio (ver sección 2.2.2.1) todas las mediciones del giroscopio estarán desviadas del cero lo que ocasionará que la estimación de los ángulos de Euler presente una deriva constante en el tiempo como se ve en la Figura 3.1. Al tomar el *offset* y restarlo de todas las medidas se corrige este problema como se puede observar en la Figura 3.2.

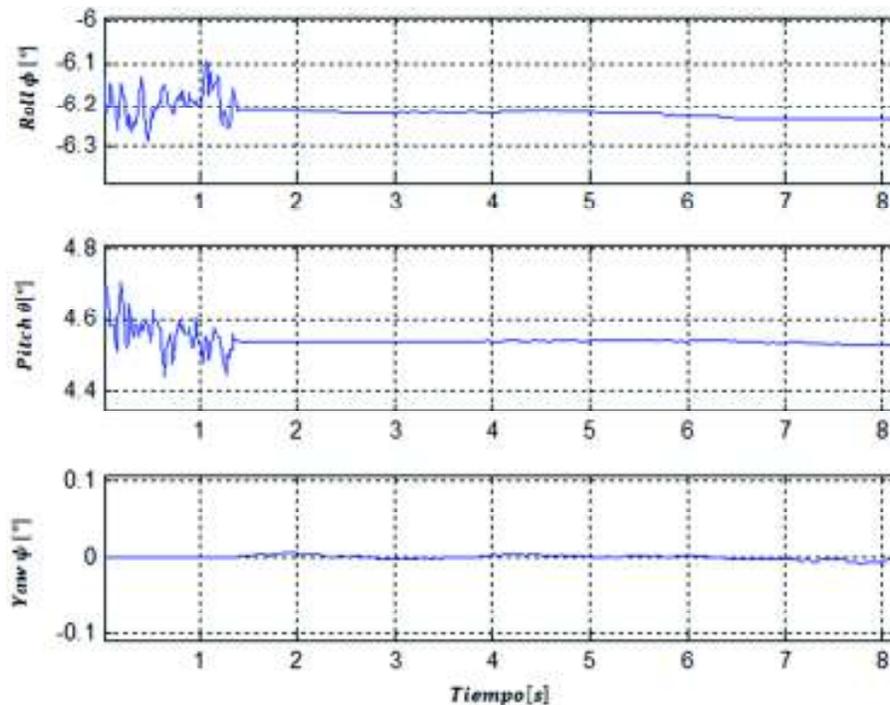


Figura 3.2. Resultado de restar el *offset* de las mediciones del giroscopio.

3.1.1.2 Filtros de Kalman del giroscopio

En la Figura 3.3 se puede observar la mejora obtenida al aplicar el filtro de alisado de Kalman a las mediciones del giroscopio (ver sección 2.2.3.1). Se observó que se debe encontrar un equilibrio entre el nivel de ruido retirado de la señal y su amplitud, ya que mientras más se busque reducir el ruido la amplitud de la señal también se verá reducida lo cual provoca errores considerables en las mediciones.

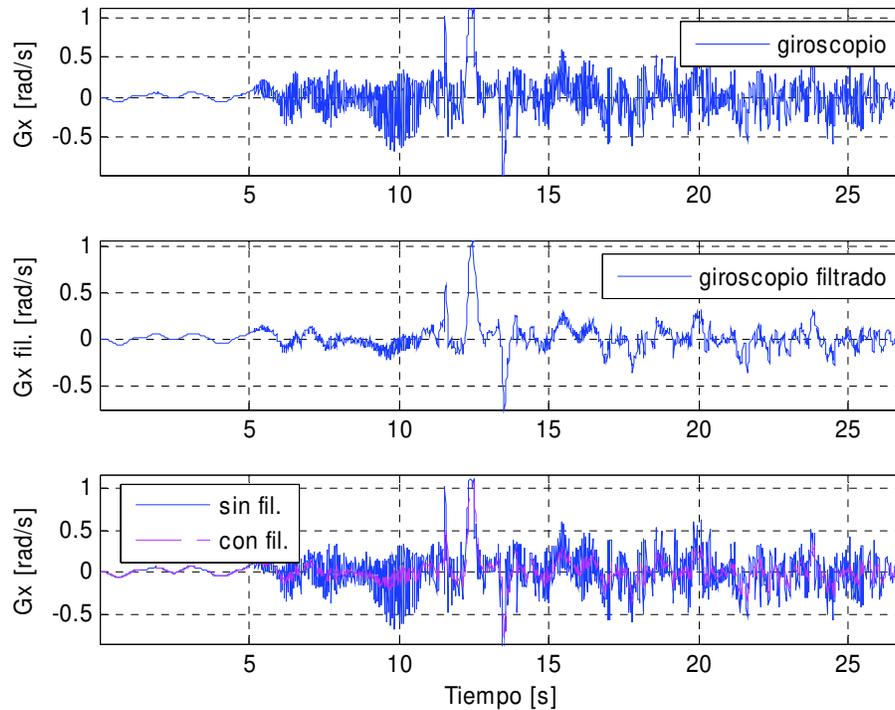


Figura 3.3. Filtro de Kalman aplicado a las mediciones del giroscopio.

3.1.1.3 Compensación de giro en yaw

La compensación de giro en yaw (ver sección 2.2.3.4) es necesaria debido a que si el ángulo roll o pitch o ambos se encuentran en un valor diferente de cero cuando ocurre un giro en el yaw los giroscopios de roll y pitch al no haber movimiento sobre sus ejes no permiten obtener la medida real de la inclinación del hexarotor [39].

Este fenómeno se explica brevemente con el siguiente ejemplo. Al iniciar el vuelo la IMU MPU6050 estará en posición horizontal por lo que la estimación de los ángulos roll y pitch estará cercana a cero. Posteriormente ocurre un giro en el ángulo pitch que lo llevó a una inclinación de 45° . Finalmente ocurre un giro en el ángulo yaw de 90° , la posición resultante es la equivalente a un giro en roll de 45° desde la posición inicial (antes de iniciar el vuelo) y no a un giro en pitch de 45° como mostraría la estimación de orientación si no se compensara el valor de roll y pitch cuando ocurre un giro en yaw [39].

Este problema se corrige trasladando la estimación del ángulo roll al ángulo pitch y viceversa sólo cuando ocurre un movimiento en el eje del yaw.

3.1.1.4 Fusión de sensores

El valor de confiabilidad debe ser alto para el giroscopio con el fin de que el ángulo estimado con el acelerómetro no influya en gran medida en la estimación, esto debido a la alta presencia de ruido en su medición.

En la Figura 3.4 se muestran los resultados con un factor de confiabilidad de 0.9996 que es el valor con el que se encuentran los filtros complementarios de roll y pitch en el programa del controlador. Se sostuvo al hexarotor y se realizaron giros de roll, pitch y yaw para verificar la precisión de la medida entregada por el giroscopio como se muestra en la Figura 3.4.

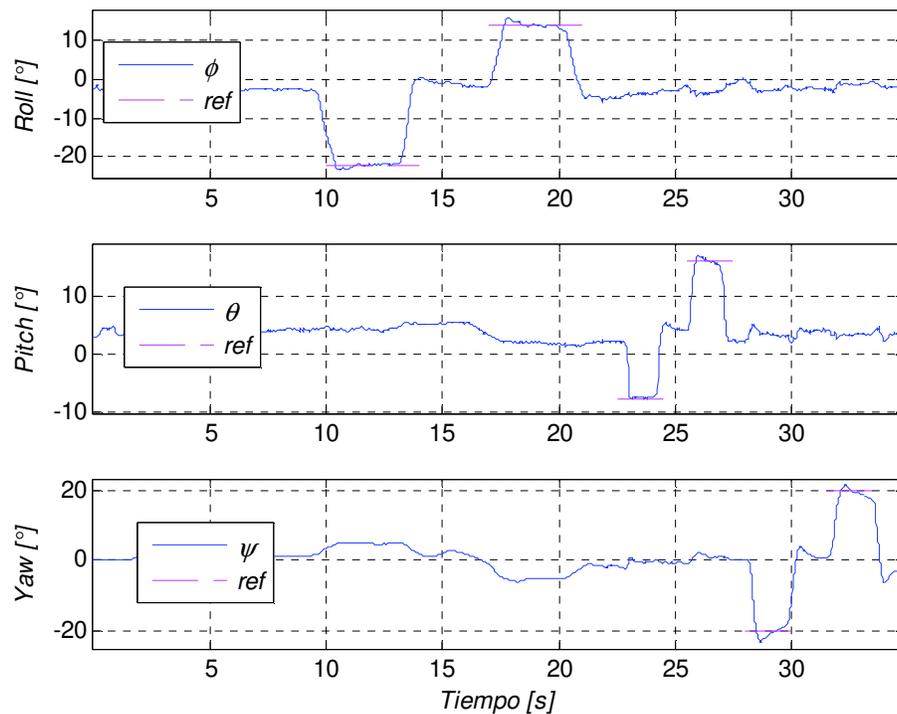


Figura 3.4. Ángulos estimados con factor de confiabilidad de 0.9996.

3.1.2 RESULTADOS DEL CONTROL SMC

La Figura 3.5 muestra al hexarotor durante el vuelo con los controladores SMC de orientación en funcionamiento y siendo maniobrado por el operador mediante el control de manejo.



Figura 3.5. Vuelo con el control SMC.

En la Figura 3.6 se puede observar los ángulos roll, pitch y yaw del drone mientras está volando, igualmente se puede apreciar los cambios de *setpoint* realizados y que su vez los controladores permiten que el hexarotor siga los *setpoints* que se le envían desde el control de manejo. Durante el vuelo se pudo observar como de forma esporádica aparecen perturbaciones que alejan a los ángulos de su *setpoint*, el control SMC es capaz de compensar estas perturbaciones rápidamente por lo que la gran mayoría de estas desviaciones pasan desapercibidas para el operador.

Con el método de suavizamiento mediante capa de frontera, se logró disminuir en gran medida el efecto del *chattering* el cual es causado por la parte discontinua del controlador SMC. En la Figura 3.7 se puede observar en mayor detalle el desempeño de los controladores durante el vuelo, se puede apreciar pequeñas perturbaciones que desvían al ángulo de su *setpoint* y como los controladores las compensan en tiempos cortos. También se puede ver el seguimiento de los *setpoints* de roll y pitch y como los controladores logran mantener a los ángulos de orientación alrededor de sus respectivos *setpoints*.

El desempeño de los controladores de roll, pitch y yaw se puede observar en la Figura 3.8, Figura 3.9 y Figura 3.10 respectivamente. De forma general en estas figuras se puede apreciar con mayor detalle las acciones que llevan a cabo los controladores SMC debido a que se muestran datos de la acción de control equivalente y la acción de control discontinua; además se puede ver como el control SMC mantiene la velocidad de los ángulos lo más baja posible con el fin de lograr un vuelo más estable.

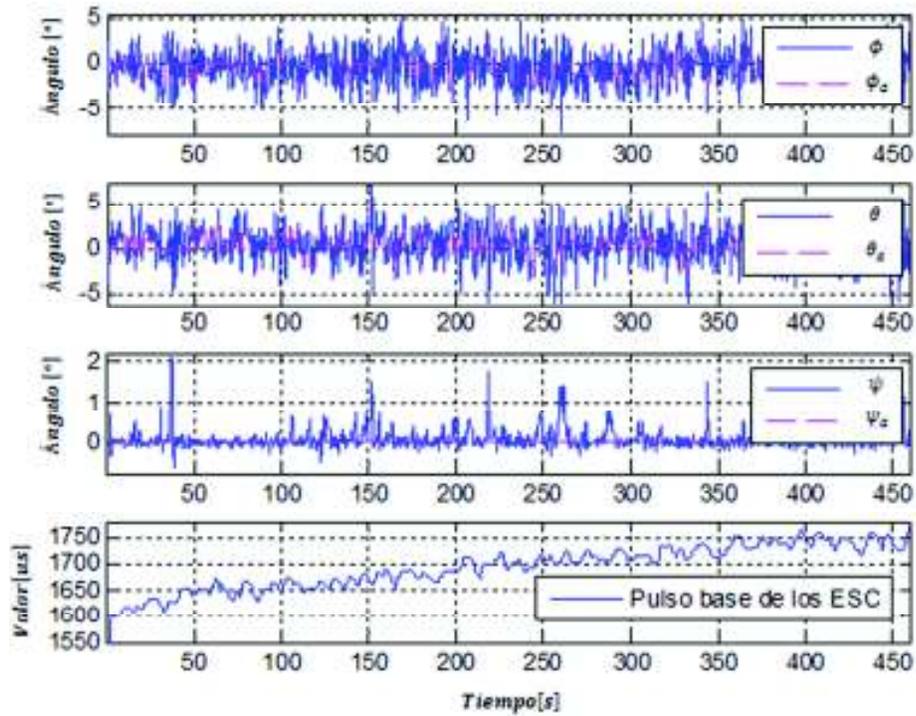


Figura 3.6. Resultados de vuelo de más de 7 minutos.

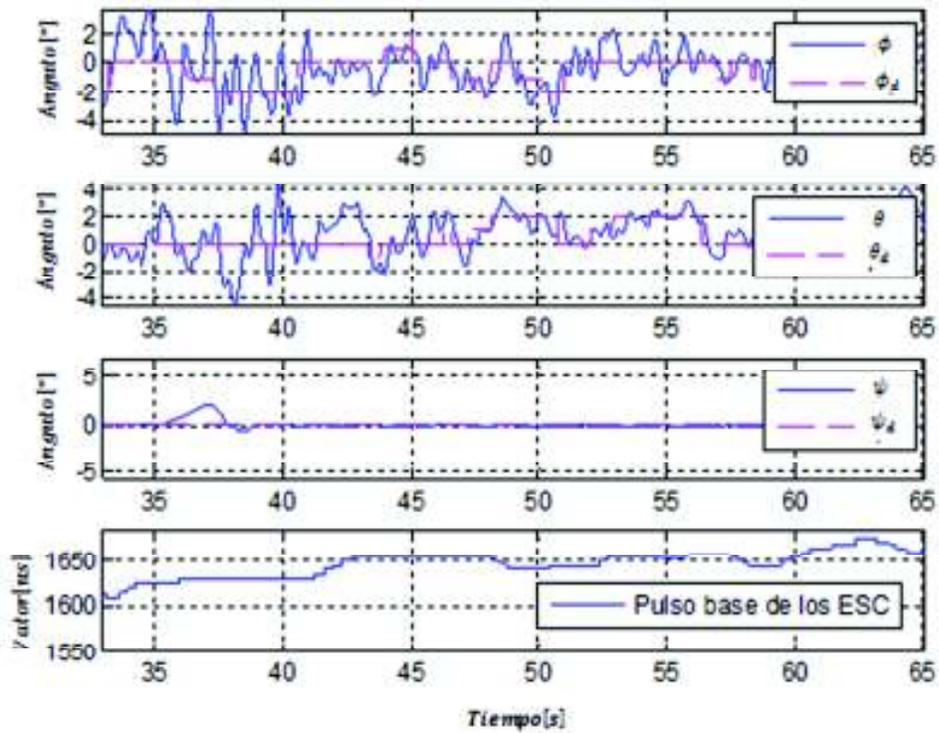


Figura 3.7. Resultados de los controladores SMC de roll, pitch y yaw.

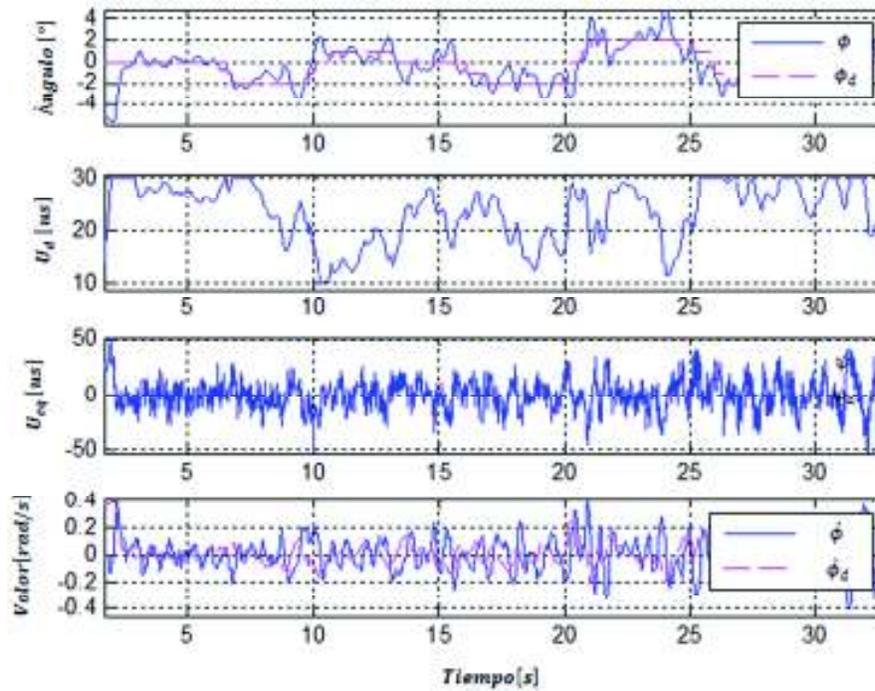


Figura 3.8. Resultados del control SMC de roll.

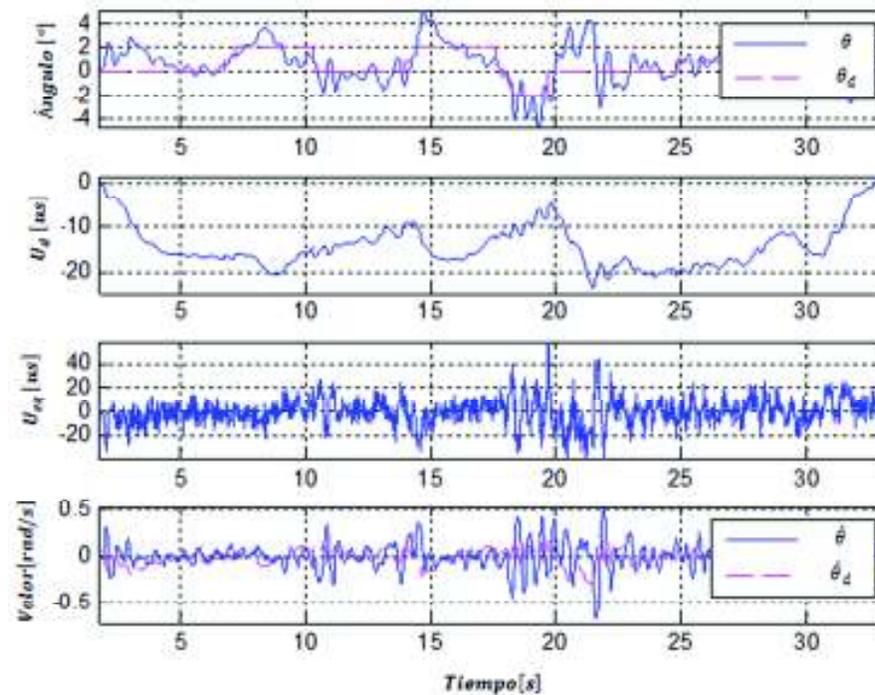


Figura 3.9. Resultados del control SMC de pitch.

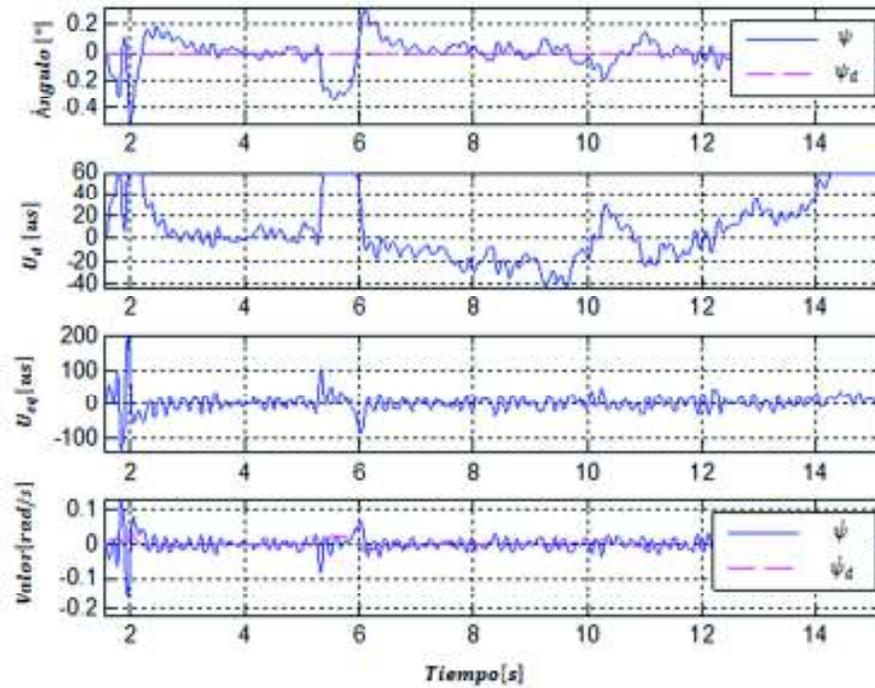


Figura 3.10. Resultados del control SMC de yaw.

En la Figura 3.11 se muestra el comportamiento del control del yaw cuando hay un cambio en su *setpoint*.

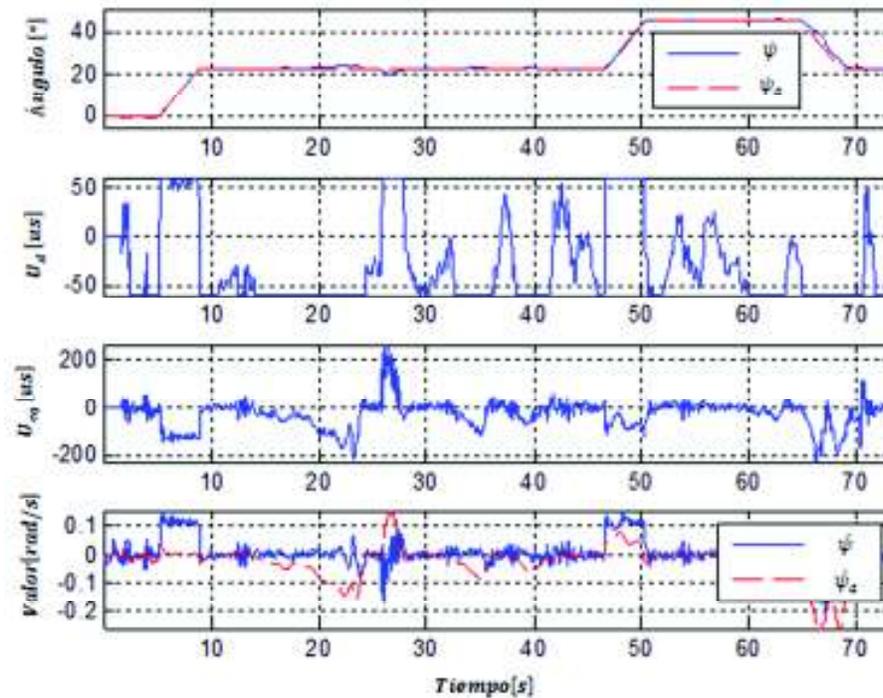


Figura 3.11. Giro en yaw con control SMC.

La Figura 3.12 muestra los resultados de un vuelo durante el cual se varía el *setpoint* de yaw. Los resultados obtenidos en cuanto al error de seguimiento del *setpoint* angular mostraron que:

- En estado estable el error no supera los $\pm 1^\circ$.
- Al cambiar de *setpoint* el error llega hasta los $\pm 2^\circ$ durante el estado transitorio.
- Las perturbaciones causan que el error llegue hasta los $\pm 4^\circ$ pero el control es capaz de superar estas perturbaciones de forma casi inmediata.

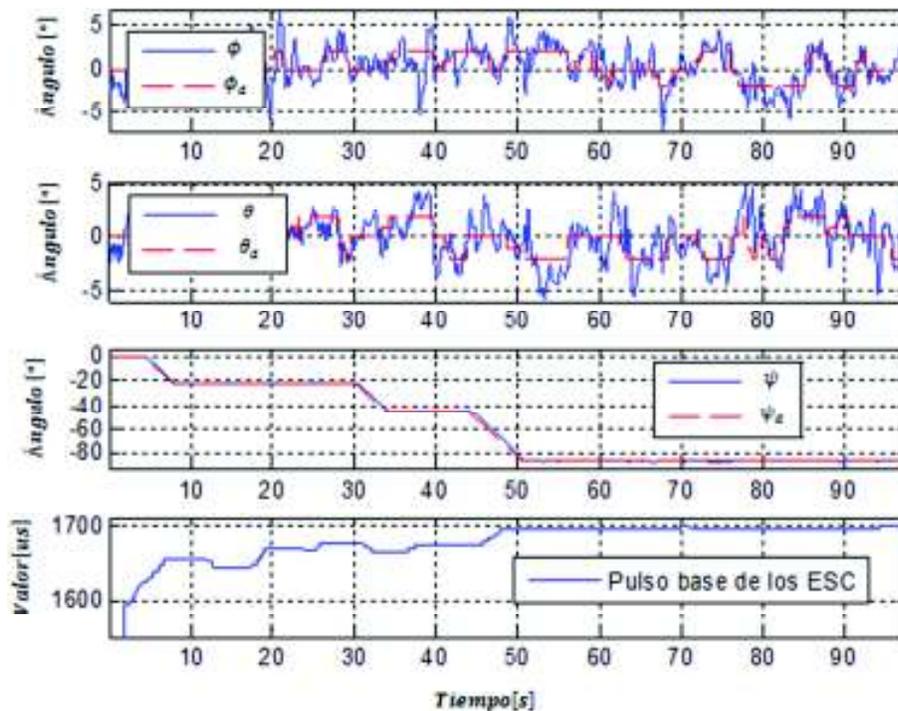


Figura 3.12. Vuelo con giro en yaw con control SMC.

3.1.3 RESULTADOS DEL CONTROL PID

En la Figura 3.13 se muestra el comportamiento de los controladores PID durante un vuelo de aproximadamente tres minutos, donde se pudo observar que el control PID se demora en compensar las perturbaciones en comparación con el control SMC; debido a esto se observan pequeñas oscilaciones del hexarotor cuando se encuentra en vuelo.

La Figura 3.14 es un zoom de la Figura 3.13, en ella se muestra con detalle el desempeño de los controladores PID durante el vuelo. En esta figura se puede ver que los ángulos tienden a desviarse más de sus *setpoints* en comparación con el control

SMC, además se observa como las perturbaciones que provocan oscilaciones alrededor del *setpoint* tardan más tiempo en ser compensadas.

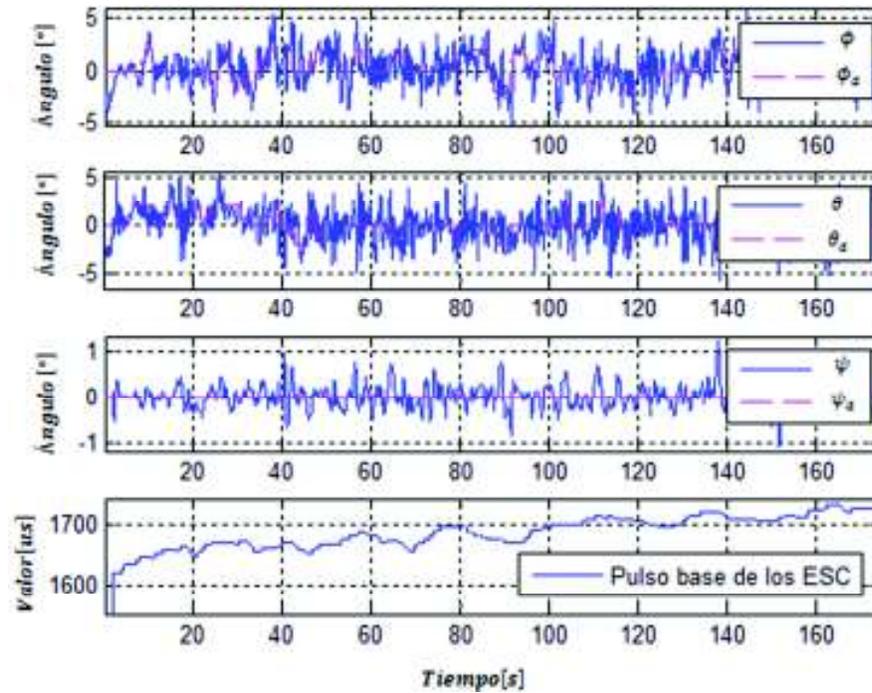


Figura 3.13. Resultados con controladores PID.

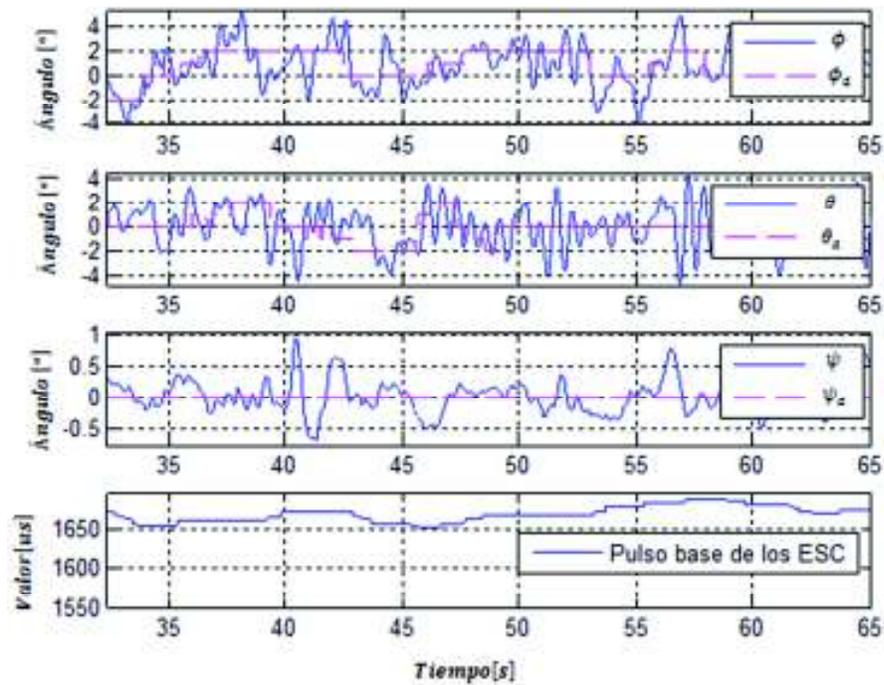


Figura 3.14. Desempeño de los controladores PID.

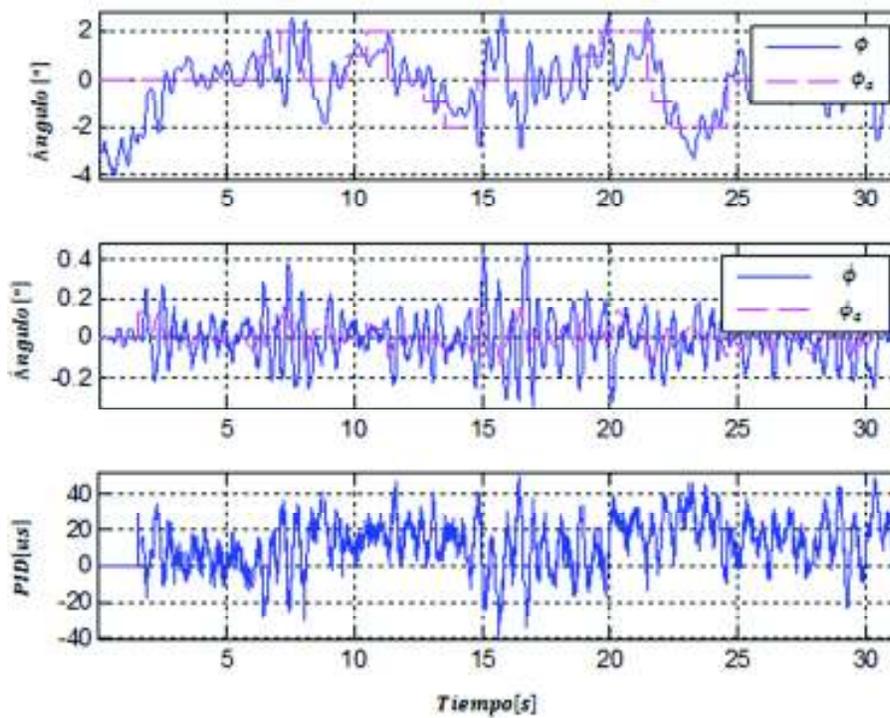


Figura 3.15. Resultados del control PID de roll.

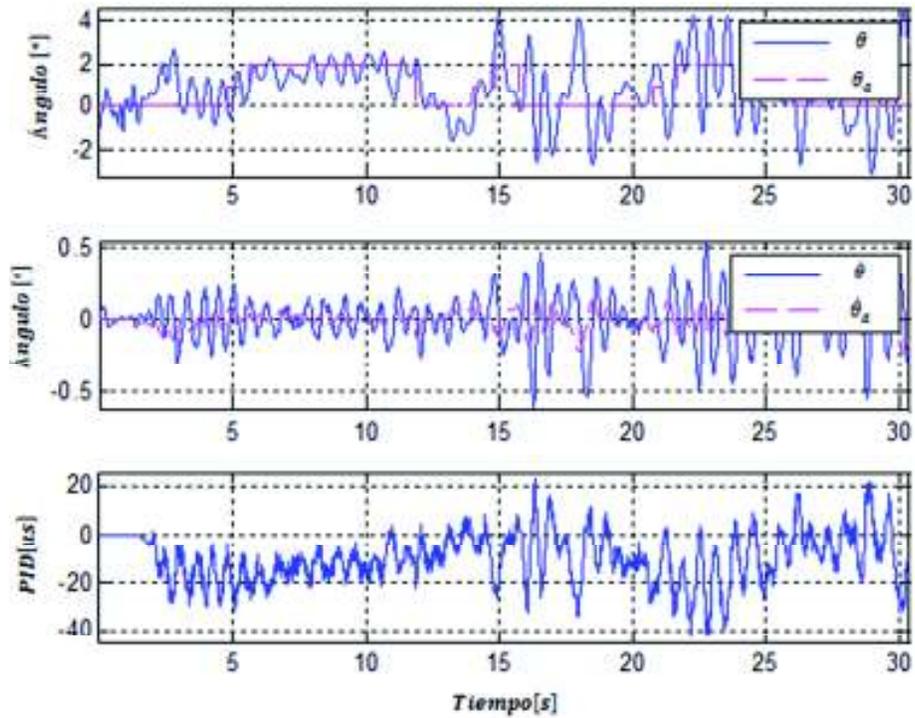


Figura 3.16. Resultados del control PID de pitch.

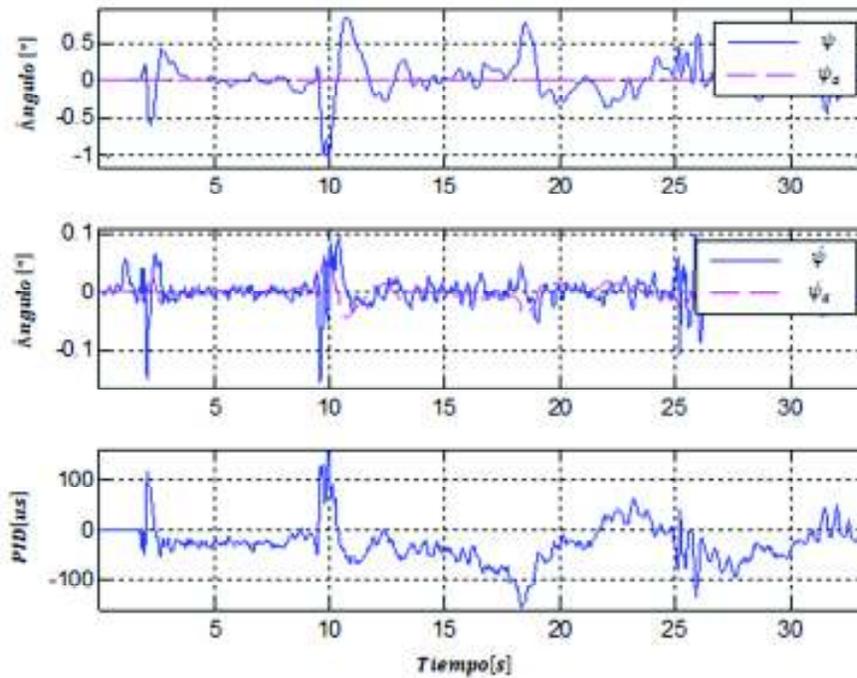


Figura 3.17. Resultados del control PID de yaw.

En la Figura 3.15, Figura 3.16 y Figura 3.17 se observa el desempeño de los controladores PID de roll, pitch y yaw respectivamente.

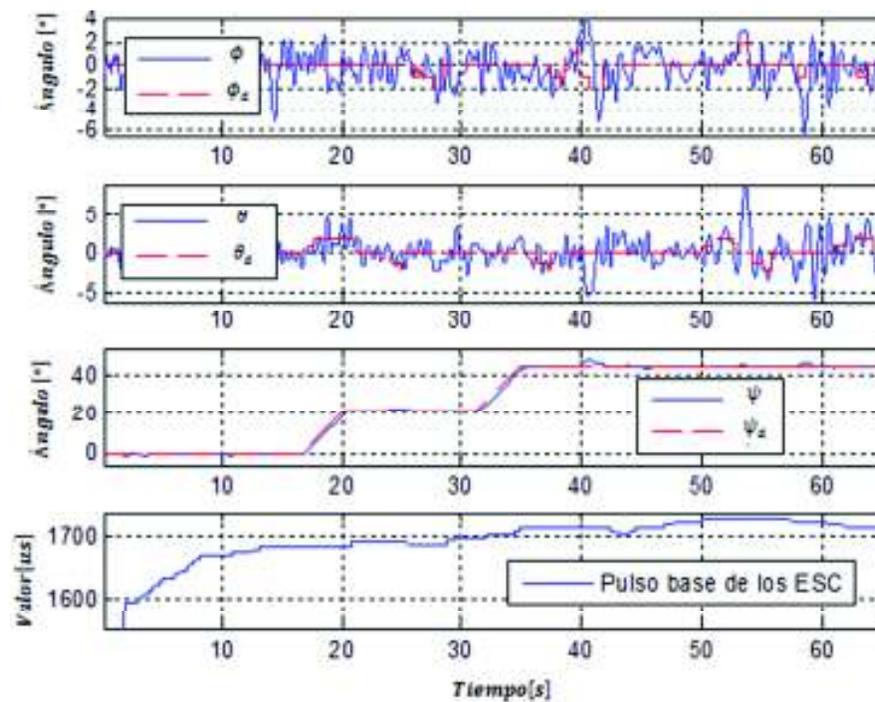


Figura 3.18. Giro en yaw con control PID.

En la Figura 3.18 se muestra el comportamiento del control PID de yaw cuando hay un cambio en su *setpoint* angular. En la Figura 3.19 se muestra un vuelo con control PID en el cual el *setpoint* del yaw se varía.

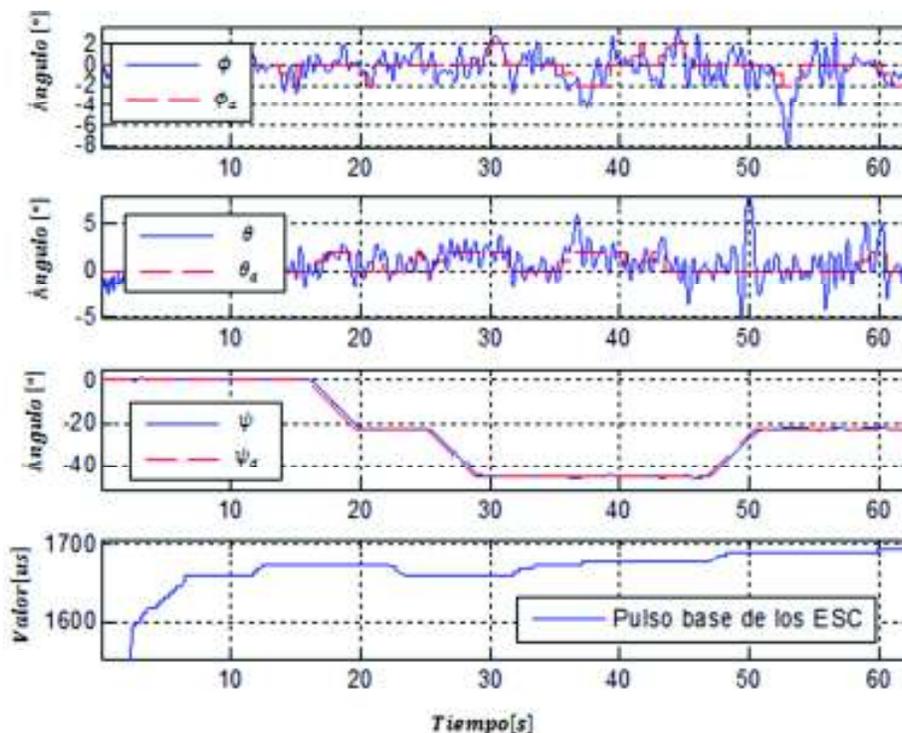


Figura 3.19. Vuelo con giro en yaw con control PID.

De forma general en estas figuras se puede ver como las señales de control PID responden ante las perturbaciones y los cambios de *setpoint*. Además se puede ver como las velocidades de los ángulos son ligeramente mayores a las obtenidas con el control SMC.

En cuanto al desempeño del control PID se pudo observar que:

- El desempeño al seguir el *setpoint* del ángulo es menor en comparación al control SMC, se observó que se llegaba al *setpoint* pero por lo general el control PID mantiene al ángulo cerca al *setpoint* mientras que con el control SMC el ángulo tiende a oscilar alrededor del *setpoint*.
- La velocidad angular obtenida con el control PID es un poco más elevada que la obtenida con el control SMC, el control PID mostró picos de velocidad lo que ocasionaba una oscilación visible del hexarotor.

- El control PID responde de forma menos eficiente ante las perturbaciones. Durante el vuelo se presentan perturbaciones que el control SMC corregía de manera inmediata, por lo que si se observaban oscilaciones éstas desaparecían rápidamente. Por el contrario, el control PID requiere de mucho más tiempo para lograr que las oscilaciones causadas por las perturbaciones desaparezcan.
- En todas las pruebas realizadas con el control PID las oscilaciones generadas por las perturbaciones provocaban que el hexarotor perdiera altura rápidamente y además dificultaban el manejo del dron ya que se sentía que el hexarotor tardaba en responder a los comandos enviados.

3.1.4 COMPARACIÓN ENTRE LOS CONTROLADORES PID Y SMC

Se usaron dos criterios de comparación: el primero consistió en el seguimiento de *setpoints* y el segundo criterio es la respuesta ante perturbaciones.

3.1.4.1 Seguimiento de *setpoints*

Para medir el desempeño de los controladores utilizados al seguir *setpoints* se usa el criterio de la integral del error cuadrático (ISE) mostrado en la Ecuación 3.1. Este criterio castiga los errores grandes de seguimiento tales como los sobre picos, los transitorios y la respuesta ante las perturbaciones [42].

Mientras menor sea el índice de desempeño esto indica que el sistema de control es capaz de eliminar los errores grandes en poco tiempo, que puede tolerar pequeños errores que persisten a lo largo del tiempo y que la respuesta del control es rápida, con poco sobre pico y con pocas oscilaciones [42].

$$ISE = \int e^2(t) dt$$

Ecuación 3.1. Criterio de desempeño de la integral del error cuadrático.

Para comparar ambos controladores se realizaron diferentes *setpoints* generados por software en un tiempo determinado, además en ciertos intervalos de tiempo se enviaron

setpoints a dos o más ángulos simultáneamente. Esta prueba se realizó con el hexarotor fijado de tal manera que se permitan giros de orientación pero no una variación de altura.

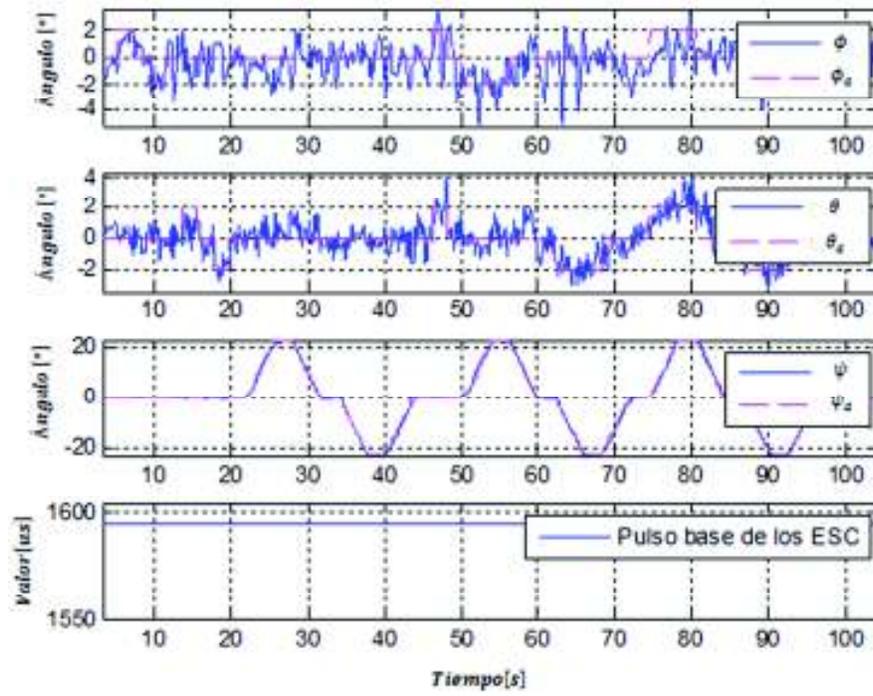


Figura 3.20. Pruebas de seguimiento de *setpoints* con control SMC.

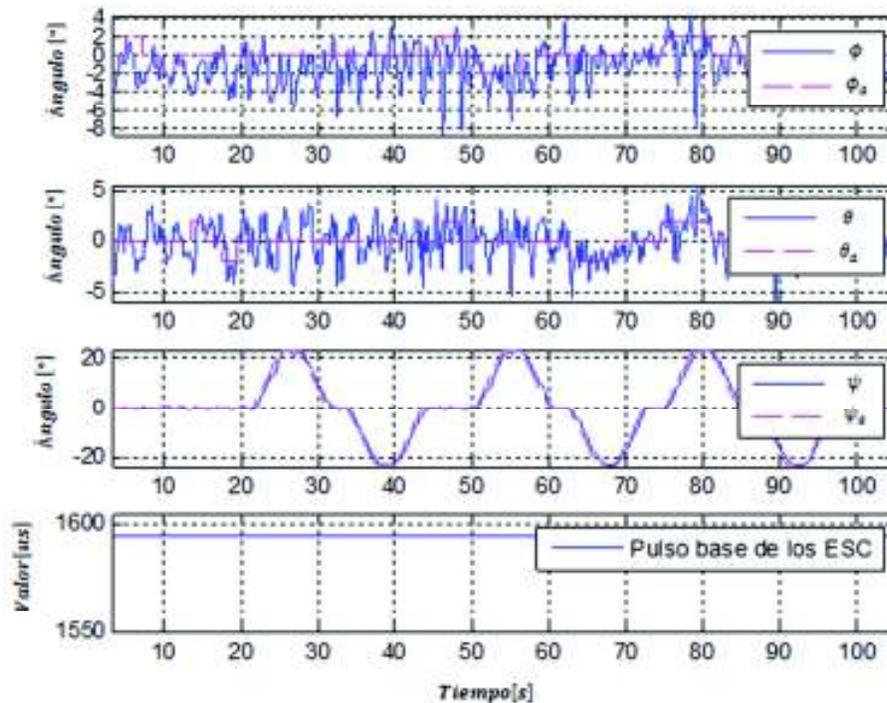


Figura 3.21. Pruebas de seguimiento de *setpoints* con control PID.

En la Figura 3.20 se muestran los controladores SMC de orientación con los diferentes *setpoints*, y en la Figura 3.21 se muestran los controladores PID de orientación con los diferentes *setpoints*.

La Figura 3.22 muestra una comparación de los errores de seguimiento (con los datos de la Figura 3.20 y Figura 3.21) de los controladores de orientación donde se puede visualizar que el control SMC tiene mejor seguimiento que el control PID.

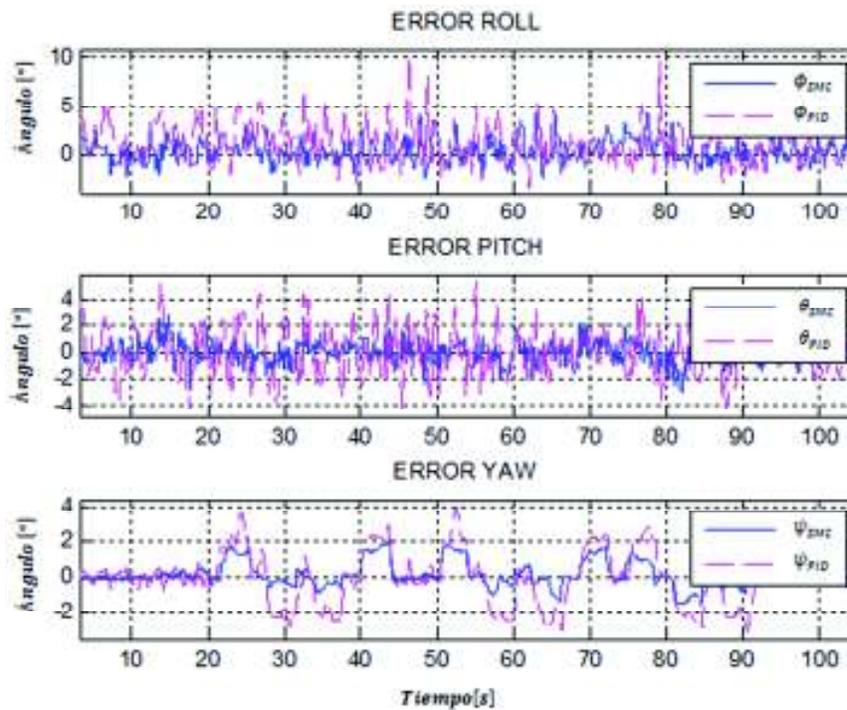


Figura 3.22. Comparación de los errores de seguimiento de los controladores de orientación.

Para facilitar la comparación del índice de desempeño se usa el porcentaje de diferencia el cual compara ambos criterios de desempeño con su valor promedio como se muestra en la Ecuación 3.2. La diferencia de los valores a comparar se divide para el promedio de los valores porque no se cuenta con un valor considerado como patrón [42].

$$\% \Delta = \frac{ISE_{SMC} - ISE_{PID}}{\frac{ISE_{SMC} + ISE_{PID}}{2}} \cdot 100\%$$

Ecuación 3.2. Porcentaje de diferencia entre dos cantidades.

Un porcentaje de diferencia positivo muestra un incremento con respecto al valor con el que se le compara; en este caso con el ISE_{PID} , en el caso de obtener un valor negativo esto indica una reducción [42].

En la Tabla 3.1 se muestra la diferencia porcentual entre los criterios de desempeño de los controladores SMC y PID, utilizando los datos de la Figura 3.20 y Figura 3.21.

Tabla 3.1. Índices de desempeño obtenidos.

Control	ISE_{SMC}	ISE_{PID}	$\% \Delta$
Roll	8709	27035	-102.5363
Pitch	3878	15540	-120.1001
Yaw	3165	14046	-126.4288

3.1.4.2 Perturbaciones

Para introducir perturbaciones se colocó al hexarotor fijado de tal manera que se permitan giros de orientación pero no una variación de altura.

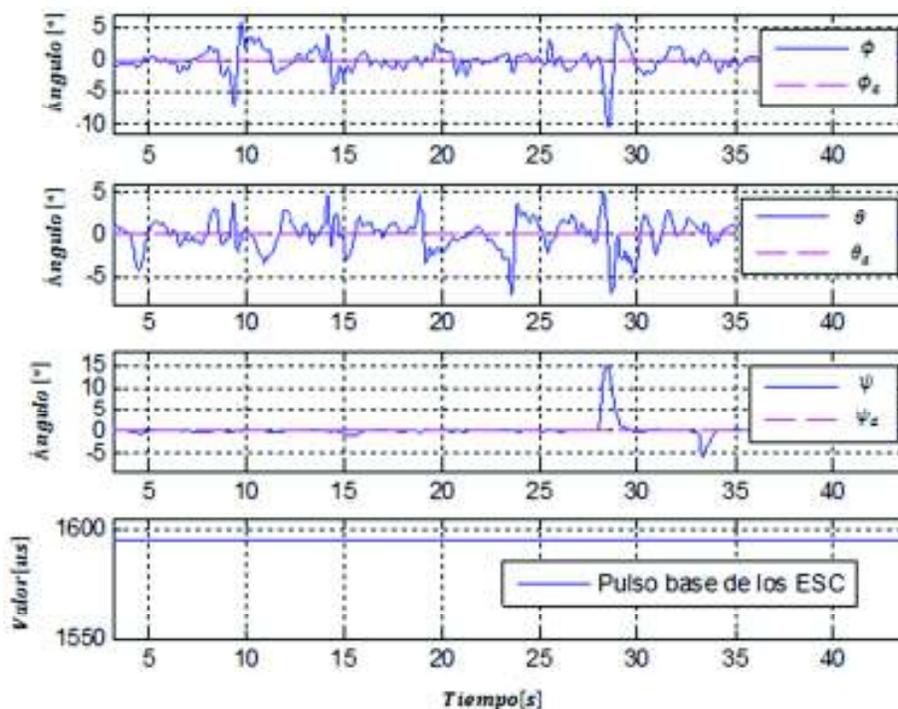


Figura 3.23. Perturbaciones con el control SMC.

Los *setpoints* de los ángulos de orientación se mantuvieron en cero y se ingresaron perturbaciones en roll, pitch y yaw; en ese orden.

La Figura 3.23 y Figura 3.24 muestran la respuesta del control SMC y del control PID ante las perturbaciones respectivamente. Se puede observar que el control PID en comparación con el SMC muestra un tiempo mayor de establecimiento luego de las perturbaciones y las desviaciones del ángulo a controlar son mayores cuando se introducen las perturbaciones.

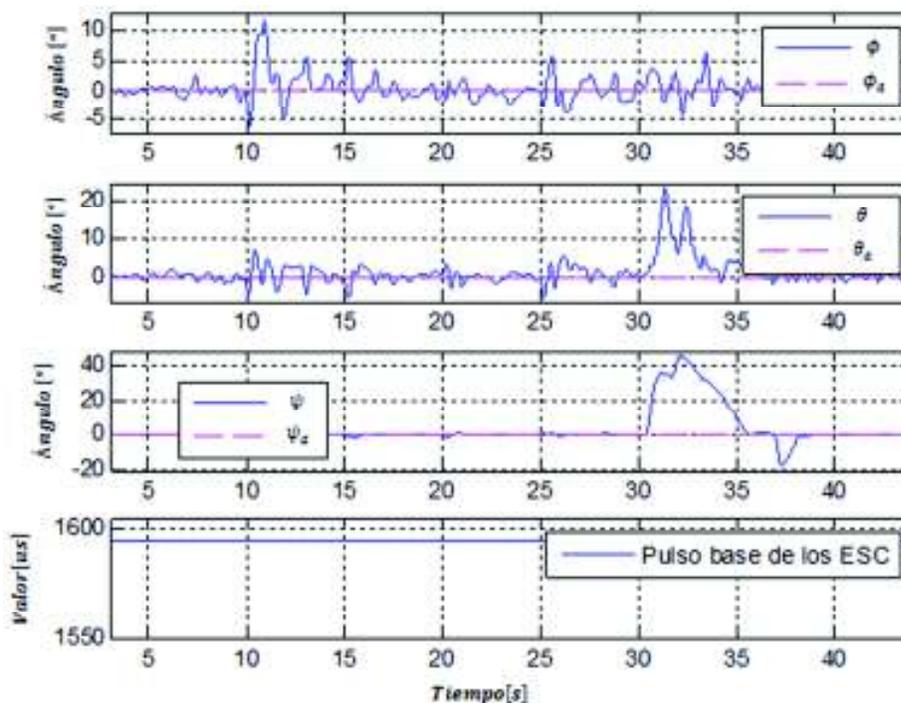


Figura 3.24. Perturbaciones con el control PID.

3.2 DISCUSIÓN

El control SMC en comparación con el control PID mostró un mejor seguimiento de los *setpoints* angulares y un mejor rechazo a las perturbaciones. Con el control SMC si alguna perturbación causaba oscilaciones visibles éstas desaparecían de forma casi inmediata logrando una estabilización más rápida en comparación con el control PID.

El uso de un término integral permitió que los controladores sean capaces de mantener los *setpoints* de los ángulos.

El uso de un *setpoint* de velocidad permitió que la acción de control discontinuo no tuviera que ser muy brusca (tomar valores altos) para llevar al ángulo a su *setpoint* y además permitió que el *setpoint* del ángulo sea alcanzado más rápidamente.

El *setpoint* de aceleración permitió que las velocidades de los ángulos se redujeran considerablemente, permitiendo así un mejor seguimiento del *setpoint* de velocidad y por tanto mejorando el seguimiento del *setpoint* angular.

Los cambios en la velocidad de los motores ocasionan que el dron se eleve demasiado y sea difícil frenarlo debido a que en muchos de los casos al elevarse también se desplaza hacia alguna dirección.

Mientras menor sea la altura de vuelo la turbulencia del aire impulsado por los motores empuja al dron hacia cualquier dirección y el usuario debe desplazar el hexarotor si desea mantener la posición, esto se debe a que no hay control de posición.

Si se desea mejorar el tiempo de vuelo del hexarotor se debería cambiar el conjunto de motores y hélices por unos que permitan generar mayor empuje con menores cantidades de energía.

La reacción del operador debe ser rápida para evitar que el dron pierda altura y a su vez que éste no se eleve demasiado, esto se debe a que durante los desplazamientos parte del empuje que mantenía la altura se direcciona a realizar los desplazamientos por lo que se perderá altura. Al volver a la posición de cero grados en roll y pitch la altura empezará a recuperarse.

Debido al desgaste de la batería y a que los ESC no consideran este desgaste, con el tiempo el hexarotor irá perdiendo altura y el usuario deberá compensar esto manualmente.

El cambio en uno de los ángulos afecta el comportamiento de los demás provocando pequeñas desviaciones que a simple vista pasan desapercibidas, esto se debe al acoplamiento existente entre las variables.

El trabajo realizado por [5], indica las diferentes dificultades por las cuales un control SCM para orientación resulta difícil de implementar en sistemas reales dando a notar que el fenómeno *chattering* influye mucho en la implementación. Esto se debe a que en los sistemas reales existen retardos entre el cálculo de la acción de control y la acción de los actuadores, lo que provoca que la acción de control discontinuo genere *chattering*. Además los modelos no son completos por lo que siempre existirán dinámicas no modeladas, en especial las dinámicas de alta frecuencia empezarán a ser relevantes ya que serán excitadas por el *chattering*.

Con las vibraciones los cables de conexión y los tornillos de la estructura se aflojan, por esto a los cables hay que sujetarlos bien y revisarlos antes de encender el dron y a los tornillos ajustarlos periódicamente.

Los índices de desempeño ISE y el cálculo del porcentaje de diferencia de los mismos muestran una superioridad del control SMC sobre el PID en cuanto a la capacidad de seguimiento de los *setpoints* y de la respuesta ante perturbaciones.

4 CONCLUSIONES

Después de la revisión bibliográfica se escogió el modelo dinámico del hexarotor descrito en este proyecto debido a que éste no considera la fuerza de arrastre ni la fuerza de la resistencia del aire; lo cual simplifica en gran medida el diseño de las ecuaciones de control SMC.

Se usó una superficie deslizante con componente integral para el diseño de los controladores SMC porque la parte integral ayuda a corregir perturbaciones y a mantener los *setpoints*.

El tener un término derivativo; como es el caso del *setpoint* de aceleración, permite tener velocidades más bajas y por tanto un mejor seguimiento de los *setpoints*.

Se pudo visualizar por medio de los datos obtenidos que el hexarotor requiere que en su controlador exista una parte integral para mantener con mayor precisión y con bajos esfuerzos los *setpoints*.

El uso de una tarjeta embebida da apertura a que el usuario implemente diferentes algoritmos de control y a su vez se tiene mayor libertad para agregar hardware.

Filtrar las medidas del giroscopio y del acelerómetro usando filtros de alisado de Kalman permite reducir el ruido en las mediciones de los mismos.

El método heurístico usado en la sintonización de los controladores permitió encontrar los valores adecuados de los parámetros de los controladores para que el error en el control angular no supere los 10° .

En las etapas iniciales de sintonización de los parámetros se puede observar con mayor claridad como el variar el parámetro de un controlador afecta el comportamiento de los demás ángulos. Esto ocurre a causa de que el hexarotor es un sistema multivariable con sus variables acopladas.

El control SMC en comparación con el control PID mostró un mejor seguimiento de los *setpoints* angulares y un mejor rechazo a las perturbaciones.

Los índices de desempeño ISE mostraron que el control SMC corrige los errores de seguimiento de los *setpoints* de orientación en menor tiempo que el control PID.

La oscilación de las variables controladas alrededor de sus *setpoints* es menor cuando se usa un control SMC.

El porcentaje de diferencia negativo entre los índices ISE mostró que el control SMC en todos los ángulos de orientación siempre presenta un mejor desempeño tanto en el seguimiento de los *setpoints* como en la respuesta ante las perturbaciones.

5 REFERENCIAS BIBLIOGRÁFICAS

- [1] N. Opfer y D. Shields, «Unmanned Aerial Vehicle Applications and Issues for Construction,» de *121st ASEE Annual Conference & Exposition*, Indianapolis, 2014.
- [2] J. Calvo, A. Escoda, G. Serra y C. Blanco, «Drones militares: la Guerra de Video Juegos con Víctimas reales,» Barcelona, Centre Delàs d'Estudis per la Pau, Justícia i Pau, 2014.
- [3] C. Romero y A. Jairo, *Uso de Drones en Logística para Entrega de Mercancías*, Bogotá: Universidad Militar Nueva Granada, Ensayo de Grado, 2015.
- [4] E. Aced Félez, «Drones: una nueva era de la vigilancia y de la privacidad,» Marzo 2013. [En línea]. Available: <http://www.redseguridad.com/especialidades-tic/proteccion-de-datos/drones-una-nueva-era-de-la-vigilancia-y-de-la-privacidad>. [Último acceso: 11 Octubre 2016].
- [5] M. Hicham , S. Adil y M. Moussid, «Dynamic Modeling and Control of hexarotor using Linear and Nolinear Methodos,» *International Journal of Applied Information Systems*, vol. 9, nº 5, pp. 9 - 17, Agosto 2015.
- [6] Universidad Valencia IRTIC, «Aplicaciones y Operaciones con Drones/RPAS,» Universidad Valencia IRTIC, Julio 2015. [En línea]. Available: <http://drones.uv.es/origen-y-desarrollo-de-los-drones/>. [Último acceso: 7 Octubre 2016].
- [7] ELDRONE, «La historia de los drones,» ELDRONE, 28 Mayo 2016. [En línea]. Available: <http://eldrone.es/historia-de-los-drones/>. [Último acceso: 3 Octubre 2016].
- [8] ECURED, «Vehículo aéreo no tripulado,» ECURED, 8 Junio 2009. [En línea]. Available: https://www.ecured.cu/Veh%C3%ADculo_a%C3%A9reo_no_tripulado. [Último acceso: 14 Octubre 2016].
- [9] E. Santana, «XDRONES,» 12 Marzo 2017. [En línea]. Available: http://www.xdrones.es/tipos-de-drones-clasificacion-de-drones-categorias-de-drones/#Tipos_de_dronessegun_laaplicacion. [Último acceso: 08 Septiembre 2017].
- [10] J. R. Baldeón Osorio y J. L. Escorza Córdor, *Automatización de un hexacóptero para despegue, aterrizaje y vuelo en un camino cerrado.*, Quito: Proyecto de titulación de pregrado, Escuela Politécnica Nacional, 2015.
- [11] C. A. Cevallos Zambonino y P. A. García Veléz , *Diseño e implementación de un sistema de auto-piloto para el control de un hexacóptero*, Quito: Proyecto de titulación de pregrado, Escuela Politécnica Nacional, 2016.
- [12] J. Fogelberg, *Navigation and Autonomous Control of a Hexacopter in Indoor Environments*, Lund: Media - Tryck, 2013.
- [13] J. Case y E. Boldt, «DJI ESC and Brushless Motor,» Drutal, 2016. [En línea]. Available: <http://robotic-controls.com/book/export/html/65>. [Último acceso: 30 Septiembre 2016].

- [14] B. Akin y M. Bhardwaj, «Texas Instruments,» Septiembre 2015. [En línea]. Available: <http://www.ti.com/lit/an/sprabq7a/sprabq7a.pdf>. [Último acceso: 12 Julio 2017].
- [15] TR3SDLAND, «Tr3sdland - Datasheet del Arduino DUE,» 2014. [En línea]. Available: <https://www.tr3sdland.com/2012/11/llega-arduino-due-con-arquitectura-de-32-bit/>. [Último acceso: 11 Septiembre 2016].
- [16] InvenSense, «Invensense - Datasheet del MPU6050,» 19 Agosto 2013. [En línea]. Available: <https://www.invensense.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>. [Último acceso: 12 Noviembre 2016].
- [17] VANT BATTERY, «Vant Battery - Product catalog,» Vant Battery, 2005. [En línea]. Available: <http://www.vantbattery.com/product.asp?bascode=102&html=Hot-Products>. [Último acceso: 6 Octubre 2016].
- [18] TURNIGY, «Turnigy - Product catalog,» Turnigy, 2017. [En línea]. Available: <http://www.turnigy.com/batteries/nano-tech/>. [Último acceso: 11 Octubre 2016].
- [19] Silicon Laboratories, «Spark Fun - Datasheet del convertor USB a UART CP2102,» 10 Abril 2004. [En línea]. Available: <https://www.sparkfun.com/datasheets/IC/cp2102.pdf>. [Último acceso: 28 Noviembre 2016].
- [20] Genius, «Genius - Datos técnicos del control G-08XU,» 1 Agosto 2017. [En línea]. Available: <http://us.geniusnet.com/supports/maxfire-g-08xu>. [Último acceso: 12 Enero 2017].
- [21] P. Prieto, . N. Cazarez-Castro y L. Aguilar, *Estudio de Métodos de Reducción de Chattering en Modos Deslizantes*, México: Congreso Nacional de Control Automático, AMCA , 2015.
- [22] Universidad del Cauca, «Sistemas de estructura variable, Sistemas no lineales,» 2009. [En línea]. Available: <ftp://ftp.unicauca.edu.co/Facultades/FIET/DEIC/Materias/Nolineales%20Master/Clase%208/clase%208.pdf>. [Último acceso: 27 Octubre 2016].
- [23] J.-J. Slotine y L. Weiping, *Applied Nonlinear Control*, New Jersey: Prentice Hall, 1991.
- [24] R. Rascón Carmona, A. Calvillo Téllez, J. d. D. Ocampo Díaz y J. R. Cárdenas - Valdez, «Control por Modos Deslizantes Mejorado,» *Investigación Básica y Aplicada*, vol. 4, nº 7, pp. 2 - 9, 2015.
- [25] J. Terán, J. Anderez y O. Camacho, «Genetics algorithms as a tuning tool for PID and sliding mode controllers,» *Revista Técnica de la Facultad de Ingeniería Universidad del Zulia*, vol. 28, nº 3, pp. 28-32, 2005.
- [26] C. A. Smith y A. B. Corripio, *Principles and Practice of Automation and Process Control*, New York: John Wiley & Sons, 2006.
- [27] E. Iglesias, J. García, O. Camacho, S. Calderón y A. Acosta, «Ecuaciones de Sintonización para Controlador por Modos Deslizantes y Control de Matriz Dinámica a partir de un Módulo Difuso,» *AXIOMA*, vol. 1, nº 14, pp. 14 - 24, 2015.

- [28] H. t. M. Nabil Elkholy, *Dynamic Modeling and Control of a Quadrotor Using Linear and Nonlinear Approaches*, Cairo: The American University in Cairo, 2014.
- [29] C. A. Smith, *Automated Continuous Process Control*, Florida: Wiley Interscience, 2002.
- [30] C. Smith y A. Corripio, *Control Automático de Procesos*, México: Noriega, 1991.
- [31] P. J. Van de Maele, «Reading a IMU Without Kalman: The Complementary Filter,» Pieter-Jan.com Creaty in Automation y More, 26 Abril 2013. [En línea]. Available: <http://www.pieter-jan.com/node/11>. [Último acceso: 4 Febrero 2017].
- [32] D. P. Vigouroux Cavolina, *Implementación de Unidad de Mediciones Inerciales (IMU) para Robótica utilizando Filtro de Kalman*, Sartenejas: Proyecto de titulación de pregrado, Universidad Simón Bolívar, 2010.
- [33] 5Hertz, «Introcucción al giroscopio,» Tutoriales de 5Hz Electrónica, 21 Febrero 2014. [En línea]. Available: <http://5hertz.com/tutoriales/?p=431>. [Último acceso: 6 Septiembre 2016].
- [34] R. I. Rincon-Jara, R. Ambrosio y J. Mireles, «Análisis y caracterización de un acelerómetro capacitivo fabricado con tecnología polymump's,» *Superficies y Vacío*, vol. 23, nº 3, pp. 26 - 31, 2010.
- [35] 5Hertz, «ABC del acelerómetro,» Tutoriales de 5Hz Electrónica, 17 Febrero 2014. [En línea]. Available: <http://5hertz.com/tutoriales/?p=228>. [Último acceso: 6 Septiembre 2016].
- [36] G. Welch y G. Bishop, «An Introduction to the Kalman Filter,» Universidad de Carolina del Norte, Departamento de Ciencias de la computación, 2001. [En línea]. Available: http://www.cs.unc.edu/~tracker/media/pdf/SIGGRAPH2001_CoursePack_08.pdf. [Último acceso: 1 Diciembre 2016].
- [37] INELMATIC, «<http://www.inelmatic.com>,» 26 Noviembre 2015. [En línea]. Available: <http://www.inelmatic.com/web/wp-content/uploads/2015/11/filtrodekalmán1.pdf>. [Último acceso: 12 Octubre 2016].
- [38] Arduino, «Arduino - Documentación en línea,» 2017. [En línea]. Available: <https://www.arduino.cc/en/Reference/HomePage>. [Último acceso: 11 Diciembre 2016].
- [39] J. Brokking, «MPU-6050 6dof IMU for auto-leveling multicopters,» Brokking, 21 Mayo 2016. [En línea]. Available: <http://www.brokking.net/imu.html>. [Último acceso: 18 Enero 2017].
- [40] ArduPilot Dev Team, «Learning the ArduPilot Codebase,» ArduPilot Dev Team, 2016. [En línea]. Available: <http://ardupilot.org/dev/docs/apmcopter-programming-attitude-control-2.html>. [Último acceso: 5 Diciembre 2016].
- [41] OpenTX University, «CleanFlight PID Tuning,» OpenTX University, 24 Mayo 2016. [En línea]. Available: <http://open-txu.org/home/special-interests/multirotor/cleanflight-pid-tuning/>. [Último acceso: 14 Octubre 2016].

- [42] B. Postlethwaite, «University of Strathclyde - Department of Chemical and Process Engineering,» 22 Julio 2009. [En línea]. Available: http://www.online-courses.vissim.us/Strathclyde/measures_of_controlled_system_pe.htm. [Último acceso: 10 Julio 2017].

ORDEN DE EMPASTADO