

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

ENSAMBLAJE Y CONTROL DE UN CUADRICÓPTERO

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA Y CONTROL

WILLIAM OSWALDO CHAMORRO HERNÁNDEZ
(willyozc_192@hotmail.com)

JORGE LUIS MEDINA MORA
(jnr_777@yahoo.com)

DIRECTOR: NELSON GONZALO SOTOMAYOR OROZCO, MSc.
(nelson.sotomayor@epn.edu.ec)

Quito, Abril 2013

DECLARACIÓN

Nosotros, Chamorro Hernández William Oswaldo y Medina Mora Jorge Luis, 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.

Chamorro Hernández William Oswaldo

Medina Mora Jorge Luis

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Chamorro Hernández William Oswaldo y Medina Mora Jorge Luis, bajo mi supervisión.

Nelson Sotomayor, MSc.

DIRECTOR DEL PROYECTO

AGRADECIMIENTO

Agradezco a Dios por darme fuerzas y sabiduría para poder cumplir todos mis objetivos sin nunca mirar atrás y sin desampararme.

A mi familia por su apoyo incondicional y motivación en cada momento a lo largo de todo el trayecto recorrido.

A mi novia Andrea por su disposición para brindarme siempre ayuda y ánimo cuando más lo he necesitado.

A mi amigo y compañero Jorge Luis por su apoyo y tenacidad para realizar este proyecto.

A mis amigos y compañeros que han compartido este duro trayecto desde los primeros semestres.

Al Ing. Nelson Sotomayor O. Msc, por su guía y colaboración prestada para el desarrollo del presente proyecto.

William

DEDICATORIA

Dedico este trabajo a mis padres Oswaldo e Inés por sus sabios consejos y sacrificios que han hecho, por guiarme y ayudarme a conseguir mis metas.

A mi abuelita Rosa por creer siempre en mí y alentarme en los momentos difíciles.

A mis padrinos Carlos y Lucía y a toda mi familia por alentarme a seguir siempre adelante y a nunca dejar de perseguir mis sueños.

William

AGRADECIMIENTO

Este es el espacio para ser justo y consecuente con todas aquellas personas que creyeron en mí y me apoyaron en los altibajos y durante cada momento de mi carrera universitaria.

Primero quiero agradecer a Dios ya que su mano invisible siempre ha sabido guiarme en la vida, a mis padres Luis y Nancy por ser un vivo ejemplo de amor, honor y humildad, a mis hermanos Nancy y Fabricio por compartir preciosos momentos conmigo durante todo este trayecto, a mis abuelos, tíos, primos y demás familiares que me enseñaron el significado de “Todos para uno y uno para todos”.

A todas mis amigas y amigos que he hecho a lo largo de esta etapa de mi vida dentro y fuera de la Escuela Politécnica Nacional, con quienes he pasado experiencias irrepetibles e invaluable que han forjado un lazo fuerte de familiaridad, que espero y confío se mantenga con los años.

A aquellas personas que en algún momento formaron parte de mi vida, gracias a quienes pude aprender y asimilar muchas cosas y crecer como persona.

A la Gloriosa Escuela Politécnica Nacional ya que formar parte de su alumnado ha sido siempre una fuerte motivación para lograr mis ideales.

A mi amigo y compañero de tesis, William por su ahínco y tenacidad y a nuestro tutor Nelson Sotomayor por su valiosa guía y consejos en este proyecto.

Jorge Luis

DEDICATORIA

A mis padres Luis Humberto y Nancy Mercedes por brindarme todo su amor, sacrificio y apoyo incondicional no sólo en este proyecto sino durante toda mi vida, por enseñarme con su ejemplo un sinnúmero de lecciones hermosas, entre ellas una que siempre estará en mi mente y es que “Sin sacrificio no hay Gloria”.

Jorge Luis

CONTENIDO

RESUMEN	IV
PRESENTACIÓN.....	V
CAPÍTULO 1	
GENERALIDADES	1
1.1 Vehículo aéreo no tripulado	2
1.1.1 Definición	2
1.1.2 Historia.....	2
1.1.3 Descripción	4
1.1.4 Clasificación.....	5
1.2 Cuadricóptero o quadrotor	9
1.2.1 Ddefinición	9
1.2.2 Historia del cuadricóptero	9
1.2.3 Principales características	11
1.2.4 Principio de funcionamiento.....	12
1.2.5 Arquitectura del cuadricóptero	14
1.2.5.1 Sensores.....	14
1.2.5.2 Unidad de control.....	15
1.2.5.3 Etapa de potencia.....	15
1.3 Tipos de cuadricópteros usados como plataforma de investigación	16
1.3.1 Cuadricóptero Hummingbird de Ascending Technologies	17
1.3.2 Draganflyer X4	18
1.3.3 Diy drones arducopter	18
1.4 Introducción al cuadricoptero A.R. Drone.....	21
1.4.1 Descripción de la plataforma A.R. Drone	21
1.5 Generalidades de los principales dispositivos electrónicos a emplearse	24
1.5.1 Microcontroladores.....	24
1.5.1.1 Características del Atmega 164p.....	24
1.5.1.2 Características del Attiny85	25
1.5.1.3 Módulos Xbee	26
1.5.1.4 Receptor infrarrojo para sistemas de control remoto	27

CAPÍTULO 2

DISEÑO Y ENSAMBLAJE DE LA PLATAFORMA MÓVIL.....	28
2.1 Partes constitutivas del cuadricóptero parrot A.R. Drone.....	29
2.1.1. Configuraciones de diseño interior y exterior.....	29
2.1.2. Motores.....	30
2.1.3. Baterías del uav.....	31
2.1.4. Cámara frontal.....	32
2.1.5. Sistemas de ayuda a la navegación.....	32
2.1.6. Sistema informático integrado.....	34
2.2 Diseño de la plataforma móvil.....	35
2.2.1 Diseño del circuito emisor.....	36
2.3 Diseño de la plataforma de seguimiento.....	41
2.3.1 Diseño del circuito receptor.....	41
2.3.2 Ensamblaje de la plataforma de seguimiento.....	44
2.4 Plataforma de control.....	45
2.4.1 Interfaz en C#.....	46
2.4.2 Comunicación entre el circuito de recepción infrarrojo y la pc.....	47
2.4.4 Descripción del modo de seguimiento automático.....	48
2.4.3 Descripción del modo de vuelo manual.....	48

CAPÍTULO 3

DESARROLLO DEL SOFTWARE.....	49
3.1 Detección envío y decodificación de datos.....	49
3.1.1 Envío de datos.....	50
3.1.2 Decodificación de los datos en Labview.....	52
3.1.3 Algoritmo de detección.....	54
3.2 Programación del controlador de seguimiento.....	62
3.2.1 Controlador de posición para los ejes x e y.....	65
3.2.1.1 Identificación de los parámetros de las plantas de pitch y roll.....	66
3.2.1.2 Diseño del controlador de pitch y roll por realimentación de estados	71
3.2.1.3 Diagrama de flujo de los controladores de pitch y roll.....	77
3.3 Software transmisor de datos para el seguimiento.....	83
3.3.1 Descripción del archivo dll del software transmisor de datos.....	84
3.3.2 Creación del archivo dll en Labview.....	84

3.3.3 Descripción de la interfaz del controlador	88
3.3.4 Programación en la interfaz de C#	92
3.3.4.1 Programación de los movimientos de datos desde la dll.....	92
CAPÍTULO 4	
PRUEBAS Y RESULTADOS.....	97
4.1 Pruebas de la placa del faro infrarrojo.....	97
4.2 Pruebas de la placa de los receptores infrarrojos	98
4.3 Pruebas de vuelo del cuadricóptero A.R. Drone ensamblado.....	100
4.4 Pruebas del algoritmo de posicionamiento del A.R. Drone.	104
4.4.1 Pruebas de la incertidumbre en el posicionamiento del cuadricóptero A.R. Drone	114
4.4.1.1 Pruebas en el eje y (pitch).....	114
4.4.1.2 Pruebas en el eje x (roll).	117
4.4.1.3 Pruebas simultáneas en el eje X (roll) e Y (pitch).....	119
4.5 Pruebas del algoritmo de seguimiento implementado.....	123
4.6 Lista de elementos y costos	125
4.7 Resultados	127
CAPÍTULO 5	
CONCLUSIONES Y RECOMENDACIONES.....	130
5.1 Conclusiones.....	130
5.2 Recomendaciones	133
REFERENCIAS BIBLIOGRÁFICAS.....	134
ANEXOS	

RESUMEN

En estos años el cuadricóptero se ha convertido en la plataforma estándar para la investigación de micro vehículos aéreos (MAV) y de proyectos con robots aéreos. Esto se ha logrado gracias a la simplicidad y efectividad del diseño de un cuadricóptero, el abaratamiento de costos e incremento en el desempeño y capacidades de los componentes necesarios para su construcción.

Un cuadricóptero es un helicóptero pequeño, generalmente en la escala de 30 cm a 1m de diámetro con 4 propulsores ordenados en pares en configuración “+” o “x”. Cada propulsor es manejado por un motor eléctrico, el cual es alimentado por una batería a bordo.

Estas aeronaves pueden ser autónomas o remotamente operadas. Una computadora o microcontrolador a bordo y una variedad de sensores que incluyen acelerómetros, giroscopios, y magnetómetros son los encargados de mantener la estabilidad del cuadricóptero en el aire.

Es por esto que se ha escogido esta plataforma para implementar un control de seguimiento de un móvil en tierra, dicho seguimiento debe ser autónomo pero con la capacidad de corregir comportamientos anómalos del cuadricóptero por parte de un operador.

PRESENTACIÓN

En el presente trabajo se desarrolla el ensamblaje y control de un cuadricóptero el cual debe seguir a un objeto móvil en tierra.

En el Capítulo 1 se resumen las generalidades de los UAVs así como una breve reseña histórica de su evolución y desarrollo, las características principales de los cuadricópteros comerciales que son utilizados como plataforma para investigación y los fundamentos básicos de los dispositivos usados para el control.

En el Capítulo 2 se detalla el diseño electrónico de la plataforma móvil, así como el acondicionamiento de la instrumentación que va montada sobre el cuadricóptero Parrot A.R. Drone que es la plataforma sobre la cual se va a trabajar, además se muestran las partes constitutivas del mismo previo al ensamblaje y una visión general del software donde se implementó el control.

En el Capítulo 3 se muestra la forma de comunicar el cuadricóptero con una estación fija para realizar el control del mismo de forma remota, los algoritmos de control de seguimiento y detección del móvil, su desarrollo matemático y la forma de implementarlos.

Las diferentes pruebas realizadas para comprobar el correcto funcionamiento del controlador de seguimiento, en los diferentes modos de trabajo tanto manual como automático, se describen en el Capítulo 4.

En el Capítulo 5 se presentan las conclusiones y recomendaciones que se obtuvieron luego del diseño, implementación y pruebas del controlador sobre el cuadricóptero Parrot A.R Drone.

CAPÍTULO 1

GENERALIDADES

Desde que el hombre posó su mirada en el cielo nació su deseo de poder volar. Contemplando la destreza de las aves descubre su fascinación por el vuelo y al mismo tiempo comprende que cuenta con sólo el instrumento de su inteligencia para poder emular lo que tanto admira. Para conquistar los cielos primero realizó modelos a escala que pudieran mantenerse a flote en el aire; durante el Renacimiento, Leonardo Da Vinci buscando revertir la incapacidad humana para volar, expone en su famoso Codex Atlanticus sus descubrimientos; considerados apócrifos en ese tiempo de máquinas voladoras, que anticipaban en cuatro siglos al avión [1].

Con el inicio de la Segunda Guerra Mundial y posteriormente la Guerra Fría se empezó a construir todo tipo de máquinas aéreas como aviones, helicópteros y a la par dispositivos voladores a pequeña escala, que tendrían como objetivo el entretenimiento de sus pilotos. El inventor francés Alphonse Penaud fabricó el primer modelo de un vehículo aéreo no tripulado que se lanzaba con la mano e iba propulsado por bandas de goma retorcidas previamente, consiguiendo en el año 1871 que volase unos 35 metros [1]. Este modelo consistía en un aeroplano con cola fija estabilizadora que armonizaba al mismo tiempo la estabilidad longitudinal. Poseía un motor de goma (banda de caucho) que era puesto en movimiento accionando su hélice de madera liviana.

Los modelos de los vehículos aéreos fueron evolucionando acorde a los recursos del tiempo y de la época, agregándoseles accesorios más sofisticados que el mercado brinda. Por su complejidad, el planeamiento, la construcción y diseño de los modelos a escala estimula el psiquismo de los niños y adultos y en especial la psicomotricidad como educación del movimiento en los más pequeños.

1.1 VEHÍCULO AÉREO NO TRIPULADO

1.1.1 DEFINICIÓN

Un vehículo aéreo no tripulado (UAV: Unmanned Aerial Vehicle), es una aeronave que vuela sin tripulación humana a bordo, que pueden ser controladas desde una estación base o tener un funcionamiento autónomo mediante un algoritmo preestablecido. Se aplica este término a todo tipo de aeronaves, no sólo aviones, que vuelan sin asistencia humana, de ahí que se pueda aplicar la designación a cualquier dispositivo capaz de volar, desde una cometa, un avión manejado por radio, helicópteros, cuadricópteros radio controlados hasta un misil [1].

La nación que cuenta con más sistemas de este tipo y que se encuentra a la vanguardia del desarrollo de los mismos es Estados Unidos, la cual cada día aumenta el número de funciones de estos robots conforme se desarrolla la tecnología. Sin embargo hoy en día el uso de los UAV se limita a tareas de reconocimiento y vigilancia en el campo de batalla o en otro tipo de ambientes peligrosos.

A pesar de haber sido desarrollados inicialmente para aplicaciones militares, los UAV tienen un campo de aplicación importante a nivel civil, ya que podrían usarse aparatos de este tipo para darle seguimiento a incendios forestales o catástrofes de diversos tipos sean o no de origen natural, seguridad civil, como la vigilancia de los oleoductos, también suelen ser preferidos para misiones que son demasiado "aburridas, sucias o peligrosas" para los aviones tripulados [1].

1.1.2 HISTORIA

El desarrollo de los UAV se puede rastrear incluso hasta poco después de la I Guerra Mundial, siendo usados durante la Segunda Guerra Mundial con el fin de entrenar a los soldados que operaban los cañones antiaéreos. En aquellos tiempos de guerra, cuando los austriacos atacaron la ciudad de Venecia, globos no tripulados cargados con explosivos fueron controlados desde una estación remota con el fin de evitar pérdidas humanas durante el ataque.

Los resultados, sin ser los esperados, dejaron como precedente el aumento de capacidades a través de estos ingenios recién creados. Paralelamente se empezó a experimentar con bombas aéreas no tripuladas [1].

Gracias a los avances en telecomunicaciones, particularmente en cuanto a la obtención, manejo y transmisión de información directamente de los campos de batalla, lo que permite conseguir comunicaciones mucho más seguras y difíciles de interferir, se empezó la construcción de aviones radiocontrolados para su uso en los campos de batalla.

Los UAV han demostrado en diferentes escenarios y especialmente en la Guerra del Golfo y en la Guerra de Bosnia, el gran potencial que pueden tener, en cuanto a la obtención, manejo y transmisión de la información, gracias a la aplicación de nuevas técnicas de protección de la misma. El ejército estadounidense se dio cuenta de la importancia y necesidad de emplear este tipo de plataformas, con el fin de manejar eficiente y discretamente su información, de esta manera en la operación Tormenta del Desierto en 1991, la Armada estadounidense utilizó el sistema UAV Pioneer Israelí para suministrar inteligencia a nivel táctico, en Afganistán durante la operación Paz Duradera el sistema UAV Predator realizó misiones de reconocimiento armado y en el 2003 en Irak atacó objetivos de gran valor para la coalición [2].



Figura 1.1 UAV Predator, tomado de [2]

El Ecuador también ha hecho uso de los UAVs, en el 2008 adquirió seis aeronaves no tripuladas dos *Heron* de largo alcance y cuatro *Searcher* de corto alcance del fabricante ISRAEL AEROSPACE INDUSTRIES (IAI), las cuales se

han integrado plenamente al sistema de vigilancia y control del territorio marítimo ecuatoriano, donde la marina y el cuerpo de guardacostas realizan operaciones de lucha contra el narcotráfico, la piratería y el contrabando por parte del crimen organizado.

Con ello Ecuador es uno de los primeros países a nivel sudamericano y latinoamericano que despliega operativamente este tipo de medios [3].



Figura 1.2 UAV Heron, tomado de [3]

1.1.3 DESCRIPCIÓN

Los vehículos aéreos no tripulados en su concepto tradicional está constituido de un conjunto de sensores para la adquisición de la información del medio que lo rodea, y transmitirla hacia una estación base en tierra. Para cumplir con sus objetivos un UAV debe estar conformado por las siguientes partes [1]:

Plataforma aérea.

Está conformado por el vehículo aéreo propiamente dicho además de su conjunto de sensores para adquirir información del medio y transmitirlo en tiempo real, así como la instrumentación necesaria para mantenerlo estable, como son tarjetas IMU (Inertial Measurement Unit), es decir sensores de tipo inercial como son acelerómetros, giroscopios y magnetómetros que deben trabajar en conjunto para obtener resultados óptimos. Un sistema de navegación es indispensable para evitar que el UAV pierda su trayectoria, para este fin estos suelen tener dispositivos GPS, cámaras de video inalámbricas entre otros.

Plataforma terrestre.

Está compuesto por una estación de control terrestre, donde se recibe y procesa la información que envía el UAV en tiempo real, además donde se encuentra todo el material y equipamiento preciso para el lanzamiento, recuperación y puesta en operación del sistema. La estación de control, dependiendo de la complejidad del sistema, puede estar conformada desde un computador hasta una compleja red, que cumple las funciones de permitir la preparación y lanzamiento del UAV, guiado al área del blanco, recepción y transmisión de imágenes, datos, etc.

1.1.4 CLASIFICACIÓN

Existen varias maneras de clasificar a los vehículos aéreos no tripulados. A continuación se presentan los principales aspectos que consideran para su clasificación [4]:

Por el tipo de misión para el que fueron diseñados

Básicamente se agrupan en los siguientes 6 tipos: De simulación de blancos (para confundir al enemigo), reconocimiento de terreno, combate, de logística (para transportar carga), investigación y desarrollo.

Por su alcance

Se pueden encontrar los Handheld que no sobrepasan los 600 metros de altura ni los 2 kilómetros de distancia o los Orbitales que vuelan en órbitas bajas terrestres a altas velocidades (este tipo de vehículo se encuentra en etapa de estudio).

Christian Nadales Real de la Universidad Técnica de Cataluña realiza la siguiente tabla comparativa:

TABLA 1.1 Clasificación de los UAV, tomada de [4]

CATEGORÍA	ACRÓNIMO	ALCANCE (Km)	AUTONOMÍA (h)	ALTITUD	CARGA
				DE VUELO (m)	MÁXIMA (Kg)
Micro <250 gr	Micro	< 10	1	250	<5
Mini <25 Kg	Mini	< 10	< 2	150 y 300	<30
Alcance cercano	CR	10 a 30	2 a 4	3000	150
Alcance corto	SR	30 a 70	3 a 6	3000	200
Alcance medio	MR	70 a 200	6 a 10	5000	1250
Altitud baja	LADP	> 250	0.5 a 1	50 a 9000	350
Altitud media	MRE	> 500	10 a 18	8000	1250
Autonomía alta Altitud baja	LALE	> 500	> 24	3000	<30
Autonomía alta Altitud media	MALE	> 500	24 a 48	14000	1500
Autonomía alta Altitud alta	HALE	> 2000	24 a 48	20000	12000
Combate	UCAV	1500	2	10000	10000
Ofensivo	LETH	300	3 a 4	4000	250
Señuelo	DEC	500	4	5000	250
Estratosférico	STRATO	> 2000	> 48	2000 y 30000	ND
EXO- estratosférico	EXO	ND	ND	> 30000	ND

La mayoría de UAVs que son empleados para aplicaciones civiles, y los que están disponibles para el público en general se encuentran en las categorías de mini, micro y de alcance cercano.



Figura 1.3 UAV micro, tomado de [5]



Figura 1.4 UAV mini, tomado de [6]



Figura 1.5 UAV Alcance medio, tomado de [7]



Figura 1.6 UAV de combate, tomado de [8]

Por el tipo de despegue

En esta clasificación se puede identificar: los de despegue vertical, como helicópteros y dirigibles, y los de despegue no vertical, como aeroplanos y parapente. La Figura 1.7 muestra un resumen de estas aeronaves.

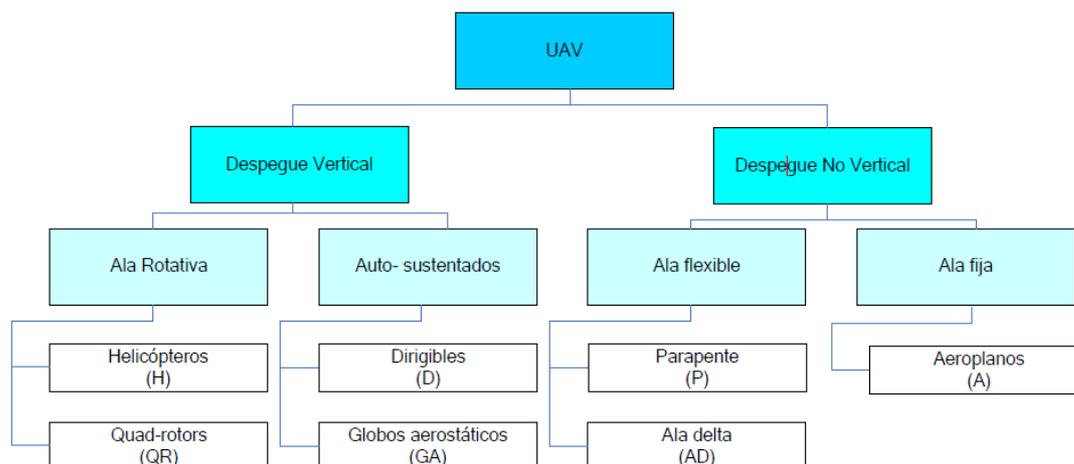


Figura 1.7 Clasificación de los UAVs según el tipo de despegue, tomado de [9]

Cada uno posee sus propias características, importantes de considerar a la hora de elegir un UAV, así por ejemplo los aeroplanos tienen una gran facilidad en el control y guiado, así como su alta fiabilidad ante condiciones extremas o difíciles, además, dependiendo del tipo de controlador que posea se puede obtener un buen alcance, sin embargo su problema más grande es su falta de maniobrabilidad en espacios reducidos. Los vehículos de hélice rotativa son muy adecuados para tareas de interiores, siempre que sean de dimensiones grandes como por ejemplo un estadio o salas de conciertos, a diferencia de los aeroplanos su control es más complejo y no tienen los mismos tiempos de respuesta.

1.2 CUADRICÓPTERO O QUADROTOR

1.2.1 DEFINICIÓN

Un cuadricóptero o quadrotor es un vehículo aéreo no tripulado con la capacidad de elevarse y desplazarse con la propulsión de 4 motores instalados al final de un marco en forma de cruz.

Para realizar el control de un avión se emplean superficies de contacto con el aire denominados alerones, por otro lado el control del cuadricóptero se lo realiza por medio de la variación de la potencia de los motores, lo que provoca variaciones en sus momentos, dando lugar al movimiento que se desea obtener. El sentido de giro de las hélices en dos motores es horario mientras que en otros dos es antihorario, de esta forma se conserva el par motor y evita que gire sobre sí mismo [10].

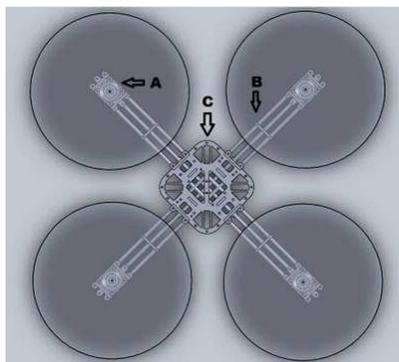


Figura 1.8 Vista superior del cuadricóptero

1.2.2 HISTORIA DEL CUADRICÓPTERO

En el siglo 20 el científico francés Charles Richet en conjunto con Louis Bréguet construyeron un pequeño helicóptero pilotado que era una aeronave de alas giratorias, donde la velocidad aerodinámica responsable de la sustentación proviene principalmente del giro de las palas del rotor, sin embargo este modelo que constaba con 4 alas giratorias como se aprecia en la Figura 1.9, se levantó del suelo en 1907, pero carecía de una suficiente estabilidad y capacidad de mando para efectuar un vuelo controlado.

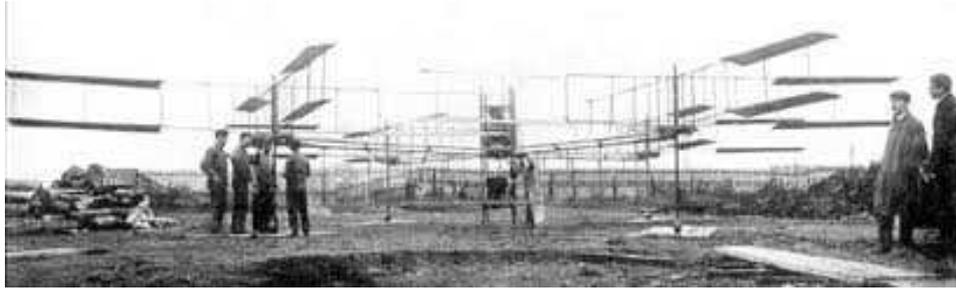


Figura 1.9 Aparato de los hermanos Richet, tomado de [11]

El aparato fue denominado “El Giroplano” de Bréguet – Richet, que básicamente es un cuadricóptero con propulsores de 8.1 metros de diámetro, un peso de 578 Kg incluido los pilotos y con sólo un motor de combustión interna de 50Hp que manejaba los rotores a través de una transmisión de correa y polea. La máquina levantó el vuelo entre agosto y septiembre de 1907, con varios testigos se elevó cerca de 1.5 metros y se mantuvo en vuelo por pocos momentos aterrizando inmediatamente [1].

Posteriormente en 1922 en Francia se crea un prototipo, llamado Convertawings Modelo A, fue diseñado para uso civil y militar. El diseño tenía dos motores que controlaban cuatro rotores. No tenía rotor de cola, ya que usaba la diferencia en las velocidades de giro para lograr el desplazamiento. La Figura 1.10 muestra una imagen del vehículo [1].



Figura 1.10 Convertawings Modelo A, tomado de [11]

Con el advenimiento de nueva tecnología se ha incrementado el interés en el diseño de cuadricópteros, la empresa Bell diseñó un Quad - Tiltrotor, (Figura 1.11)

capaz de llevar una gran carga útil, alcanzando altas velocidades, emplea un pequeño espacio tanto para el despegue como para el aterrizaje y puede modificar la dirección de sus cuatro rotores, hacia delante para tener un vuelo vertical como un avión convencional [10].

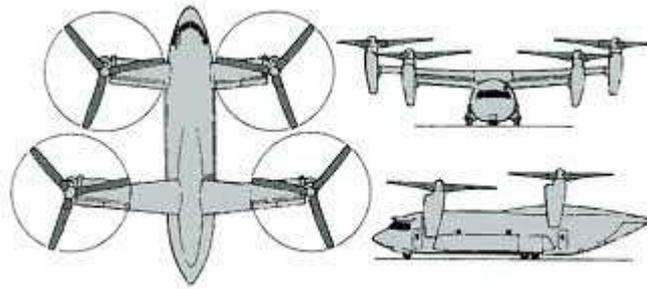


Figura 1.11 Quad Tiltrotor de la compañía BELL, tomado de [12]

El Moller Skycar como se indica en la Figura 1.12 es un famoso cuadricóptero prototipo, a manera de auto volador. Consta de cuatro rotores en forma de ductos de ventilación que permiten una operación segura y eficiente a bajas velocidades. Su inventor Paul Moller lo intentó vender a través de una subasta, sin éxito alguno. Actualmente se ha enfocado en un modelo mejorado el M200G Volantor, que consta de 8 ductos de aire controlados por computador, capaz de levantarse a 3 metros sobre el suelo [13].



Figura 1.12 Moller Skycar, tomado de [13]

1.2.3 PRINCIPALES CARACTERÍSTICAS

Un cuadricóptero; como se mencionó en su definición; es un vehículo aéreo no tripulado que entra en la categoría de ala rotativa, a nivel “mini”. De acuerdo a la Tabla 1.1 esta categoría tiene las características de alcance menor a 10 Km, una

altitud de vuelo menor a 300 m, una autonomía menor a 2 horas con la capacidad de llevar una carga máxima de 5 Kg. Aparte de las características técnicas propias de la categoría a la cual pertenece se pueden destacar las siguientes [16]:

Maniobrabilidad.

Los cuatro rotores del vehículo permiten una mayor exactitud en su control, permitiendo ser empleado en sitios con espacio reducido y navegación en interiores.

Capacidad de vuelo vertical

Al igual que los helicópteros los cuadricópteros poseen esta capacidad, esta característica resulta ventajosa cuando se desea realizar un vuelo estacionario en lugar de un vuelo horizontal. La autonomía de vuelo de los cuadricópteros a nivel mini no suele ser muy buena (< 2 h), incluso puede estar en el orden de los minutos ésta fue una de las limitaciones por la que los UAV tardaron un cierto tiempo en avanzar [16].

Actualmente se están realizando avances importantes en las baterías, proporcionando más capacidad (mAh) y reduciendo los tamaños.

1.2.4 PRINCIPIO DE FUNCIONAMIENTO

En el cuadricóptero cada rotor es responsable de una cierta cantidad de torque y empuje en su centro de rotación. Sus hélices no son todas iguales, en efecto se dividen en dos pares: dos de empuje y dos de desplazamiento, que trabajan en contra rotación. Como consecuencia, el torque neto resultante puede ser nulo si todas las hélices giran con la misma velocidad angular, permitiendo que el vehículo se mantenga estable en el aire alrededor de su centro de gravedad [9].

Con el propósito de definir la orientación en la que se mueve el cuadricóptero (o actitud) alrededor de su centro de masa, los ingenieros aeroespaciales han definido tres parámetros dinámicos: los ángulos de yaw (guiñada o giro en torno a

un eje), pitch (cabeceo o movimiento hacia adelante y atrás) y roll (balanceo o movimiento de derecha a izquierda), siendo estos ángulos muy útiles para controlar las fuerzas que actúan sobre el vehículo aéreo al momento de su traslación o rotación [14].

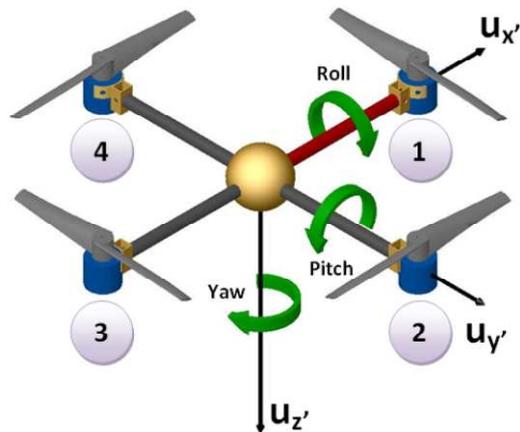


Figura 1.13 Ángulos de yaw, pitch y roll de un cuadricóptero común, tomado de [8]

Un cambio en el ángulo de pitch se produce por la variación de velocidades de los motores 1 y 3 en la Figura 1.13, disminuyendo la velocidad angular del motor 3 y aumentando la del 1 o viceversa, se tendrá como resultado un movimiento de traslación hacia adelante o atrás. De la misma manera si se varia la velocidad angular en los motores 2 y 4 en la Figura 1.13 se producirá una variación en el ángulo de roll, teniendo como resultado un movimiento de traslación hacia la derecha o hacia la izquierda. El movimiento de yaw, o alrededor de su propio eje, es inducido por un desbalance aerodinámico de los torques, si se incrementa la velocidad de los motores 1 y 3 y se disminuye la de los motores 2 y 4, se produce el desbalance mencionado y el cuadricóptero gira en torno a su propio eje en sentido antihorario. De esta manera cambiando estos tres ángulos se tendrá la capacidad de dirigir al vehículo en cualquier dirección y posicionarlo en un punto con gran precisión.

1.2.5 ARQUITECTURA DEL CUADRICÓPTERO

Para analizar la arquitectura del cuadricóptero se lo dividirá en bloques como se aprecia en la Figura 1.14

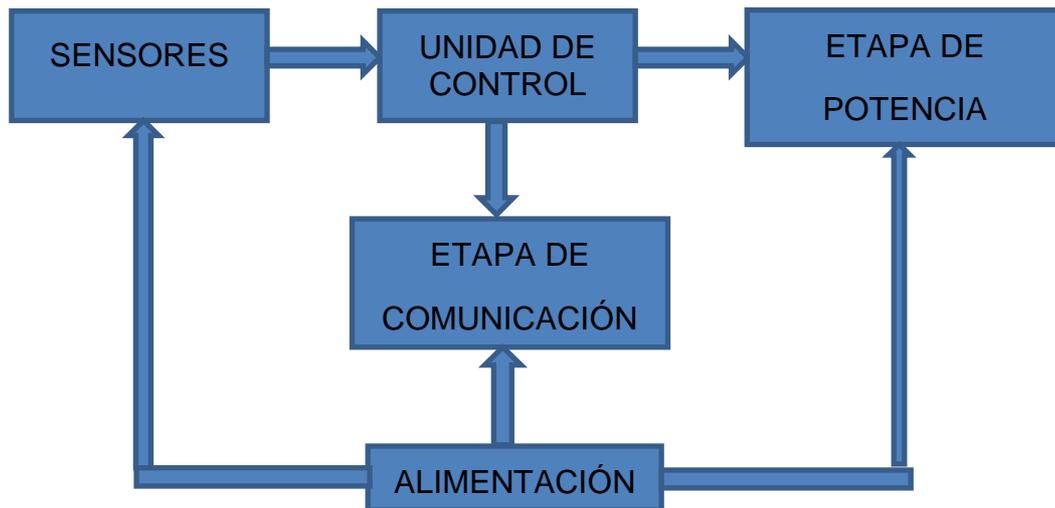


Figura 1.14 Arquitectura genérica del cuadricóptero

1.2.5.1 Sensores

La instrumentación que permitirá leer las magnitudes físicas involucradas para mantener estable y dar movimiento al cuadricóptero y además convertirlas a un valor digital para poder controlar los motores son los siguientes:

- Acelerómetro: mide la aceleración asociada a la variación de posición.
- Giroscopio: mide la aceleración angular.
- Magnetómetro: o brújula digital, empleado para no perder la orientación.
- Sensores ultrasónicos: permiten detectar y estimar la proximidad de un objeto, generalmente son usados tanto para el despegue como para el aterrizaje, y estimar la altura del vehículo.

La señal de los sensores es de tipo analógica (variaciones de voltaje), para poderlas procesar es necesario digitalizarlas a través de un conversor A/D. Un microprocesador será el encargado de procesar las señales digitales y con los resultados calcular la respuesta que se debe enviar a los motores. Este conjunto forma un sistema de lazo cerrado pues la realimentación se produce a través del entorno físico por el que se mueve, variando sus magnitudes dependiendo de los requerimientos de altura o posición que sean demandados.

1.2.5.2 Unidad de control

Los UAVs y en especial los cuadricópteros requieren de sensores inerciales para controlar tanto los movimientos como la estabilidad. Los datos provenientes de los sensores son procesados en varios microprocesadores de alta velocidad para evitar centralizar el control del sistema, ganando así velocidad de respuesta.

El uso de cámaras resulta atractivo hoy en día, los controles basados en visión artificial son temas de investigación en muchos laboratorios del mundo así por ejemplo Michael Moguenson del "Tufts University Center for Engineering Education and Outreach" realizó su tesis previo a la obtención del título de "Master of Science" de control de seguimiento e identificación de rostros con un cuadricóptero utilizando visión artificial en LabView.

Tanto la parte sensorial como el sistema de video integrado al cuadricóptero actúan en conjunto para brindar estabilidad y controlabilidad al sistema, incluso se han desarrollado colaboradores automáticos para hacerlo más fácil de manejar y hacerlo atractivo a usuarios inexpertos.

1.2.5.3 Etapa de potencia

Los cuadricópteros generalmente emplean motores tipo brushless que se puede traducir como "sin escobillas" para dar propulsión al sistema. Las escobillas son los elementos que hacen contacto en el colector de un motor común. En los motores de DC más pequeños, son de una aleación de cobre y en motores más grandes son de un compuesto a base de carbón.

Los motores brushless en vez de funcionar con DC funcionan con AC, la mayoría se alimentan con una señal trifásica, esta señal idealmente debería ser sinusoidal, pero en la práctica son pulsos, sin embargo se los clasifica como motores de DC porque al igual que los motores comunes tienen imanes permanentes. Los imanes son atraídos por la polaridad de un campo magnético generado en las bobinas, las cuales como se explicó anteriormente reciben pulsos en un patrón específico. Si se desea que el motor gire más rápido, simplemente se hace girar el campo magnético secuencial a mayor velocidad, es decir sería necesario aumentar la frecuencia de los pulsos. Un control de velocidad adecuado de los motores permitirá al cuadricóptero desplazarse en cualquier dirección.



Figura 1.15 Motor brushless, tomado de [17]

1.3 TIPOS DE CUADRICÓPTEROS USADOS COMO PLATAFORMA DE INVESTIGACIÓN

A lo largo de varios centros de investigación tanto en Ecuador como a nivel internacional se está usando cuadricópteros como plataformas aéreas para investigar nuevas y novedosas técnicas de control que van desde un control a distancia, hasta complejas aplicaciones militares las cuales requieren de compleja instrumentación y la capacidad para responder rápidamente ante estímulos externos.

La mayoría de plataformas no están disponibles para el público en general, su complejidad implica un elevado costo que los hace poco asequibles y son

destinados principalmente a laboratorios de investigación. Los cuadricópteros más usados en laboratorios de investigación se describen en los puntos siguientes.

1.3.1 CUADRICÓPTERO HUMMINGBIRD DE ASCENDING TECHNOLOGIES

El AscTec Hummingbird es el UAV que mejor ha sido desarrollado por Ascending Technologies, una empresa alemana, diseñado para realizar agresivas y rápidas maniobras de vuelo. Su marco robusto y hélices flexibles le permiten realizar aterrizajes bruscos y su estructura simple lo hace fácil de reparar. Esto lo convierte en la perfecta herramienta para investigar nuevas técnicas de control de vuelo y maniobras dinámicas [18].

Entre sus características se pueden resaltar las siguientes [18]:

1. Ligero en peso y ágil, pesa 510gr y es capaz de llevar una carga de hasta 200gr.
2. Programable vía AscTec SDK, además que consta de su propio toolkit para Simulink.
3. Acelerómetro de 3 ejes, giroscopio de 3 ejes, magnetómetro de 3 ejes, sensor de presión barométrica y GPS.
4. Costo en el mercado de aproximadamente \$5000.00.

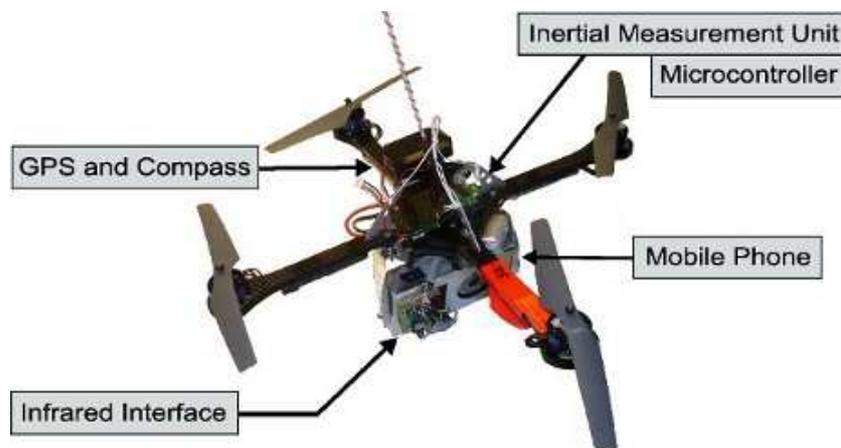


Figura 1.16 AscTec Hummingbird, tomado de [18]

1.3.2 DRAGANFLYER X4 [19]

El Draganflyer X4 es un UAV que ha sido diseñado para llevar cámaras de video inalámbricas cuyo propósito es realizar fotos y videos aéreos de alta calidad, esto es posible ya que la cámara HD que lleva a bordo es giro estabilizada en los ejes de Pitch con lo que es aislada de la vibración de su estructura.

El UAV posee 7 sensores de vuelo que le permite auto estabilizarse en el aire, como acelerómetros, giroscopios, sensores de presión barométrica, etc trabajando todo en conjunto con un avanzado control de vuelo que permite al usuario concentrarse en capturar la imagen deseada, todo se realiza mediante un control remoto de radiofrecuencia. Construida a partir de fibra de carbono de alta calidad, la relación de resistencia a peso de la aeronave es muy alta, junto con un bastidor plegable fuerte ayuda a reducir los daños en caso de una desaceleración brusca o golpear una obstrucción. Su costo en el mercado bordea los \$12000.



Figura 1.16 Draganflyer X4, tomado de [14]

1.3.3 DIY DRONES ARDUOPTER [20]

ArduCopter es una herramienta fácil de configurar y de volar sin embargo sus funciones van mucho más allá que las de un simple helicóptero controlado remotamente, es completo UAV, que ofrece control remoto y vuelo autónomo además de incluir una completa telemetría que se muestra en una poderosa estación terrestre. ArduCopter está en la vanguardia de la robótica aérea y destinada a aquellas personas que quieran probar la tecnología avanzada, las principales técnicas y nuevos estilos de vuelo.

El proyecto Arducopter se basa en el ArduPilot Mega 2, un piloto automático creado por la comunidad DIY Drones. Entre sus características se pueden notar las siguientes:

1. Control de nivel y de altura de alta calidad que le permite volar nivelado y recto. Posee un modo de "vuelo sencillo", lo que hace a ArduCopter más fácil de pilotar, el cuadricóptero se orienta automáticamente y se auto nivela permitiéndole al usuario únicamente deslizarlo en el aire en cualquier dirección.
2. No es necesario programar la tarjeta Arduino, mediante una aplicación la PC es posible controlarlo y observar a la vez la interfaz que provee información valiosa como datos de altitud, nivelación, etc.
3. Posee un GPS integrado que permite posicionar al Arducopter en su interfaz, además de acelerómetros y giroscopios de 3 ejes empleados para su control.
4. Posee una estructura reforzada con aluminio y fibra de carbono la cual protege a los circuitos a bordo de posibles daños en caso de que se presente aterrizajes bruscos.
5. Su costo en el mercado bordea los \$860.00.

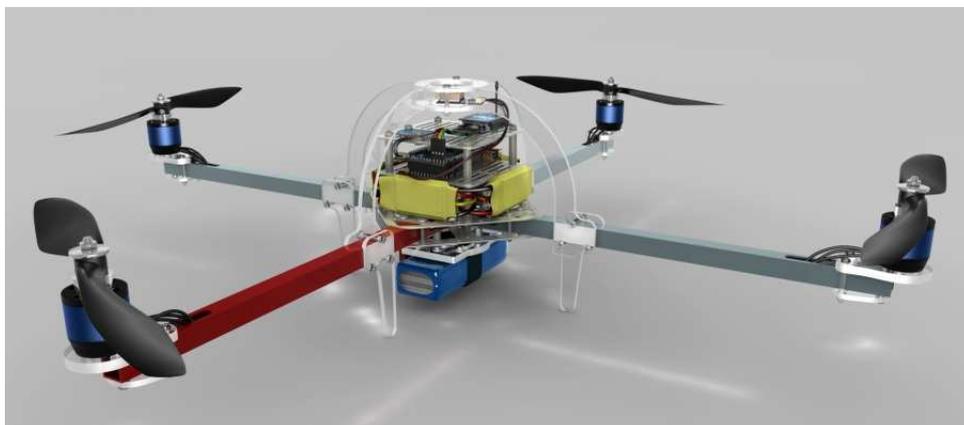


Figura 1.16 Arducopter, tomado de [20]

Las principales características de los cuadricópteros comerciales incluyendo las de cuadricóptero Parrot A.R. Drone que es la plataforma sobre la que se va a implementar el control, cuyas características se detallarán más adelante, se observan en la Tabla 1.2.

Tabla 1.2 Cuadro comparativo entre cuadricópteros comerciales

Cuadricóptero	Características	Precio (U.S.A)
Parrot A.R. Drone	Acelerómetro y giroscopio de 3 ejes, altímetro ultrasónico, 1 cámara frontal y 1 cámara inferior.	\$300.00
AscTec Hummingbird	Acelerómetro, giroscopio y magnetómetro de 3 ejes, sensor barométrico de presión y GPS	\$5000.00
Dragan Fly	Acelerómetro y giroscopio de 3 ejes, sensor barométrico de presión y sistema autoestabilizador para cámaras.	\$12000.00
DIY Drones Arducopter	Acelerómetro y giroscopio de 3 ejes y sistema de control Arduino.	\$ 800.00

1.4 INTRODUCCIÓN AL CUADRICÓPTERO A.R. DRONE

En 2004 la compañía francesa Parrot empezó un proyecto llamado A.R. Drone, el objetivo era producir un micro vehículo aéreo no tripulado o UAV para el mercado masivo de los juegos de video y entretenimiento en el hogar. En este proyecto han participado de 5 a 12 ingenieros de Parrot con el apoyo técnico de ingenieros de SYSNAV y Paris Tech para juntar esfuerzos en la construcción del sistema de navegación y el diseño del control [21].

Una de sus características únicas es que es una plataforma estabilizada aérea, controlada a distancia a través de una interfaz gráfica fácil de usar que se ejecuta a través de un Apple iPhone, iPad o iPod Touch, actualmente la interfaz puede ser ejecutada y modificada mediante Linux o Windows, lo que involucra avanzados conocimientos de programación en aquel sistema operativo.

Este proyecto es un ejemplo perfecto del uso sofisticado de sensores de relativamente bajo costo. Los algoritmos integrados en la plataforma AR Drone tienen la particularidad de ser muy estables, robustos, la plataforma es capaz de vuelo estacionario y vuelo de avance rápido en espacios estrechos.

1.4.1 DESCRIPCIÓN DE LA PLATAFORMA A.R. DRONE [15]

El vehículo aéreo Parrot A.R. Drone se basa en un diseño quadrotor clásico. Cuatro motores sin escobillas se utilizan para mover las 4 hélices. Cada motor está montado sobre unos pies de plástico altamente resistentes, cada uno de las cuales lleva la placa de control de los motores sin escobillas (BLCB por sus siglas en inglés). Cada BLCB usa su propio microcontrolador de 8 bits (ATMEGA8L), los cuatro generadores de empuje son unidos por medio de cuatro tubos de fibra de carbono y un plástico reforzado en forma de cruz central.

La estructura de carbono soporta una estructura de polipropileno expandido (espuma flex) la cual lleva la batería de polímero litio (LiPo). El recipiente que

contiene la batería también es de polipropileno expandido, ya que filtra las vibraciones de los motores.

La parte electrónica consta de dos placas atornilladas a la cesta de plástico principal. La primera placa es la madre que se encuentra conectada a la placa de navegación. La placa madre incorpora un procesador de PArrot P6 (32 bits ARM9-core, que funciona a 468 MHz), un chip Wi-Fi, una cámara orientada verticalmente y un conector para la cámara frontal.

El procesador incorpora un sistema operativo en tiempo real basado en Linux y todo el software donde se ejecutan todos los cálculos, gestiona al mismo tiempo: conexión WIFI, datos, video, procesamiento de imágenes, sensores de adquisición, la estimación del estado y el control en lazo cerrado. La adquisición de datos y el hilo de control funcionan a una tasa de 200 Hz.

La placa de Navegación usa un microcontrolador PIC de 16 bits funcionando a 40 MHz, y sirve como una interfaz con los sensores. Estos sensores son un acelerómetro de 3 ejes, un giroscopio de 2 ejes, un giroscopio vertical de 1 eje, y un sensor ultrasónico en la parte inferior. El microcontrolador PIC controla el sensor ultrasónico que tiene una frecuencia de resonancia de 40 kHz el cual es utilizado para estimar la altitud.

Los acelerómetros y giroscopios constituyen una Unidad de Medición Inercial (IMU). Se usa un acelerómetro Bosh BMA150 de 3 ejes con un convertidor A/D de 10 bits, con un rango de +/- 2 g. El giroscopio de 2 ejes es un "Invensense IDG500", es un sensor analógico que es digitalizado por el conversor A/D de 12 bits del microcontrolador PIC con el cual se puede medir tasas de rotación de hasta 500 grados/s.

Las señales tanto de los acelerómetros y giroscopios deben filtrarse antes de ser procesadas, usando filtros digitales principalmente el de Kalman, las señales son utilizadas para realizar el diseño de algoritmos para estimar la altitud y la actitud del cuadricóptero (comportamiento rotacional), todo basado en el uso de matrices. La arquitectura de control se resume en la Figura 1.17.

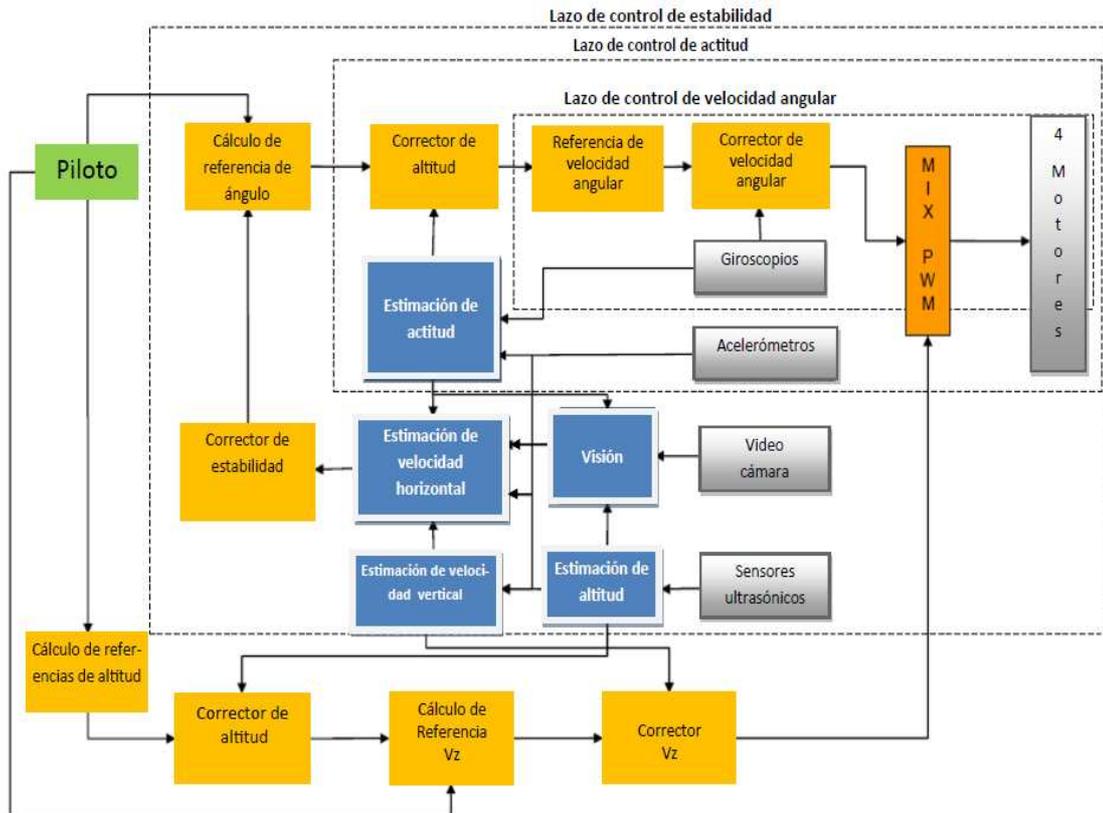


Figura 1.17 Descripción de la arquitectura de control, tomado de [9]



Figura 1.18 Parrot A.R. Drone

1.5 GENERALIDADES DE LOS PRINCIPALES DISPOSITIVOS ELECTRÓNICOS A EMPLEARSE

1.5.1 MICROCONTROLADORES

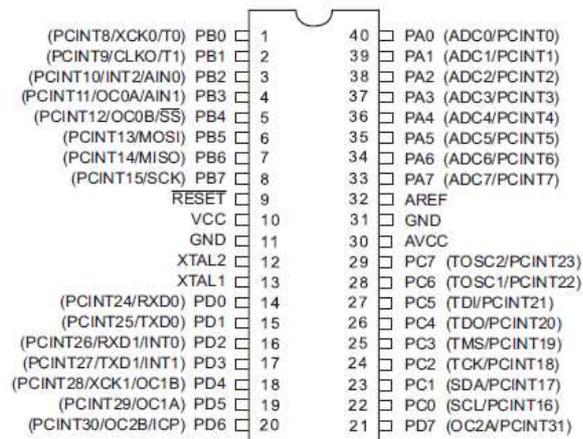
Por su extensa variedad, arquitectura moderna, la gran variedad de información disponible y sobre todo por su disponibilidad en el mercado, se emplearán microcontroladores de la familia AVR, del fabricante estadounidense Atmel. La arquitectura de los AVR fue concebida por dos estudiantes del Norwegian Institute of Technology, y posteriormente refinada y desarrollada en Atmel Noruega, la empresa subsidiaria de Atmel, fundada por los dos arquitectos del chip. El lazo de control que será implementado en el cuadricóptero tendrá un bloque de procesamiento, un actuador y una realimentación sensorial, cada bloque debería entregar una señal de control en tiempo real para que la respuesta del cuadricóptero al seguimiento sea instantáneo, sin embargo cada elemento que conforma cada uno de los bloques antes mencionados tienen un tiempo de respuesta que dificulta tener una respuesta instantánea. Para que cada bloque responda de la manera más rápida posible, se evitará centralizar todo el proceso del control en un solo microcontrolador, y se emplearán varios trabajando independientemente uno del otro, pero comunicados entre sí. Los microcontroladores que se emplearán se describen en los siguientes puntos.

1.5.1.1 Características del ATmega 164P

El ATmega 164P es un microcontrolador de 8 bits de tipo CMOS de bajo consumo basado en una arquitectura RISC, tiene la ejecución de las instrucciones de gran alcance en un solo ciclo de reloj, el dispositivo logra rendimientos que se acercan 1 MIPS por MHz, equilibrando el consumo de energía y velocidad de procesador. Las características principales se detallan en la Tabla 1.3.

Tabla 1.3 Características del Atmega324P

Memoria Flash (Kb)	16
Memoria EEPROM (b)	512
Memoria SRAM (bytes)	1024
Máxima Frecuencia de Oscilación (MHz)	20
Número de Interrupciones	32
Puertos USART	2
Salidas SPI	Si
Salidas I2C	Si
Canales AD	8
Timers 8 bits	2
Timers 16 bits	1
Rango de voltaje	1.8V – 5.5V

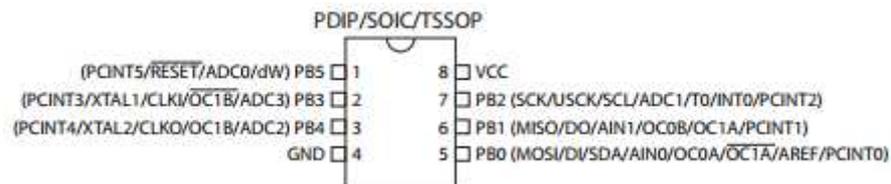
**Figura 1.19** Atmega324P, tomado de [26]

1.5.1.2 Características del ATtiny85

El ATtiny85 es un microcontrolador de 8 bits de tipo CMOS de bajo consumo basado en una arquitectura RISC, tiene la ejecución de las instrucciones de gran alcance en un solo ciclo de reloj, su principal diferencia con el ATmega164P es su tamaño, puesto que solo posee 8 pines y resulta versátil para generar señales que no requieran de demasiados pines para su salida. Las características principales se detallan en la Tabla 1.4.

Tabla 1.4 Características del ATtiny85

Memoria Flash (Kb)	8
Memoria RAM (b)	256
Memoria EEPROM (Kb)	512
Memoria SRAM (bytes)	512
Máxima Frecuencia de Oscilación (MHz)	20
Número de Interrupciones	6
Puertos USART	0
Salidas SPI	Si
Salidas I2C	No
Canales AD	2
Timers 8 bits	2
Timers 16 bits	0
Rango de voltaje	1.8V – 5.5V

**Figura 1.20** ATtiny85, tomado de [27]

1.5.1.3 Módulos XBee

Los módulos XBee son módulos de radio frecuencia que emplean el protocolo de comunicación Zigbee el mismo que está basado en el estándar de comunicaciones para redes inalámbricas IEEE_802.15.4, creado por Zigbee Alliance, una organización, teóricamente sin fines de lucro, de más de 200 grandes empresas (destacan Mitsubishi, Honeywell, Philip, Invensys, entre otras). Las comunicaciones Zigbee se realizan en la banda libre de 2.4GHz, a través de una única frecuencia, es decir, de un canal.

El uso del protocolo Zigbee va desde remplazar un cable por una comunicación serial inalámbrica, hasta el desarrollo de configuraciones punto a punto,

multipunto, peer-to-peer o redes complejas de sensores. El Cuadricóptero se comunica continuamente con el controlador inalámbricamente vía comunicación serial mediante los módulos XBee. Como se lo explicará en el Capítulo 2, el software de control del cuadricóptero envía las ordenes a la tarjeta madre mediante una red WiFi lo que limita su alcance a aproximadamente 50m en base a esta consideración , se seleccionó el módulo cuyas características constan en la Tabla 1.5.

Tabla 1.5 Características principales de los módulos XBee

Nombre	XBee Pro serie 1
Banda de trabajo	2.4 GHz
Tasa de envío de datos	254 Kbps
Tasa de Comunicación Serial	1200 bps a 250 kbps
Potencia	1mW
Alcance	30m en interiores 100m con línea de vista
Tipo de antena	Chip
I/O Digitales	8
Salidas PWM	2
Conversores A/D	Hasta 7 de 10 bits
Voltaje de alimentación	2.4V – 3.3V



Figura 1.21 XBee pro serie 1, tomado de [28]

1.5.1.4 Receptor infrarrojo para sistemas de control remoto

Como se explicará en el siguiente capítulo, el seguimiento al objeto móvil se lo realiza mediante un sistema de emisión y recepción infrarroja. El emisor es un led infrarrojo que envía un tren de pulsos a una cierta frecuencia, el receptor es el integrado TSOP 34438 que es un decodificador empleado en sistemas a control

remoto, este dispositivo recibe la señal infrarroja, la amplifica y filtra, dando como resultado una señal discreta, en respuesta a la frecuencia a la que está diseñado, que es fácil de procesar por cualquier microcontrolador y sobre todo libre de ruido.

El receptor está optimizado para suprimir cualquier pulso de energía proveniente principalmente de lámparas fluorescentes y su empaquetado actúa como un filtro infrarrojo. Las características principales de este dispositivo se resumen en la Tabla 1.6.

Tabla 1.6 Características del receptor infrarrojo

Serie	TSOP 34438
Frecuencia portadora	38 KHz
Tasa de envío de datos	254 Kbps
Máxima irradiancia	30 W/m ²
Mínima irradiancia	0.25 mW/m ²
Directividad	40 grados
Protección contra EMI	Si
Inmunidad contra la luz ambiente	Si
Voltaje de alimentación	2.5 V – 5.5 V

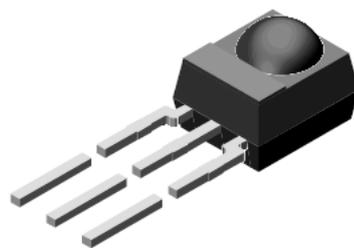


Figura 1.22 Receptor infrarrojo TSOP 34438, tomado de [29]

CAPÍTULO 2

DISEÑO Y ENSAMBLAJE DE LA PLATAFORMA MÓVIL

En este capítulo se describirá el funcionamiento del A.R. Drone así como también sus principales partes constitutivas, información que permitirá el desarrollo de los siguientes capítulos.

2.1 PARTES CONSTITUTIVAS DEL CUADRICÓPTERO PARROT A.R. DRONE

El AR.Drone es un UAV de la compañía francesa Parrot, es un producto a simple vista sencillo, pero que esconde mucha tecnología detrás. La estructura mecánica está compuesta por los 4 rotores en los 4 extremos del UAV, y en su parte central se encuentra la batería junto con el Hardware principal de funcionamiento.

Cada par de rotores opuestos giran en el mismo sentido. Un par de ellos está girando en sentido horario y el otro en sentido anti horario.



Figura 2.1. EL UAV Parrot A.R. Drone

2.1.1. CONFIGURACIONES DE DISEÑO INTERIOR Y EXTERIOR.

El A.R. Drone puede ser configurado para dos modalidades de vuelo, para vuelos en interiores y en exteriores. Para interiores posee una carcasa, de colores gris y plateado, tiene 4 anillos que rodean todas y cada una de las hélices (Figura 2.2). Es muy útil para los primeros vuelos y protegen al cuadricóptero de posibles choques, además de evitar hacer daño a otras personas en caso de un choque accidental.



Figura 2.2. Configuración para vuelo en interiores

La configuración para vuelos en exteriores es recomendable usarla en lugares despejados, donde no pueda chocar con personas u objetos, es más aerodinámica ya que no usa la carcasa de protección como se aprecia en la Figura 2.3



Figura 2.3. Configuración para vuelo en exteriores

2.1.2. MOTORES

El A.R. Drone cuenta con 4 motores de 15 W sin escobillas que funcionan a 35.000 revoluciones por minuto. Cada motor cuenta con leds de color rojo, naranja y verde, que indican el estado del motor y permiten verificar si hay algún problema, además de permitir diferenciar la parte trasera de la delantera mientras está en vuelo. El control es realizado por un microcontrolador incorporado en el UAV, el mismo que detecta automáticamente a los motores que están conectados y ajusta el control de los mismos, también detecta si todos los motores están

girando o están detenidos, en caso de que una hélice que esté girando encuentre algún obstáculo, el A.R. Drone detecta si alguna de las hélices se bloquea y en tal caso todos los motores se detienen de inmediato. Este sistema de protección impide repetidos choques y el consecuente daño de los motores.

El cuadricóptero también cuenta con 4 hélices de alta eficacia y una estructura de tubos, ambos elaborados a base de fibra de carbono.

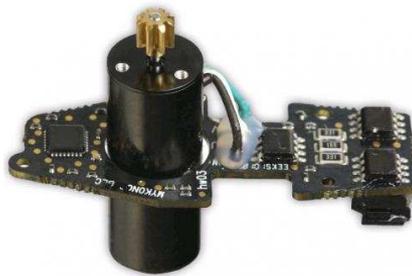


Figura 2.4. Motor del AR Drone

2.1.3. BATERÍAS DEL UAV

El A.R. Drone utiliza una batería recargable LiPo (Lithium Polymer) de 3 celdas de 11,1 V y 1000 mAh para alimentarse. Durante el vuelo la batería disminuye la tensión de carga completa que es 12.5 V a 9 V, el A.R. Drone convierte esta tensión en un porcentaje de duración de la batería (100% si la batería está llena, 0% si la batería está baja) [16].

Cuando el UAV detecta un bajo voltaje de la batería, primero envía una advertencia para el usuario, de forma automática. Si la tensión alcanza un nivel crítico el sistema se apagará con el fin de evitar cualquier comportamiento inesperado.

En conjunto con el cuadricóptero se tiene un cargador LiPo balanceado para recargar la batería y mantenerla siempre en perfectas condiciones, alargando su vida útil. Incluye distintos adaptadores para poderlo usar en varios países que no usan el mismo tipo de enchufe. La desventaja es que la batería dura tan sólo 12 minutos durante el vuelo, y se necesitan 90 minutos para recargarla.



Figura 2.5. Batería y cargador del AR Drone

2.1.4. CÁMARA FRONTAL

Esta pequeña cámara se encargará de grabar y transmitir todo lo que ve. Cuenta con una lente de gran angular a 93°, sensor CMOS y resolución VGA (640 X 480), con una velocidad de video de 15 fps (tramas por segundo). Esta cámara además de permitir ver en tiempo real lo que está viendo el AR.Drone, también permitirá detectar otros objetos, como por ejemplo, otros AR.Drone [16].



Figura 2.6. Cámara frontal del AR Drone

2.1.5. SISTEMAS DE AYUDA A LA NAVEGACIÓN

Para la navegación el A.R. Drone cuenta con una cámara vertical de alta velocidad (Figura 2.7) a 60 fps con un ángulo de 64° . Esta cámara permite que el sistema informático lo estabilice incluso con viento leve (velocidad del viento máxima de 15 km/h). También se tiene la posibilidad de conectar con esta cámara desde la aplicación de control, una opción realmente útil para aterrizarlo correctamente [16].

Cuenta también con un emisor y un receptor de ultrasonidos a una frecuencia de 40 kHz y con un alcance de hasta 6 metros. Estos sensores permitirán estabilizar verticalmente y le permitirá saber al sistema informático a qué altura se encuentra.

Se cuenta también con los sistemas de orientación inercial (Figura 2.8) con micro sistema electromagnético, que consiste en un acelerómetro de 3 ejes, un giroscopio de 2 ejes y de un segundo giroscopio de precisión de viraje en 1 eje.

Se tiene sistemas de seguridad como aterrizaje automático en caso de que esté a punto de agotarse la batería, descenso automático a 6 metros en caso de pérdida de la señal, estabilización y bloqueo de las hélices en caso de contacto o choque.

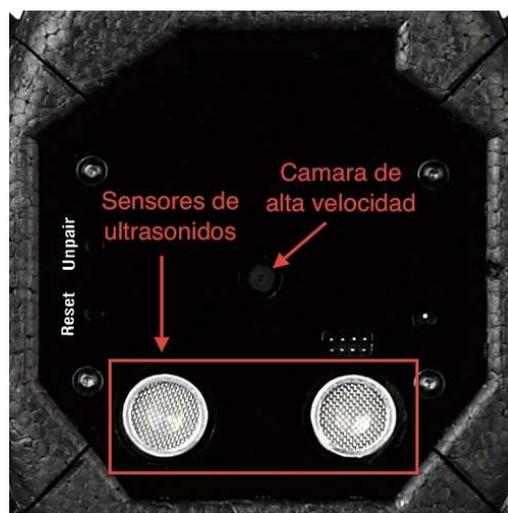


Figura 2.7 cámara vertical y sensores ultrasónicos del AR Drone

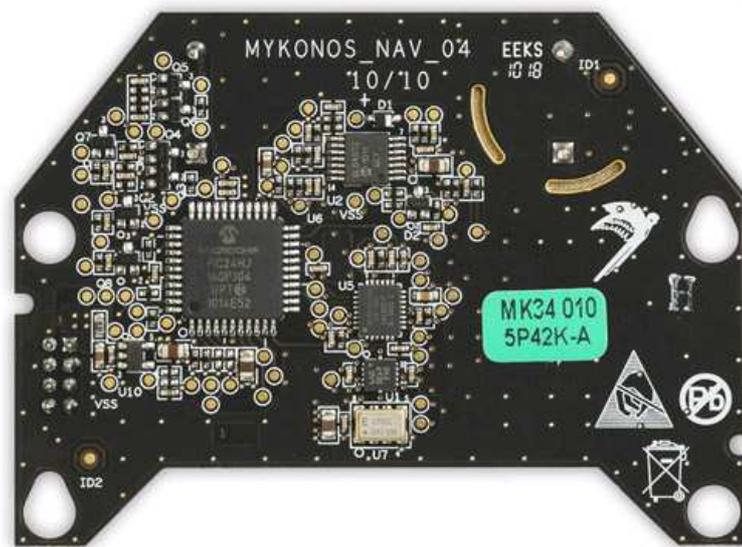


Figura 2.8 Sistema de orientación inercial

2.1.6. SISTEMA INFORMÁTICO INTEGRADO

El AR.Drone, al contrario de la mayoría de helicópteros y aviones teledirigidos convencionales, tiene un sistema informático complejo, que supera en potencia a algunos teléfonos móviles actuales.

Cuenta con un microprocesador ARM 9 a 468 MHz y dispone de 128 MB de memoria RAM DDR a 200 MHz, un chip WiFi b/g el cual usa para crear una red wifi al cual se tendrá acceso para controlar el AR.Drone.

A nivel de software, usa sistema operativo Linux, y el software es totalmente actualizable y permite ser personalizado. Mediante la interfaz USB, se puede cargar un software propio [16].



Figura 2.9 Sistema informático y procesador principal

2.2 DISEÑO DE LA PLATAFORMA MÓVIL

La plataforma móvil la cual va a ser seguida por el cuadricóptero estará constituida principalmente por un vehículo terrestre manejado a control remoto, dentro del cual se montó un dispositivo que envíe constantemente una señal infrarroja acondicionada.

Desde la plataforma móvil se envía una señal mediante un arreglo de emisores infrarrojos. Un arreglo de leds IR se encarga de transmitir una señal periódica con una luz que tiene longitud de onda que se encuentra en el espectro infrarrojo, existe una gran variedad de leds IR, fáciles de conseguir en el mercado que dependiendo de su acondicionamiento (de la intensidad de la señal) pueden tener un alcance de 10 m a 15 m.

La decodificación de la señal infrarroja resulta compleja si se utiliza un fotodiodo o un fototransistor, debido a que la luz del medio ambiente influye sustancialmente

sobre estos dispositivos electrónicos, esto introduce ruido al sistema lo que hace que el diseño del circuito de acondicionamiento sea complejo, con la posibilidad de que no responda adecuadamente a una distancia considerable. Ventajosamente en el mercado existen decodificadores que emplean la misma tecnología que los que usan las televisiones, equipos de sonido en general todo aparato electrónico que funcione a control remoto, estos receptores ofrecen una respuesta libre de ruido y totalmente confiable como se explicará en el punto 2.3.

2.2.1 DISEÑO DEL CIRCUITO EMISOR

El circuito emisor consta principalmente de un arreglo de leds infrarrojos SID313BP (Figura 2.10) cuyas características se describen en la Tabla 2.1.

Tabla 2.1 Características del led ir SID313BP

Característica	Típico	Máximo
Voltaje en polarización directa VF	1.25 V	1.4 V
Voltaje en polarización inversa VR	5V	
Corriente en polarización directa IF	50 mA	100 mA
Longitud de onda de la emisión λ_p	940 nm	
Potencia de disipación	-	150 mW



Figura 2.10 Led infrarrojo

Como se indica en el punto 2.3, la frecuencia óptima para que funcione el receptor infrarrojo es de 38 KHz, para tener una señal lo más precisa posible se emplea un microcontrolador para generarla.

Mediante una resistencia se limita la corriente a través del led IR, la Figura 2.11 indica la forma de onda, que será la misma que la del voltaje por no existir elementos capacitivos o inductivos en el circuito que genera la señal de 38KHz (Figura 2.12).

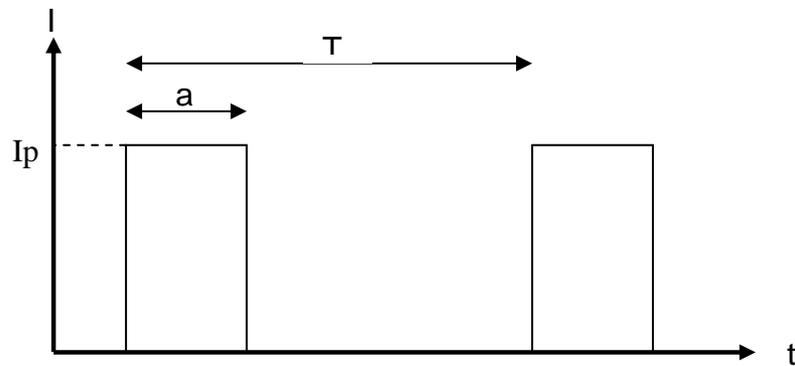


Figura 2.11 Corriente a través del led IR

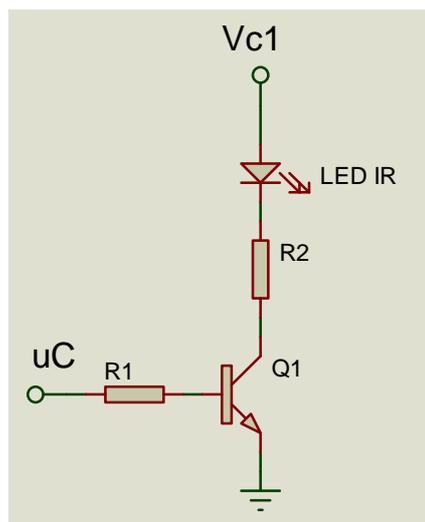


Figura 2.12 Acondicionamiento para la emisión de luz infrarroja

El alcance de la señal infrarroja va a depender de la potencia de emisión, es decir principalmente de la corriente que circula a través del led IR y del voltaje aplicado entre $Vc1$ y tierra.

La corriente RMS que va a circular se calcula mediante la siguiente expresión:

$$I_{RMS}^2 = \frac{1}{T} \int_0^a I^2 dt$$

Integrando la expresión y sabiendo que el ancho de pulso “a” se puede expresar en función de la relación de trabajo “ δ ” como: $a = \delta * T$, donde T es el periodo de la señal, se tiene:

$$I_{RMS} = I * \sqrt{\delta}$$

Aumentando la relación de trabajo en la señal de 38 KHz, se puede aumentar la corriente IRMS que circula a través del led, por lo tanto se incrementa la potencia de emisión y con esto su alcance. Para asegurar una corriente IRMS de 50 mA, que sería la máxima que puede soportar, con una relación de trabajo de $\frac{1}{4}$, la corriente pico es:

$$I = \frac{I_{RMS}}{\sqrt{\delta}} = \frac{50 \text{ mA}}{\sqrt{1/4}} = 100 \text{ mA}$$

Para asegurar la corriente pico de 100 mA, el transistor Q1 se colocará uno de la serie 2N3904 que soporta una corriente de colector de 200 mA. Para mejorar el rango de detección del faro infrarrojo, se puede aumentar el número de diodos emisores, la mejor opción es disponerlos en serie porque se limita de este modo la potencia que va a disipar la resistencia limitadora de corriente. El faro infrarrojo consta de 20 diodos infrarrojos dispuestos en serie de manera circular, en cada diodo el voltaje en polarización directa (V_f) es de aproximadamente 1.2 V según la hoja de datos, lo que significa que se necesitará de al menos 24 V de voltaje de la fuente.

Para un voltaje de 30 V en la fuente el voltaje colector emisor en el transistor será:

$$V_{CEO} = 30 \text{ V} - 24 \text{ V} = 6 \text{ V} \text{ en saturación}$$

$V_{CEO} = 0.2 \text{ V}$ en corte según la hoja de datos, como el transistor va a trabajar con una relación de trabajo del 25%, el valor rms del voltaje en el transistor para la señal de 38 KHz será:

$$V_{ceo}rms^2 = \frac{1}{T} * \left(\int_0^{\frac{T}{4}} 6^2 dt + \int_{\frac{T}{4}}^T 0.2^2 dt \right)$$

$$V_{ceo}rms = \sqrt{\frac{1}{T} * \left(6^2 * \frac{T}{4} + 0.2^2 * \frac{3T}{4} \right)} = 3.005 V$$

Con el voltaje de la fuente, el del grupo de diodos emisores y el del transistor se puede calcular el voltaje en la resistencia limitadora el mismo que es igual a:

$$V_{R2} = 30 - 24 - 3.005 = 2.995 V rms$$

Por lo tanto el valor de la resistencia limitadora será:

$$R2 = \frac{V_{R2}}{I} = \frac{2.995 V}{50 mA} = 59.9 \Omega$$

Normalizando la resistencia estándar que se utiliza en R2 es de 68 Ω .

La resistencia R1 se calcula en base a la corriente de base del transistor; de la hoja de datos del transistor 2N3904 la corriente de base cuando el dispositivo está en saturación, y se tiene un voltaje base – emisor de 0.65 V, es de 1 mA. Cuando se aplique 5 V de la señal de control a la base, la resistencia R1 que se deberá poner será de:

$$R1 = \frac{V_B}{I_B} = \frac{5 V}{1 mA} = 5000 \Omega$$

La resistencia estándar a utilizar es una de 5.6 k Ω .

Para el circuito final de la Figura 2.13 se emplea el microcontrolador ATtiny85 que posee 8 pines, ya que se requiere 1 pin de salida para generar la señal. Mediante un regulador LM7805 se alimenta al microcontrolador y el conjunto de leds

infrarrojos serán alimentados con 30 V proporcionados por 4 baterías de 9 V en serie.

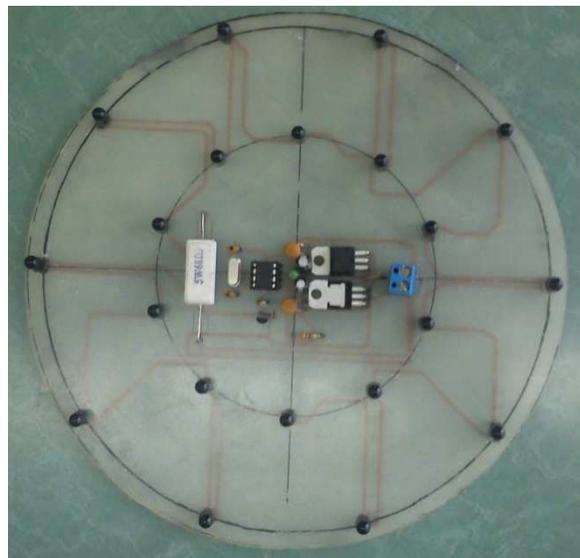
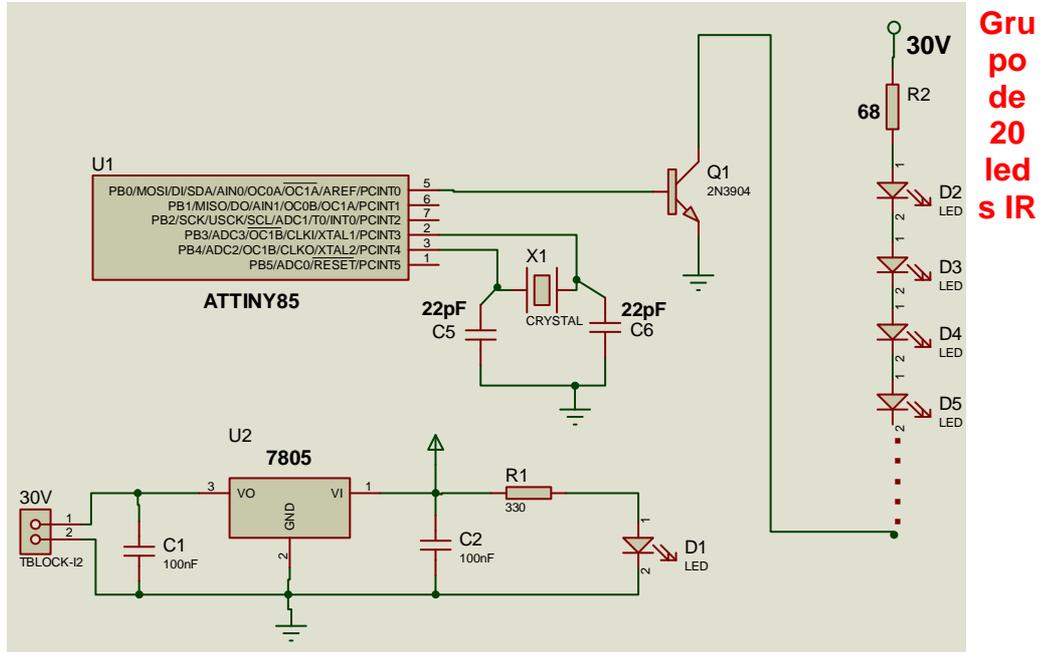


Figura 2.13 Circuito emisor de 38 KHz

2.3 DISEÑO DE LA PLATAFORMA DE SEGUIMIENTO

La plataforma móvil montada en un carro a control remoto envía constantemente un tren de pulsos a 38KHz mediante un arreglo de leds infrarrojos. La señal infrarroja luego de ser procesada indica donde se encuentra el móvil y mediante un arreglo de receptores infrarrojos se detecta la presencia del mismo para que el cuadricóptero pueda seguir al objeto móvil.

2.3.1 DISEÑO DEL CIRCUITO RECEPTOR

La señal infrarroja será detectada con el receptor TSOP32438, que en su hoja de datos se especifica un circuito de acondicionamiento que es el mas óptimo, como se ve en la Figura 2.14.

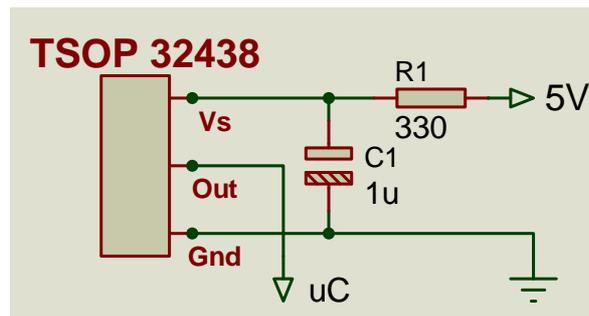


Figura 2.14 Circuito de acondicionamiento para el receptor infrarrojo, tomado de [22]

El receptor ofrece una salida discreta en respuesta a la señal emitida, por lo que es necesario realizar un arreglo de receptores para determinar hacia donde se mueve el objeto, consecuentemente hacia donde ha de moverse el cuadricóptero. Como se indicó en el punto 2.2, el emisor no tiene una salida direccional, sino que forma un cono que se ensancha conforme aumenta la altura, teniendo en cuenta este aspecto, se decidió colocar 9 receptores distribuidos como indica la Figura 2.15 de tal modo que se pueda cubrir la mayor cantidad de direcciones posibles a pesar del cono que forma la emisión del led infrarrojo.

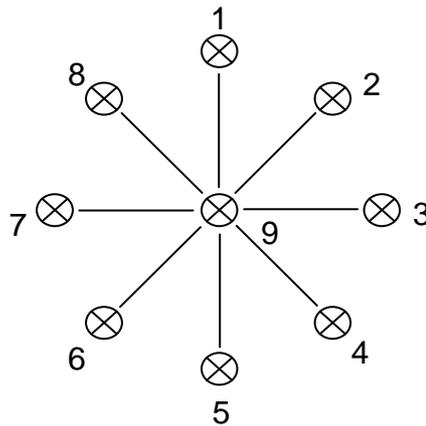


Figura 2.15 Disposición de los receptores en el cuadricóptero

Para eliminar cualquier tipo de ruido que pueda enviar el receptor hacia el microcontrolador, se coloca un filtro pasa bajos debido a que en el umbral de detección del cono infrarrojo se produce una señal oscilatoria de frecuencia aleatoria que fue medida aproximadamente entre 70 Hz y 500 Hz.

La frecuencia de corte necesaria para asegurar que el microcontrolador reciba una señal confiable de detección se fijó en 50 Hz, entonces:

$$f_c = \frac{1}{2\pi RC}$$

Si $R=330 \Omega$ resulta

$$C = \frac{1}{2\pi * 330\Omega * 50Hz} = 9.64e^{-6}F$$

Por lo tanto el capacitor a seleccionar deberá ser uno de 10 μF .

Las 9 señales de los receptores se procesan mediante un microcontrolador ATmega324P, haciendo uso de una de sus características como es la de hacer que cualquiera de sus pines de entrada o salida puedan actuar como interrupciones. Mediante comunicación serial se enviarán los resultados hacia un ATmega328P, evitando así centralizar el funcionamiento en un solo

microcontrolador, el mismo que utiliza esa información para enviarla vía comunicación serial inalámbrica mediante un módulo XBee.

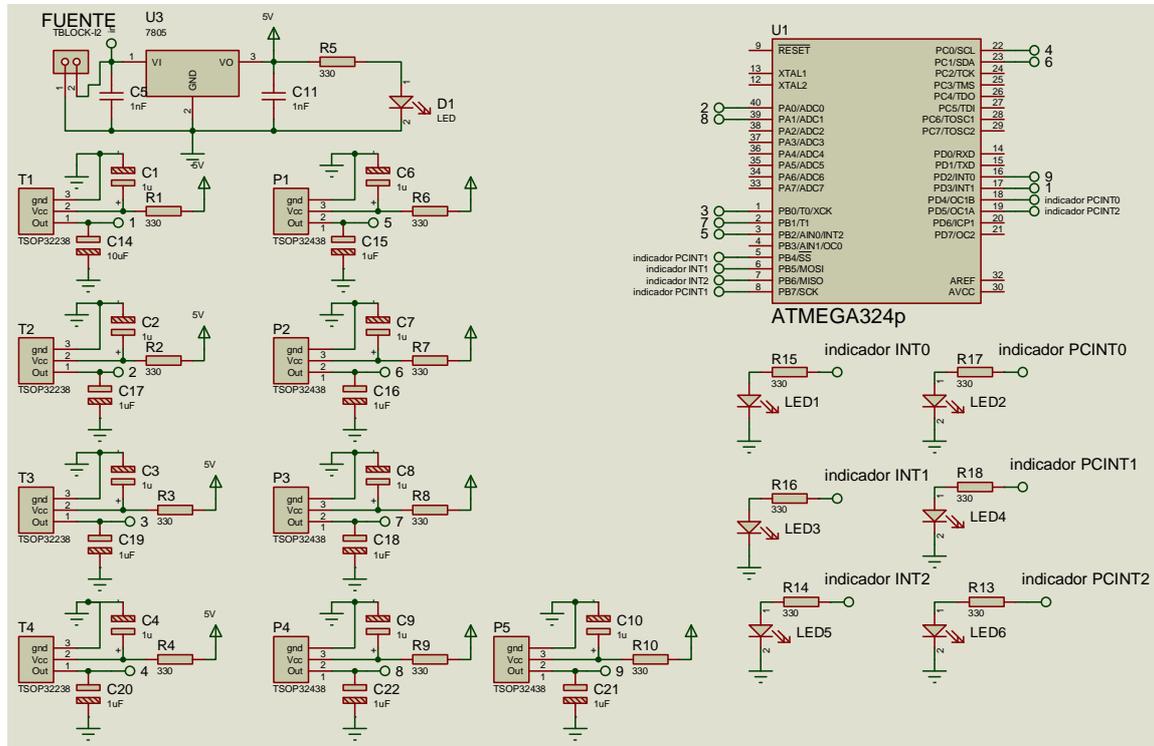


Figura 2.16 Circuito receptor, diagrama y placa armada

El buen funcionamiento del receptor TSOP32438, depende principalmente de que tan nítida y potente sea la señal infrarroja emitida. Como se explicó en el punto anterior el led infrarrojo forma un cono de aproximadamente 40 grados según su diagrama de radiación (Figura 2.17), donde se puede apreciar también que la máxima intensidad se tiene a 0 grados y se reduce en sus alrededores, teniendo

así una pequeña zona donde la señal es clara y perceptible para el receptor, consecuentemente un problema de directividad.

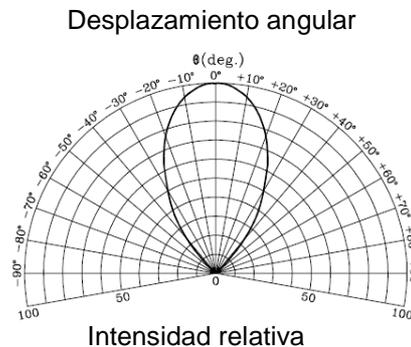


Figura 2.17 Diagrama de emisión del led IR, tomado de [22]

2.3.2 ENSAMBLAJE DE LA PLATAFORMA DE SEGUIMIENTO

El circuito conformado por el arreglo de leds infrarrojos debe ir montado sobre un móvil terrestre para lo cual se seleccionó un carro de juguete a control remoto el cual tiene las características de desplazamiento hacia adelante, atrás, derecha e izquierda con tracción en las ruedas posteriores y la dirección en las delanteras.

La velocidad del móvil debe ser baja para asegurar el seguimiento del cuadricóptero para lo cual la alimentación del vehículo será mediante una fuente de 5 V además de una extensión de cable para la polarización de 30 V necesarios para la placa de emisores infrarrojos.



Figura 2.18 Móvil a seguir por el cuadricóptero

2.4 PLATAFORMA DE CONTROL.

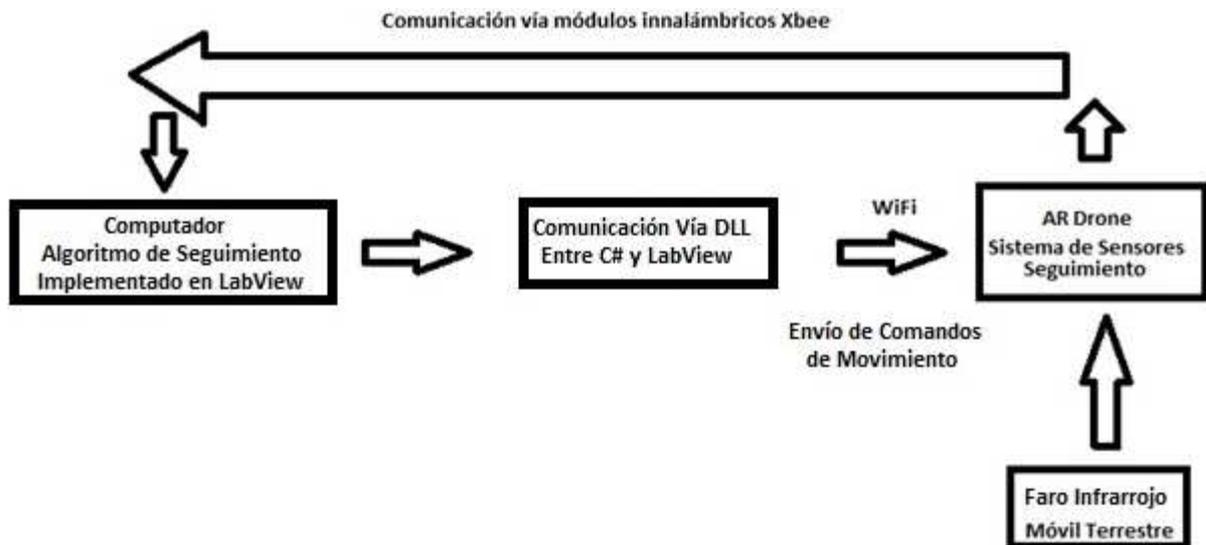


Figura 2.19 Esquema de Control

Se ha basado el control en el diseño de una HMI mediante la cual se podrá realizar el seguimiento automáticamente, con la posibilidad de corregir comportamientos anómalos de vuelo del AR Drone mediante un teclado, es decir, se podrá pilotear el AR Drone manualmente.

Los datos enviados por el Xbee del cuadricóptero se reciben mediante otro Xbee comunicado vía RS-232 con LabVIEW donde mediante un algoritmo que será explicado en el siguiente capítulo se obtienen los comandos de movimiento a ser enviados al AR Drone para lo que se requiere una comunicación WiFi la cual se encuentra implementada en un programa escrito en C# desarrollado por Thomas Endres, Steve Hopley y Julien Vinel, del cual se dispuso el código fuente para modificar e implementar aspectos puntuales que permiten recibir los comandos de movimiento desde LabVIEW mediante una DLL.

2.4.1 INTERFAZ EN C#.

El programa realizado en C# está basado en el SDK distribuido por Parrot y desarrollado por Thomas Endres, Steve Hopley y Julien Vinel, el cual permite el uso de su código fuente para modificar cualquier aspecto del mismo, pero ya que las modificaciones que son necesarias para la realización del proyecto son puntuales, se ha reutilizado todo el código y agregado ciertas características al programa para convertirlo en un instrumento viable para la consecución de los objetivos planteados, las modificaciones que se realizó en la interfaz se detallará en el Capítulo 3.



Figura 2.20 AR.DRONE Control. NET desarrollado en C#

2.4.2 COMUNICACIÓN ENTRE EL CIRCUITO DE RECEPCIÓN INFRARROJO Y LA PC.

Para realizar la comunicación entre el circuito receptor y la computadora se usan dos módulos inalámbricos XBee de radio frecuencia que trabajan en la banda de 2.4 GHz con protocolo de comunicación 802.15.4 fabricados por DIGI. Se usa la Serie 1 con tipo de antena chip ya que no se necesita hacer una red, satisface las necesidades, requiere niveles mínimos de potencia y proporciona una entrega fiable de datos entre dispositivos, además de que su pequeño tamaño ayuda a su integración en una PCB, a éstos se los configuró a 57600 baudios en una comunicación serial asíncrona. La siguiente figura muestra la estructura de un carácter que se trasmite en forma serial asíncrona.

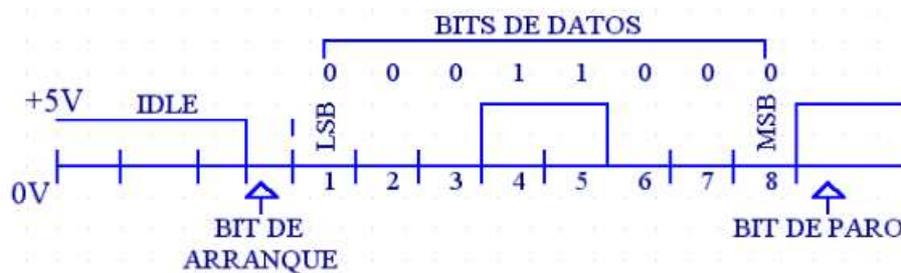


Figura 2.21 Trama de datos comunicación serial asíncrona, tomado de [15]

Normalmente cuando no se realiza ninguna transferencia de datos, la línea del transmisor se encuentra en estado de (idle) esto quiere decir en estado alto.

Para iniciar la transmisión de datos, el transmisor coloca esta línea en bajo durante determinado tiempo, lo cual se le conoce como bit de arranque (start bit) y a continuación empieza a transmitir con un intervalo de tiempo los bits correspondientes al dato, empezando siempre por el BIT menos significativo (LSB), y terminando con el BIT más significativo (MSB). Si el receptor no está sincronizado con el transmisor, este desconoce cuándo se van a recibir los datos, por lo tanto el transmisor y el receptor deberán tener los mismos parámetros de velocidad, paridad, número de bits del dato transmitido y de BIT de parada.

La configuración de los módulos XBee se realiza mediante el programa X-CTU, donde se indica la velocidad de comunicación, el bit de inicio y el de parada, entre otros parámetros.

2.4.3 DESCRIPCIÓN DEL MODO DE VUELO MANUAL.

Para el vuelo manual el HMI de C# se encarga de tomar los comandos de movimiento del AR Drone directamente del teclado enviándole diferentes comandos de vuelo permitiéndole despegar, aterrizar, maniobrar y visualizar el video de las cámaras del AR Drone.

Un aspecto importante es que cuando se encuentra en el modo de seguimiento automático se puede rápidamente cambiar al modo manual para usar el teclado y corregir comportamientos anómalos del AR Drone durante el vuelo.

2.4.4 DESCRIPCIÓN DEL MODO DE SEGUIMIENTO AUTOMÁTICO

Para realizar el seguimiento autónomo, en el HMI de C# se selecciona el modo automático el cual llama a una HMI de LabVIEW mediante una DLL, la cual es la encargada del algoritmo de control valiéndose de la información proporcionada por el conjunto de sensores infrarrojos montados en el AR Drone para el seguimiento del móvil en tierra.

CAPÍTULO 3

DESARROLLO DEL SOFTWARE

En este capítulo se analiza y explica el envío de datos hacia el computador, la decodificación de los mismos, el algoritmo de detección, el control de seguimiento y el uso de la interface de control para el envío de los comandos de vuelo.

Para que el seguimiento del faro infrarrojo sea posible hace falta recurrir primero a la programación de un microcontrolador ATMEGA 164P para que obtenga la información de los sensores durante el vuelo y los envíe hacia la computadora en tierra mediante el módulo XBee, luego se requiere de la implementación de un programa en LabVIEW 2012 para que sea el que se encargue de la decodificación de los sensores y de la implementación del algoritmo de detección y control, posteriormente se realiza la modificación y manipulación bajo LICENCIA PÚBLICA GENERAL GNU del programa AR.DRONE.CONTROL.NET escrito en C#, lo que se hará es que se comuniquen estos dos programas dinámicamente a través de una DLL (Dinamyc Library Link), para que los datos de control que resultan del VI de LabVIEW sean enviados al AR. Drone mediante el programa escrito en C#.

3.1 DETECCIÓN ENVÍO Y DECODIFICACIÓN DE DATOS

Para poder detectar al móvil terrestre y posteriormente realizar su seguimiento se montó un sistema emisor infrarrojo que formará un faro, cuya presencia y movimiento será detectado por los sensores infrarrojos TSOP32438 colocados en el AR Drone. La detección del faro infrarrojo, el mismo que consta de 20 leds infrarrojos, se realiza mediante un arreglo de 8 sensores infrarrojos dispuestos como lo indica la Figura 2.15, esta disposición es clave para posteriormente procesar las detecciones individuales y de esta forma obtener la posición relativa del cuadricóptero con respecto al faro infrarrojo mediante el tratamiento de las señales de activación de los sensores IR.

3.1.1 ENVÍO DE DATOS

Para poder enviar los datos desde el circuito de detección, primero se los debe recibir y empaquetar individualmente, el circuito receptor obtiene las 8 señales confiables de presencia de la señal infrarroja acondicionada, gracias al filtro pasabajos implementado en el circuito receptor explicado en el literal 2.3.1, en este caso lo que se obtiene en cada uno de los pines del microcontrolador ATMEGA 164P es un estado bajo (0 lógico) cuando un sensor detecta la presencia de la señal infrarroja de 38 KHz y un estado alto (1 lógico) cuando el sensor deja de detectar la señal infrarroja, el microcontrolador se encarga de empaquetar estos estados lógicos en un solo byte para poder enviarlo vía comunicación serial al XBee y posteriormente ser enviados inalámbricamente hacia el computador, donde serán decodificados y procesados.

Una vez que los datos de detección son empaquetados en un byte, éste es enviado vía comunicación serial a una velocidad de 57600 baudios al módulo XBee, el que se encarga de enviarlo inalámbricamente, el módulo receptor el cual se comunica con el HMI de control a través de una DLL construida en LabView, la cual es llamada desde C#.

Cuando el byte ha sido recibido exitosamente por la PC se realiza la decodificación del mismo para tener nuevamente los estados individuales de activación o no activación de los sensores infrarrojos.

La forma en que el microcontrolador empaqueta los datos de detección se presenta en el diagrama de flujo de la Figura 3.1:

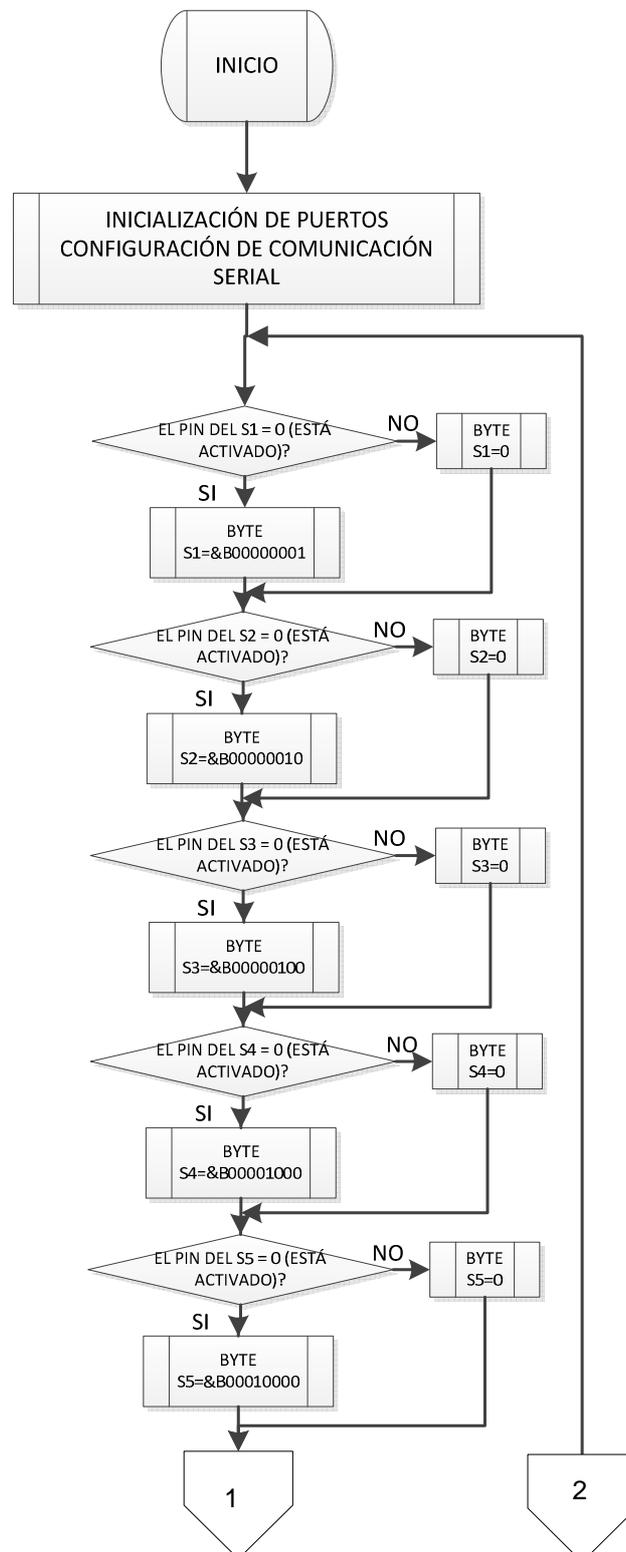


Figura 3.1 Diagrama de flujo del empaquetamiento de datos en el microcontrolador ATmega324P

PARTE A.

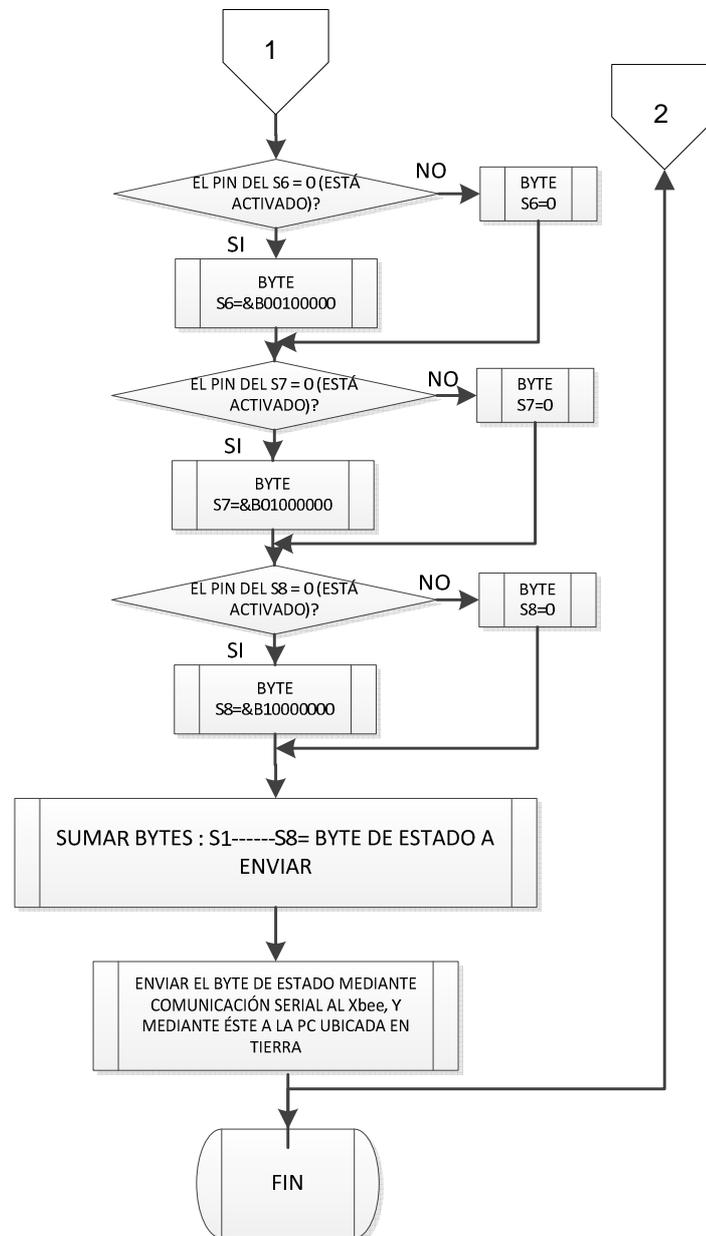


Figura 3.2 Diagrama de flujo del empaquetamiento de datos en el microcontrolador ATmega324P
PARTE B.

3.1.2 DECODIFICACIÓN DE LOS DATOS EN LABVIEW

La decodificación de los datos es sumamente importante ya que permite conocer individualmente qué sensor es el que está siendo activado por la señal infrarroja y de esta forma poder implementar el algoritmo necesario para tener la posición

relativa del cuadricóptero con respecto al móvil. La decodificación se explica mediante el diagrama de flujo de la Figura 3.3:

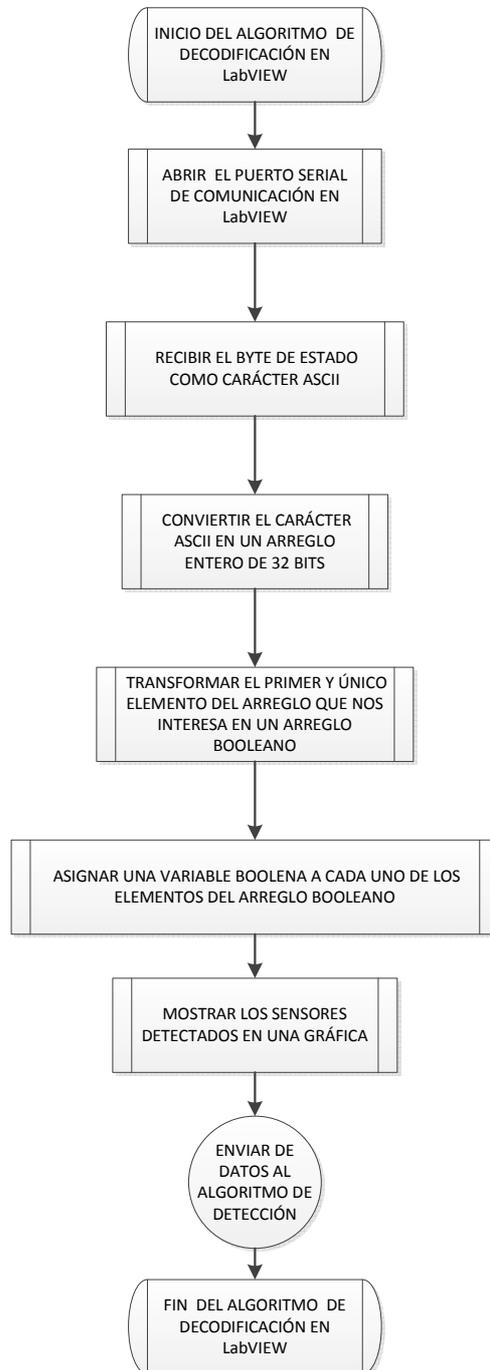


Figura 3.3 Diagrama de flujo del algoritmo de decodificación implementado en LabView

Una vez que se han decodificado los datos provenientes del circuito receptor montado en el AR Drone se puede empezar a tratarlos y procesarlos con el

programa implementado en LabView 2012. Para facilitar la identificación de los sensores que se encuentran sobre del faro infrarrojo y los que no, se presenta un esquema de la activación individual de los sensores numerados del 1 al 8.

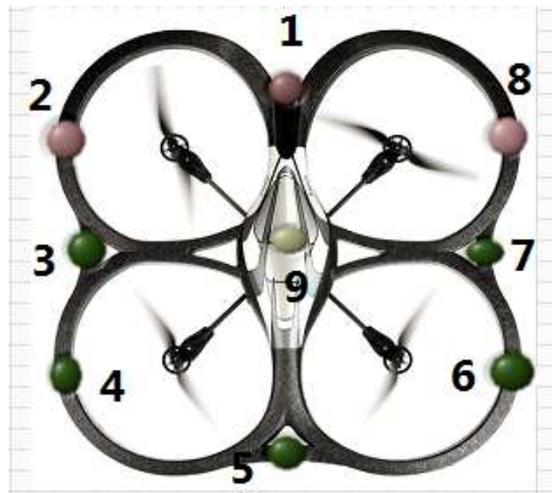


Figura 3.4 Esquema de la disposición de los sensores montados en el AR Drone

3.1.3 ALGORITMO DE DETECCIÓN

El algoritmo de detección que se implementó en el HMI de LabVIEW tiene por objetivo dar como resultado una posición relativa entre el AR Drone y el móvil terrestre a seguir en el plano XY y así poder tomar una acción de control para el seguimiento.

Como se explicó en el punto 2.3.1 y en la Figura 2.15 los sensores infrarrojos están dispuestos en una configuración de una circunferencia cuyo centro coincide con el centro del AR Drone y cuyo radio es $R= 13.5$ cm.

Para el análisis, se realiza el procesamiento de las señales booleanas de activación de los 8 sensores por separado para el eje X y para el eje Y. En el eje X (roll) si uno de los sensores está activado este adquiere cierto valor, dicho valor es una ponderación de su posición relativa respecto al centro del AR Drone, en cambio si el sensor no está activado adquiere el valor de 0.

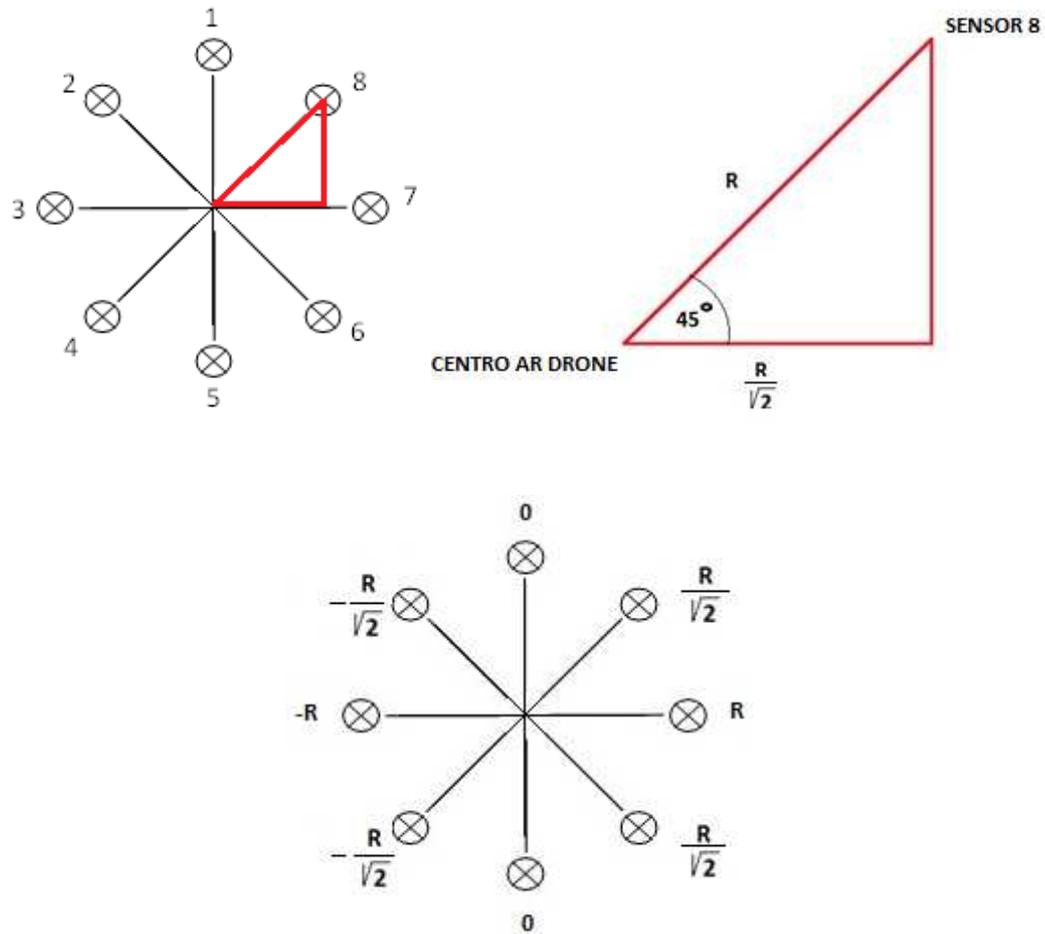


Figura 3.5 Posición de los sensores en vista superior y modo de obtención de la ponderación del sensor 8 en el eje X cuando son activados.

Tabla 3.1 Cuadro resumen de los valores que obtienen los sensores en el eje X

SENSOR	1	2	3	4	5	6	7	8
PONDERACIÓN EJE X	0	$-R/\sqrt{2}$	$-R$	$-R/\sqrt{2}$	0	$R/\sqrt{2}$	R	$R/\sqrt{2}$

Como se puede observar todos y cada uno de los valores que obtienen los sensores son ponderaciones que vienen de la relación geométrica relativa a la posición del sensor con respecto al centro del AR Drone y a cada uno de los ejes,

ésta es la forma que se ha escogido para obtener la información necesaria y realizar el seguimiento. El tratamiento en el eje Y es parecido y sucede algo similar como se observa en la Figura 3.6:

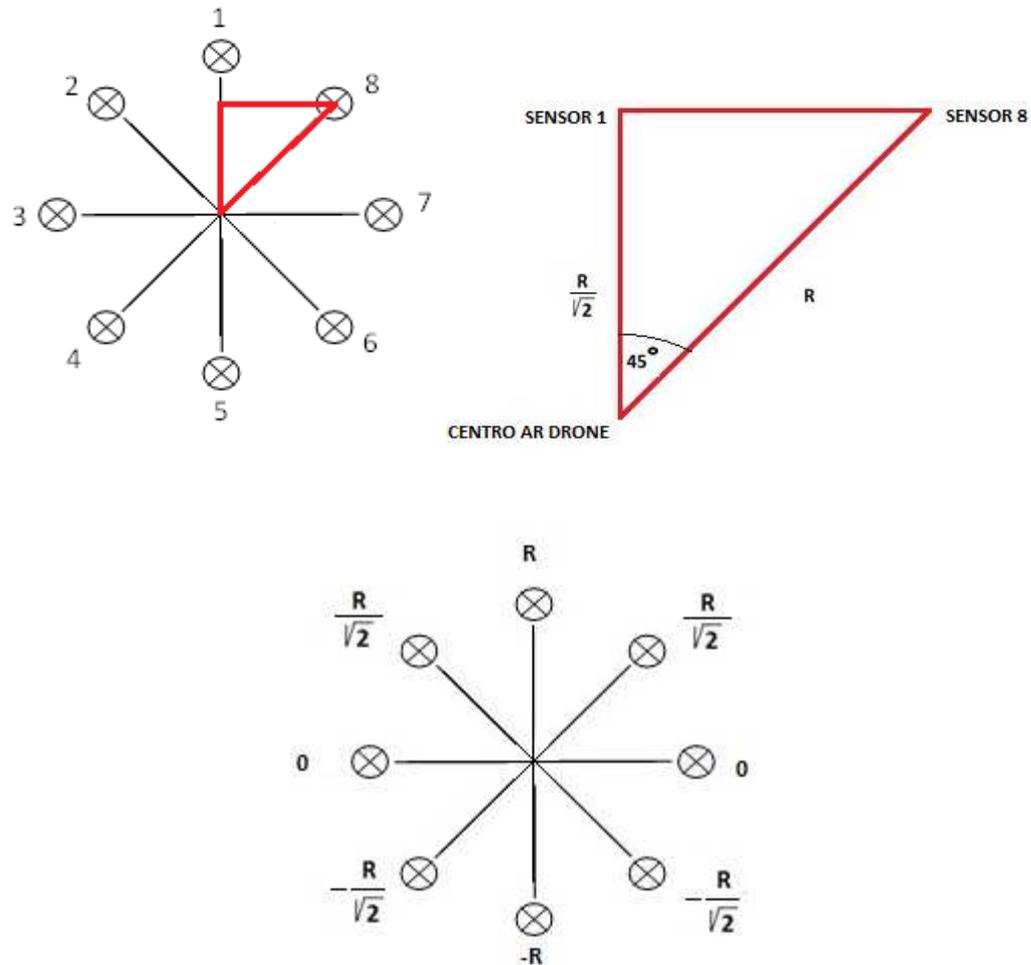


Figura 3.6 Posición de los sensores en vista superior y modo de obtención de la ponderación del sensor 8 en el eje Y cuando son activados.

Tabla 3.1 Cuadro resumen de los valores que obtienen los sensores en el eje Y

SENSOR	1	2	3	4	5	6	7	8
PONDERACIÓN EJE Y	R	$R/\sqrt{2}$	0	$-R/\sqrt{2}$	-R	$-R/\sqrt{2}$	0	$R/\sqrt{2}$

La ponderación de los 8 sensores cuando están activados tanto en Y (pitch) como en X (roll) se presentan en la siguiente figura en el formato $[X, Y]$, con $R= 13.5$ cm.

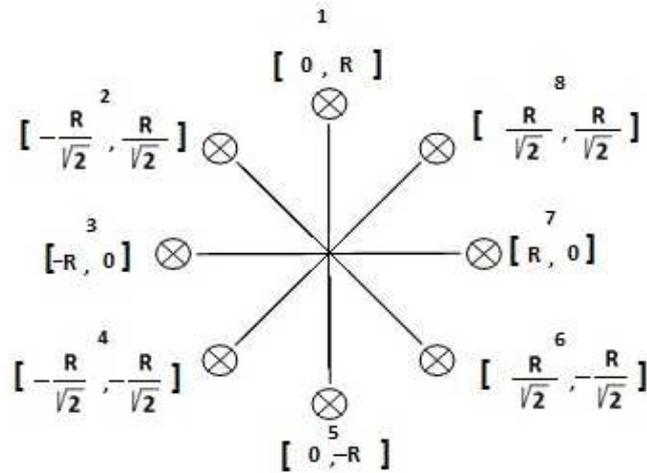


Figura 3.7 Ponderación de los sensores cuando están activados $[X, Y]$

Siendo R el valor de distancia del centro del AR Drone a cada uno de sus 8 sensores y en este caso particular $R=13.5$ cm, se puede notar que el valor que toma el sensor 1 es $[0, R]$ dado que en el eje X no produce ninguna variación relativa, es decir su valor en el eje de X es de 0 mientras que en el eje Y su valor es R ya que su variación de distancia relativa respecto al eje Y es R .

Una vez que cada uno de los 8 sensores obtiene su valor cuando se activan tanto en X como en Y , lo que hace el algoritmo es sumar dichas ponderaciones por separado, una suma en X y otra en Y , luego dichas sumas son divididas para el número de sensores activados, con lo que se deduce la expresión (3.1) que representa la distancia que debe recorrer el cuadricóptero de tal modo que todos los sensores detecten el faro infrarrojo, es decir cuánto debería moverse para posicionarse sobre el móvil.

$$D = \frac{\sum_{i=1}^n [X, Y]_n}{n} \quad (3.1)$$

Siendo n el número de sensores activados, el diagrama de flujo del programa que realiza el algoritmo de detección se presenta a continuación.

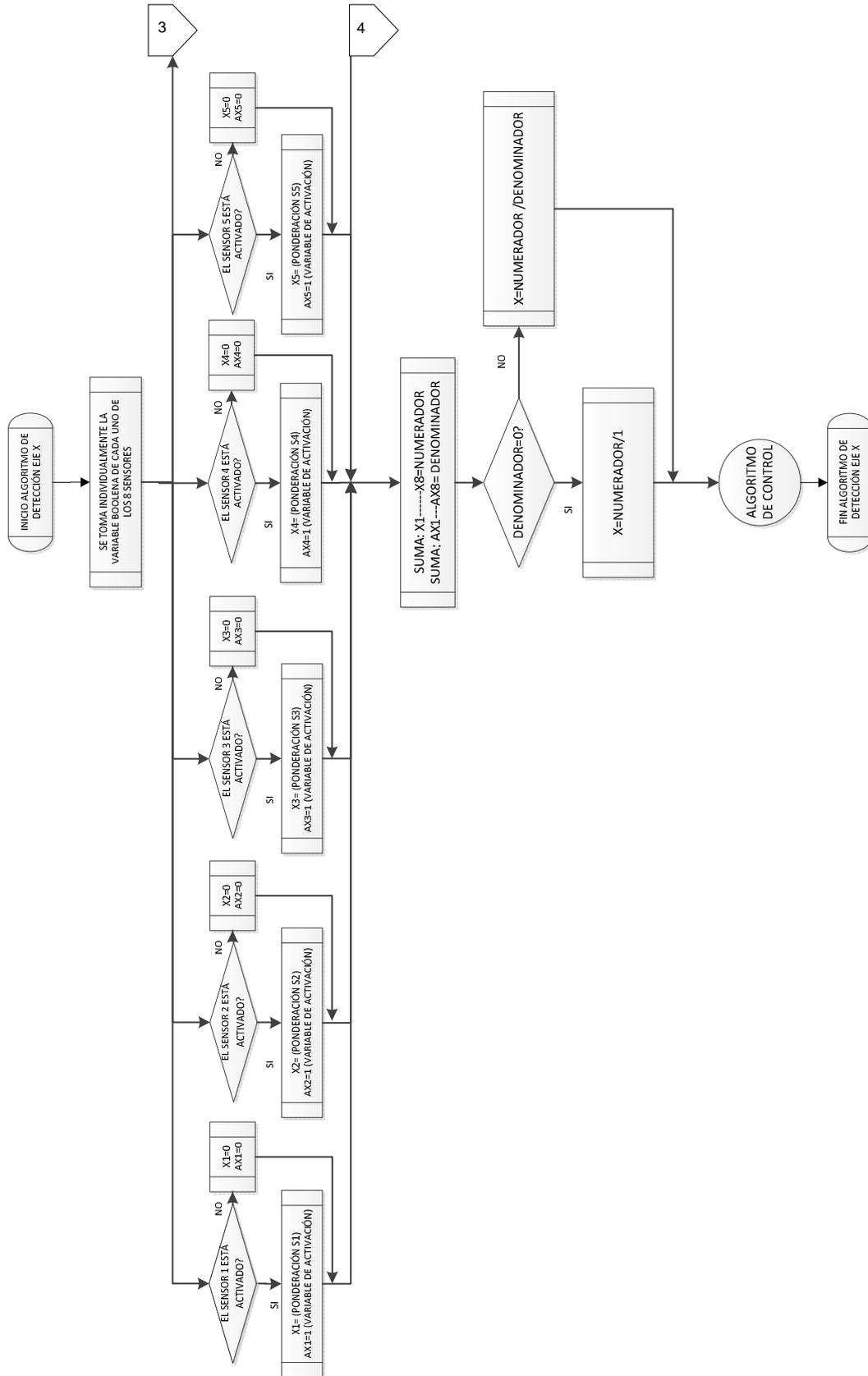


Figura 3.8 Diagrama de flujo del algoritmo de detección para el eje X, PARTE A

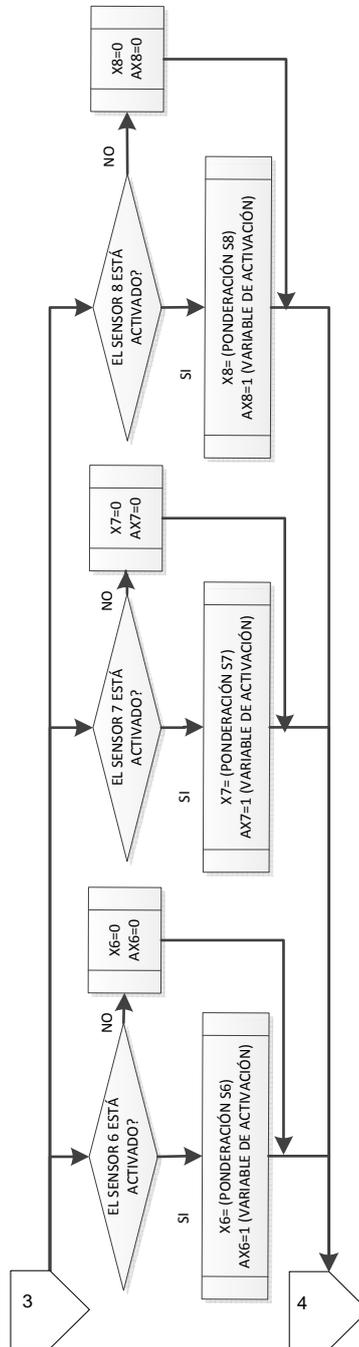


Figura 3.9
Diagrama de flujo del algoritmo de detección para el eje X, PARTE B

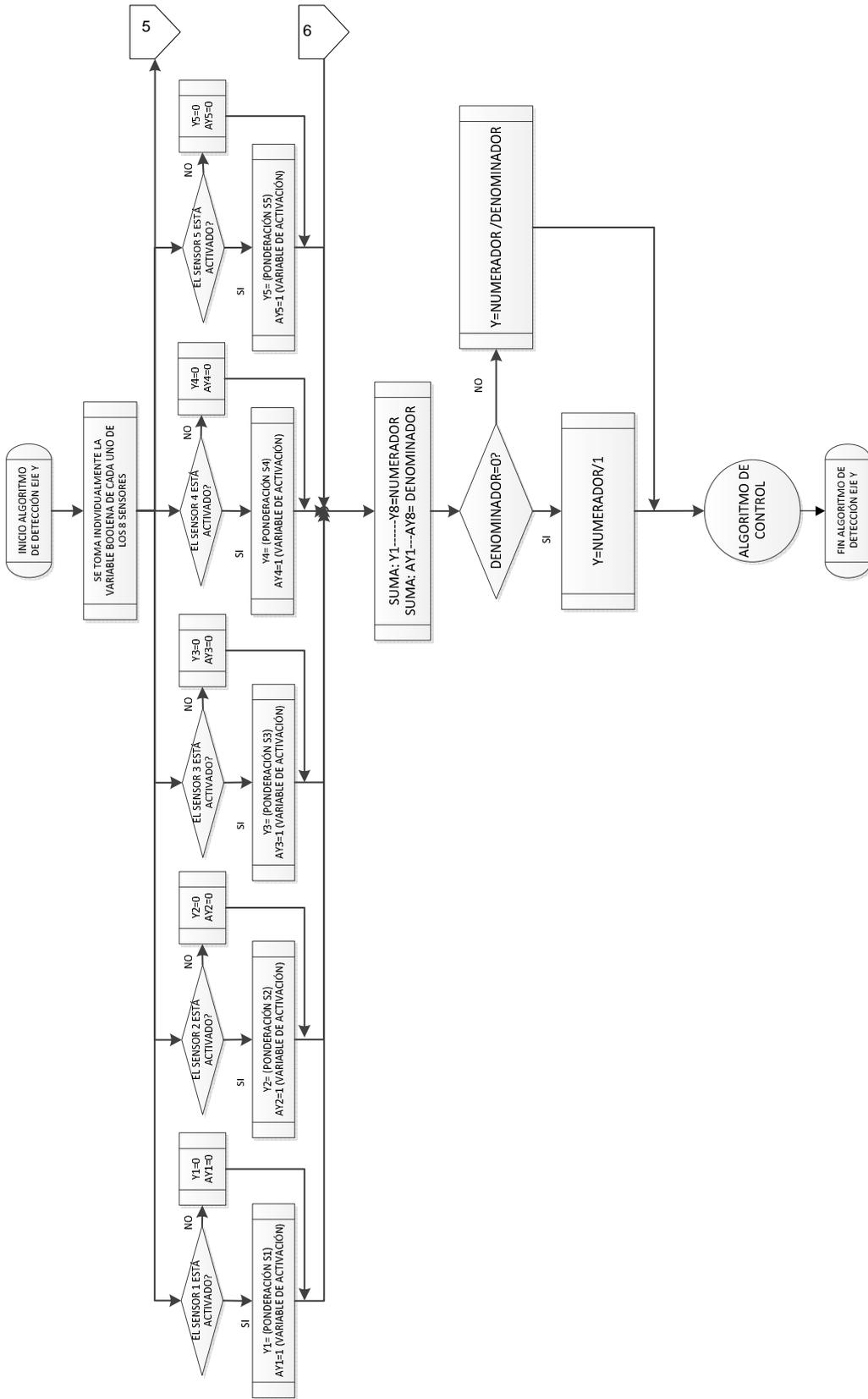


Figura 3.10 Diagrama de flujo del algoritmo de detección para el eje Y, PARTE A

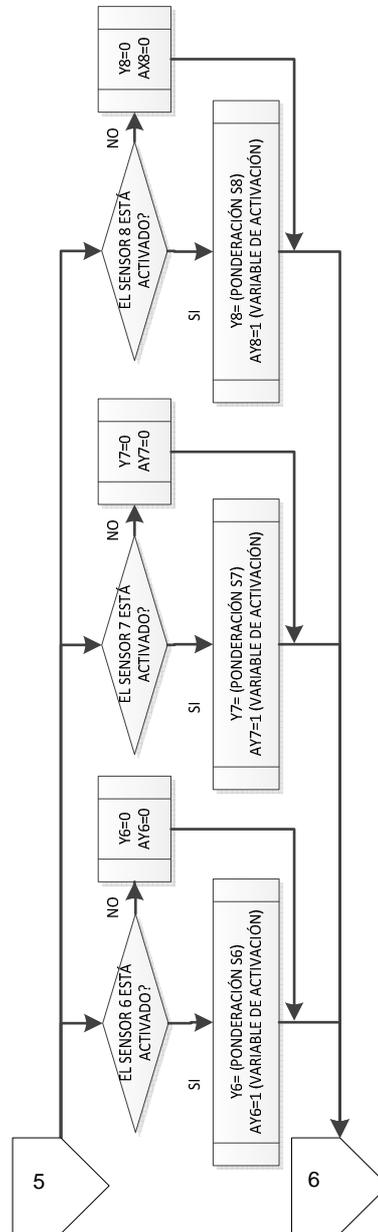


Figura 3.11 Diagrama de flujo del algoritmo de detección para el eje Y, PARTE B

Cabe destacar que la utilización del algoritmo descrito mediante la ecuación (3.1) es el adecuado debido a las limitaciones que se tienen al realizar un seguimiento de un objeto en tierra sin disponer de un sistema de realimentación de posición o de un sistema de visión artificial.

El algoritmo implementado permite obtener una ruta tentativa de seguimiento, la incertidumbre de posición del móvil terrestre en cada uno de los ejes es de 45° como se puede observar en las Figuras 3.5 y 3.6, esta incertidumbre es un limitante en cuanto a realizar un seguimiento fluido se refiere, ya que el cuadricóptero AR Drone va a tener que corregir su movimiento constantemente, no sólo por el hecho de que el móvil terrestre se mueva sino también por la existencia de dicha incertidumbre.

El Algoritmo de detección ha resultado ser funcional ya que cuando todos los sensores se encuentran activados, el resultado de la posición relativa para el seguimiento es $[0,0]$, es decir, 0 en X y 0 en Y, la interpretación de este resultado es que el AR Drone se encuentra dentro del faro infrarrojo y por lo tanto no debe seguir ninguna ruta porque en este caso se encontraría sobre el móvil.

El caso más crítico resulta cuando se activa uno solo de los sensores ya que esto significaría que el AR Drone está a punto de salir del faro infrarrojo, en este caso el resultado de la posición relativa sería máximo.

3.2 PROGRAMACIÓN DEL CONTROLADOR DE SEGUIMIENTO

Como se analizó en el punto 3.1 el algoritmo de control proporciona tanto la magnitud como la dirección de la posición del cuadricóptero con respecto al cono de luz del faro infrarrojo y por ende cuanto debería moverse para lograr situarse sobre el mismo, partiendo de estos resultados se puede realizar tanto una acción directa de control, enviando directamente estas señales al A.R Drone, o bien implementar un controlador que permita realizar un movimiento suave al instante de posicionarse sobre el faro infrarrojo.

Las señales que comandan los ángulos de Roll, Pitch y Yaw, los cuales hacen posible los diferentes movimientos en el cuadricóptero, deben estar en el rango de -1 a 1, siendo 1 el mayor ángulo de cabeceo que es de aproximadamente 12 grados y que consecuentemente determina la mayor velocidad a la que se desplazará. Para realizar un control suave se limitará la salida a un valor inferior a 1 para reducir la posibilidad de que se pierda de vista al faro infrarrojo y también disminuir la inercia cuando se presente un cambio de dirección.

La señal de respuesta del algoritmo de detección se la podría aproximar a un sistema continuo para lo cual se debería incrementar el número de receptores montados en el cuadricóptero, esto implica también que se incrementaría el peso que tendría que elevar. El peso es un factor crítico en la plataforma AR Drone, el cuadricóptero tiene una masa de 320gr sin su casco de protección y de 420 gr con éste, la instrumentación necesaria para realizar el seguimiento que consta de los 9 detectores IR, la placa de procesamiento y la del módulo XBee incrementan el peso del cuadricóptero a 465 gr aproximadamente teniendo en cuenta que el vuelo se va a realizar sin el casco de protección, si se incrementa el número de sensores existe una alta probabilidad que se presenten anomalías en el vuelo en el caso de que el Drone pueda despegar y soportar el sobrepeso.

El diagrama de flujo de la Figura 3.12 indica de manera global el procesamiento interno del software de control que ha sido implementado en LabView, el cual mediante la detección, decodificación e implementación de acciones de control permitirá obtener un vuelo controlado alrededor del móvil, hasta el almacenamiento de los datos resultantes en una variable global, la cual será leída por la DLL para, haciendo uso de C#, enviar los comandos vía WiFi hacia el cuadricóptero.

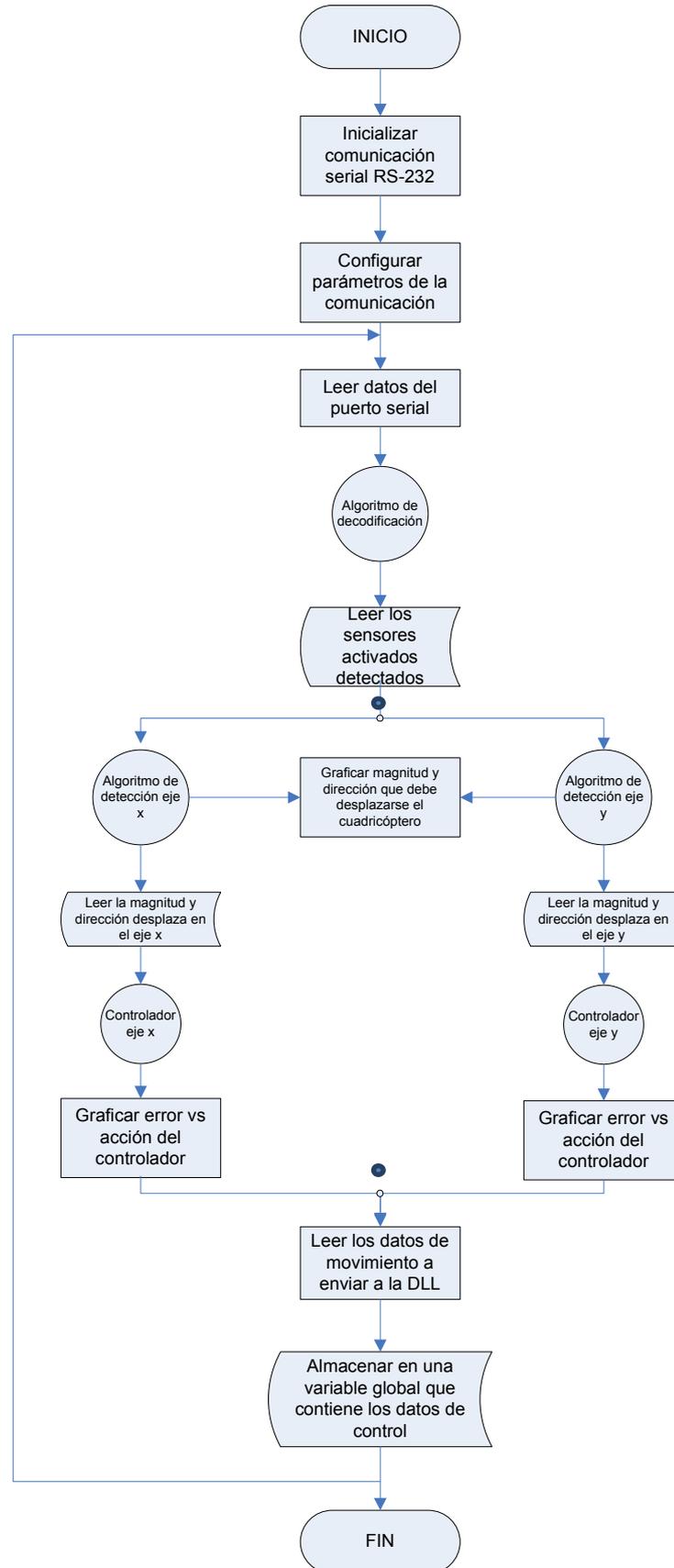


Figura 3.12 Diagrama de bloques general del controlador implementado en LabView

3.2.1 CONTROLADOR DE POSICIÓN PARA LOS EJES X E Y

El algoritmo de detección analiza independientemente cada eje de tal modo que en conjunto permita obtener un radio vector con magnitud y dirección que indique la posición relativa del cuadricóptero con respecto al faro infrarrojo, del mismo modo se tendrá un controlador de la misma topología para cada eje.

Para realizar un control óptimo capaz de posicionar al cuadricóptero sobre el faro, compensando cualquier tipo de perturbaciones y mantenerlo estable es necesario conocer sus parámetros dinámicos. El modelo estándar del A.R. Drone como un cuadricóptero cualquiera, considera la velocidad de las hélices, el empuje aerodinámico, etc. como entradas y los ángulos de Euler como salidas, complicando notablemente su modelado. Para evitar el complicado proceso del modelado clásico los investigadores Tomas Krajník, Vojtech Vonasek, Daniel Fiser, y Jan Faigl de la Universidad de Praga realizaron trabajos con cuadricópteros considerando sus controladores internos, es decir que una vez que el cuadricóptero responda a órdenes remotas de Roll, Pitch, Yaw o Vertical Speed y sea capaz de mantenerse estable se lo puede considerar como un sistema que tiene un ángulo deseado como entrada y ángulos a un cierto instante como salida.

Como resultado de un ángulo se tiene una determinada velocidad sea en los ejes X o Y, empleando la misma metodología se puede descomponer al sistema en sus componentes fundamentales para su movimiento como son los ángulos de Euler además de las velocidades en cada eje, lo que permite identificar los parámetros de cada planta por separado.

El modelo descompuesto del cuadricóptero propuesto por investigadores Tomas Krajník, Vojtech Vonasek, Daniel Fiser, y Jan Faigl tiene la estructura mostrada en la Figura 3.13.

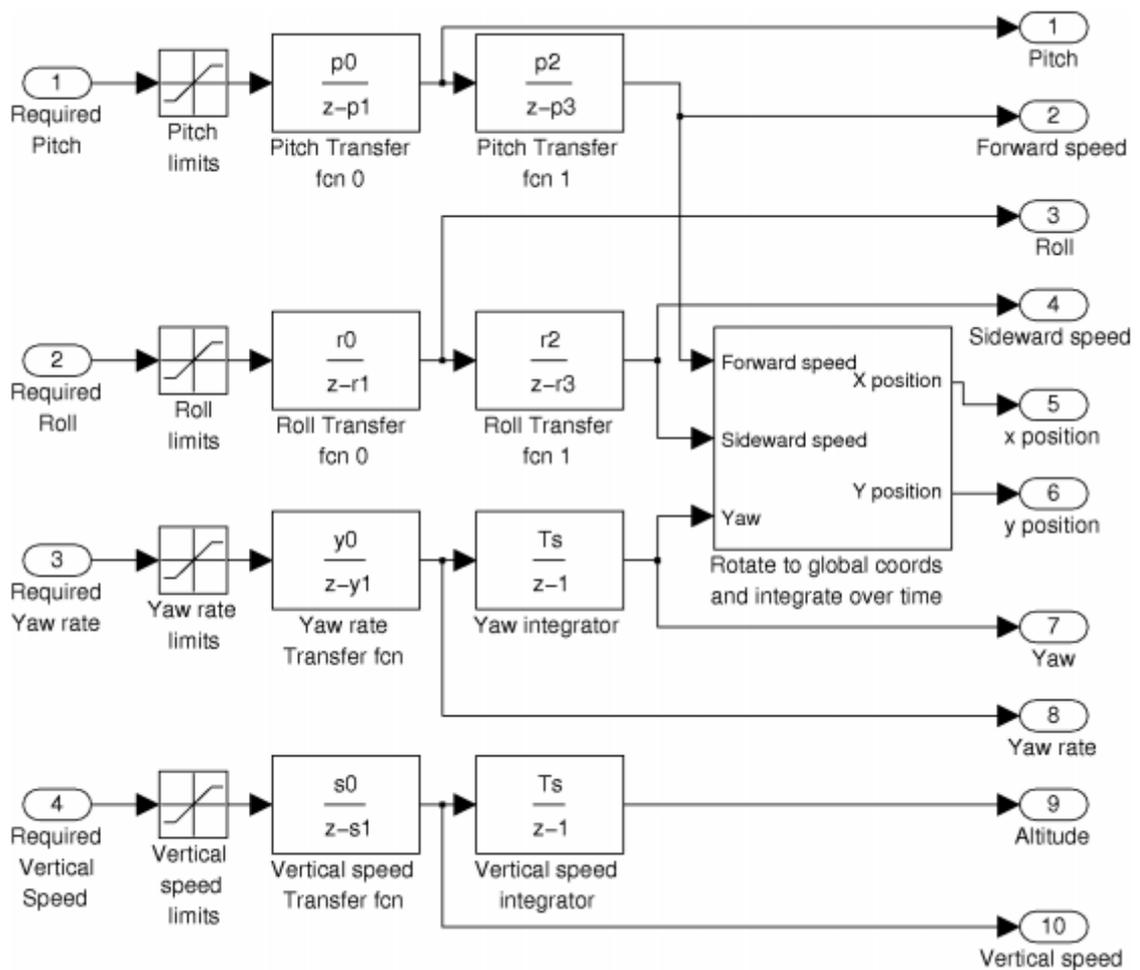


Figura 3.13 Diagrama de bloques de la planta del cuadricóptero A.R Drone, tomado de [23]

3.2.1.1 Identificación de los parámetros de las plantas de Pitch y Roll

Los parámetros de la función de transferencia en Pitch se los puede obtener experimentalmente, colocando un cierto ángulo de Pitch al cual el cuadricóptero tiene que responder con cierta inclinación (cabeceo) lo que representa un movimiento a una cierta velocidad.

La entrada (ángulo requerido) viene a ser una entrada tipo paso la cual cambia de un estado inicial de cero hasta el valor máximo de 12° que es el máximo ángulo de cabeceo de Pitch y que viene a ser un valor que se haya configurado por defecto, con este ángulo se obtienen velocidades razonables, para ángulos superiores la velocidad aumenta considerablemente y se requerirá de un gran

espacio abierto para poder observar su comportamiento y adquirir los suficientes datos para obtener los respectivos parámetros.

De la función de transferencia de Pitch se obtienen las siguientes ecuaciones:

$$\varphi(z) = \frac{P_0}{z - P_1} * U(z) \quad (3.2)$$

$$V(z) = \frac{P_2}{z - P_3} * \varphi(z) \quad (3.3)$$

Las expresiones (3.2) y (3.3) corresponden a los dos bloques de la función de transferencia de Pitch en función de sus entradas y salidas, donde U viene a ser el ángulo requerido, φ es el ángulo de Pitch con el que responde el cuadricóptero y V la velocidad de respuesta. Desarrollando ambas expresiones resulta:

$$\varphi(z) = \frac{1}{z} * P_0 * U(z) + \frac{1}{z} * P_1 * \varphi(z)$$

$$V(z) = \frac{1}{z} * P_2 * \varphi(z) + \frac{1}{z} * P_3 * V(z)$$

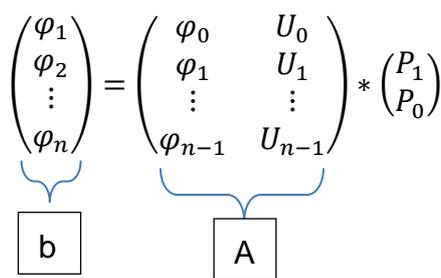
Transformando ambas expresiones a ecuaciones en diferencias se tiene:

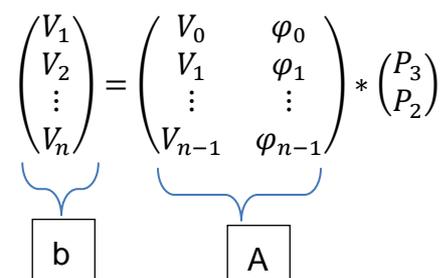
$$\varphi[n] = P_0 * U[n - 1] + P_1 * \varphi[n - 1] \quad (3.4)$$

$$V[n] = P_2 * \varphi[n - 1] + P_3 * V[n - 1] \quad (3.5)$$

Las expresiones (3.4) y (3.5) dan una idea de cómo se relacionan las constantes con las variables de entrada y salida. Tomando la suficiente cantidad de datos se puede plantear un sistema de ecuaciones lineales de donde, mediante métodos estadísticos es posible despejar cada uno de los parámetros de la planta.

Tomando una serie de muestras se puede plantear las siguientes matrices:

$$\begin{pmatrix} \varphi_1 \\ \varphi_2 \\ \vdots \\ \varphi_n \end{pmatrix} = \begin{pmatrix} \varphi_0 & U_0 \\ \varphi_1 & U_1 \\ \vdots & \vdots \\ \varphi_{n-1} & U_{n-1} \end{pmatrix} * \begin{pmatrix} P_1 \\ P_0 \end{pmatrix}$$


$$\begin{pmatrix} V_1 \\ V_2 \\ \vdots \\ V_n \end{pmatrix} = \begin{pmatrix} V_0 & \varphi_0 \\ V_1 & \varphi_1 \\ \vdots & \vdots \\ V_{n-1} & \varphi_{n-1} \end{pmatrix} * \begin{pmatrix} P_3 \\ P_2 \end{pmatrix}$$


Los parámetros P_0 , P_1 , P_2 y P_3 se los determinarán mediante el método de mínimos cuadrados partiendo de los datos de sus respectivas matrices.

El máximo tiempo para la adquisición de los datos es de 0.067 s, que es lo que se demora la computadora en leer el dato proveniente vía WiFi, procesarlo y enviarlo a Excel. De las pruebas realizadas se pudo obtener los resultados que se muestran en las Figuras 3.14, 3.15, 3.16 y 3.17:

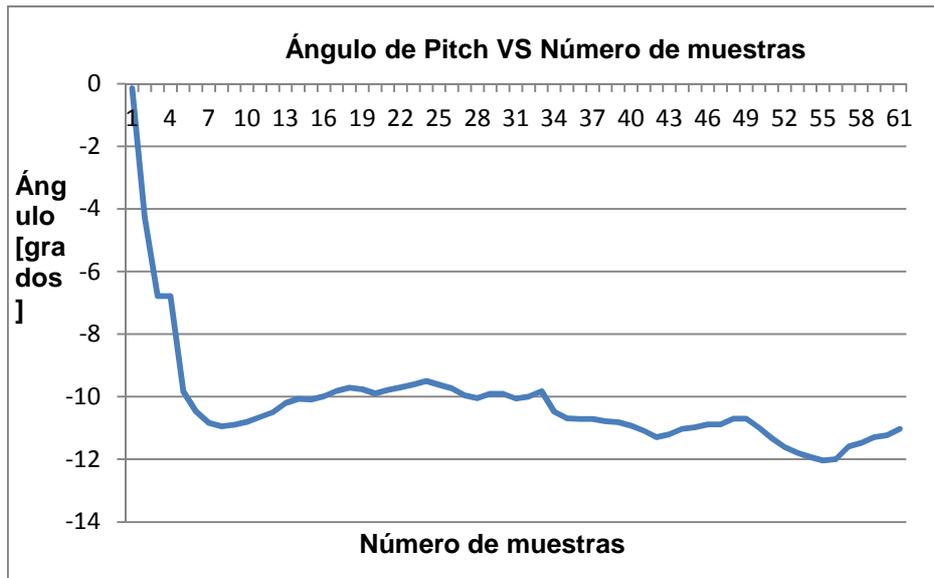


Figura 3.14 Respuesta del ángulo de Pitch a la orden de -12°

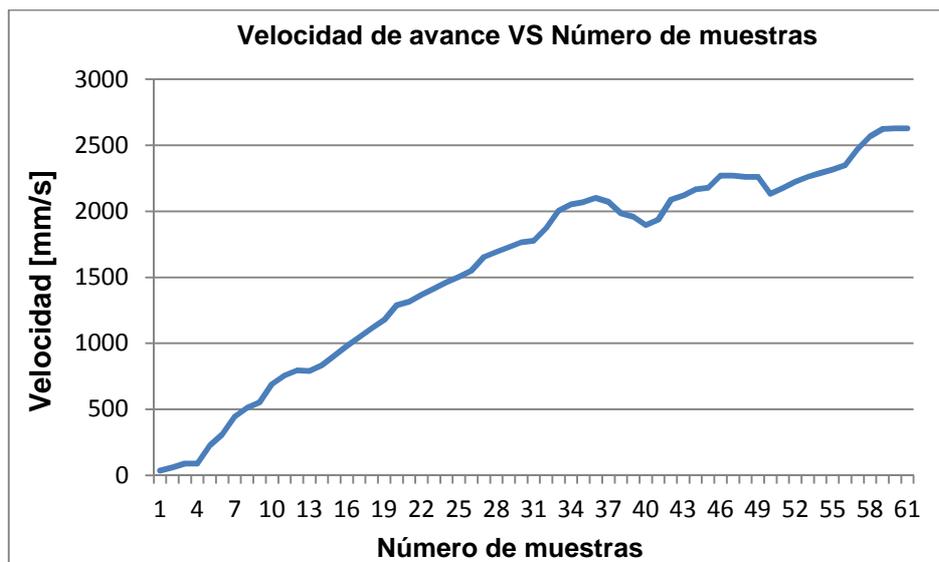


Figura 3.15 Respuesta de velocidad a -12° de Pitch en mm/s

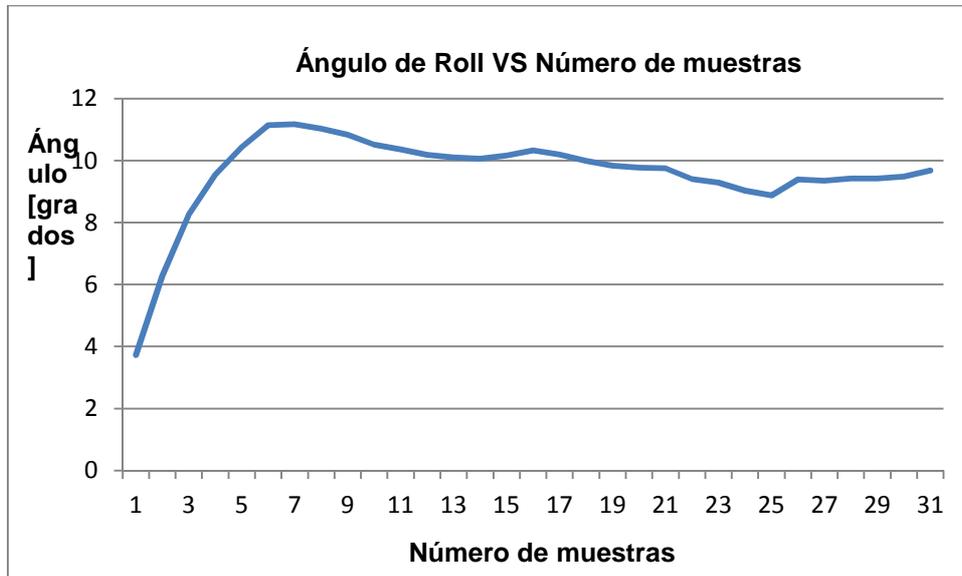


Figura 3.16 Respuesta de Roll a la orden de 12°

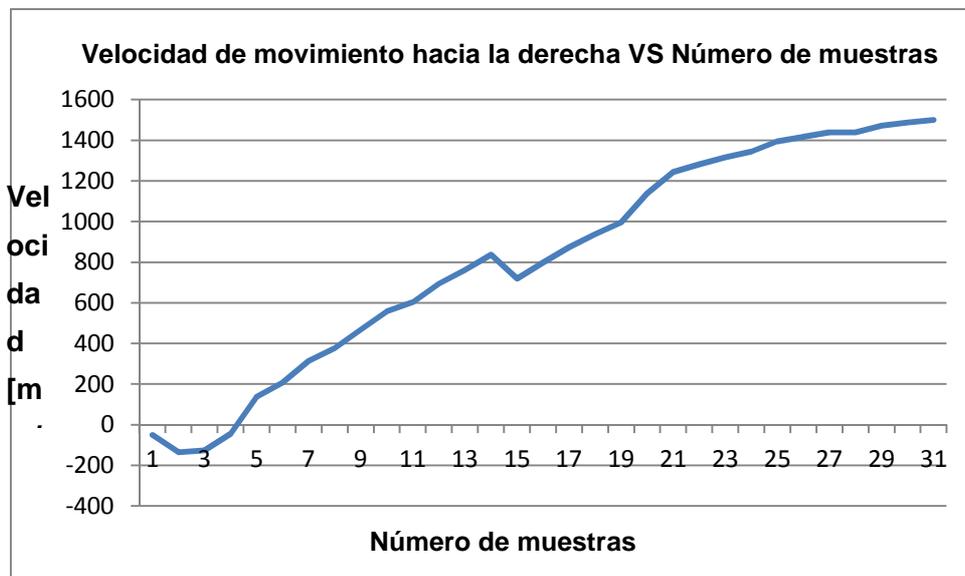


Figura 3.17 Respuesta de velocidad de Roll a 12°

Los datos muestran que los resultados son muy similares tanto en Roll como en Pitch con la única variante que para un ángulo positivo la respuesta en Roll es con una velocidad también positiva es decir se mueve hacia la derecha, mientras que

en Pitch sucede lo contrario, para el mismo ángulo se espera una velocidad negativa es decir que se mueva hacia atrás.

Como se mencionó anteriormente para encontrar el valor de las constantes se empleará el método de mínimos cuadrados, el mismo que permite encontrar los valores de los parámetros que mejor ajustan el modelo a los datos. Este método dice que para un sistema de tipo $Ax'=b$ si y sólo si x' es solución del sistema este debe cumplir que $A^T * Ax' = A^T * b$.

Al tener las matrices A y b de ambas plantas las cuales contienen todos los valores adquiridos, se plantean los siguientes sistemas de ecuaciones luego de aplicar el teorema de los mínimos cuadrados:

$$\begin{pmatrix} 7844,270441 & 8908,236 \\ 8908,236 & 10512 \end{pmatrix} * \begin{pmatrix} P_1 \\ P_0 \end{pmatrix} = \begin{pmatrix} 7812,066985 \\ 8909,124 \end{pmatrix}$$

$$\begin{pmatrix} 273221924,09499 & 1357211,50435311 \\ 1357211,50435311 & 7844,270441 \end{pmatrix} * \begin{pmatrix} P_3 \\ P_2 \end{pmatrix} = \begin{pmatrix} 276069602,60402 \\ 1383085,53086528 \end{pmatrix}$$

Resolviendo los sistemas de ecuaciones se obtienen los parámetros P_0 , P_1 , P_2 y P_3 de la planta de Pitch, y el diagrama de bloques resulta el que se observa en la Figura 3.18:



Figura 3.18 Diagrama de bloques de Pitch.

El diagrama de bloques varía si las características aerodinámicas del cuadricóptero se alteran y será necesario realizar todo el procedimiento mencionado para hallar los nuevos parámetros de la planta.

La planta de Roll tiene exactamente el mismo modelo, por lo tanto para determinar sus parámetros se debe seguir el mismo procedimiento, es decir adquirir los datos, plantear las matrices A y b y aplicar el método de mínimos cuadrados para resolver el sistema de ecuaciones con lo que se obtiene la planta representada en el diagrama de bloques de la Figura 3.19:

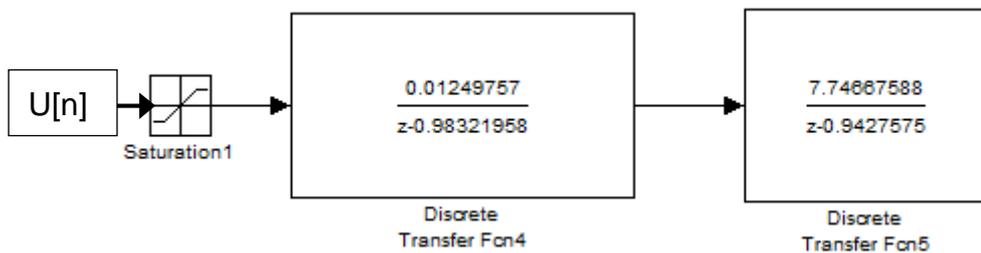


Figura 3.19 Diagrama de bloques de Roll.

La salida de los diagramas de bloques de las Figuras 3.18 y 3.19 indican la velocidad del sistema, para determinar la posición instantánea del cuadricóptero es necesario integrar la velocidad en el tiempo. El integrador más utilizado es el que emplea el método Tustin o integración trapezoidal, la planta general para Roll y para Pitch resulta como lo indica la Figura 3.20.

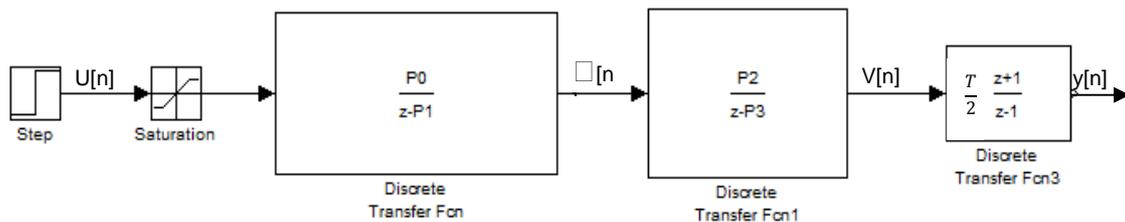


Figura 3.20 Diagrama de bloques con integrador trapezoidal.

3.2.1.2 Diseño del controlador de Pitch y Roll por realimentación de estados

Como indica la Figura 3.13 se conoce y se puede medir las variables de ángulo, velocidad y determinar mediante un integrador la distancia recorrida, los cuales representan los estados del sistema por lo tanto el control que se ha decidido implementar es un control por realimentación de estados.

Entre las ventajas que ofrece la realimentación de estados están el utilizar toda la información del sistema para calcular una salida adecuada, es decir que toma en cuenta el ángulo de Pitch o Roll que tiene el cuadricóptero en un instante determinado, el cual tiene un efecto en la velocidad y a partir de esta última variable se estima la posición instantánea. Otra ventaja es que no utiliza derivadores que en ciertos casos tienden a hacer inestable al sistema, sino únicamente ganancias proporcionales.

Los sistemas que representan tanto a la respuesta de Pitch como de Roll son de tipo discreto y se pueden expresar en función de sus estados como indican las expresiones (3.6) y (3.7).

$$x[n + 1] = A * x[n] + B * u[n] \quad (3.6)$$

$$y[n] = C * x[n] \quad (3.7)$$

Para realizar el presente análisis se asume que la señal de entrada es una función paso. El esquema estándar de realimentación de estados tiene la forma de la Figura 3.21:

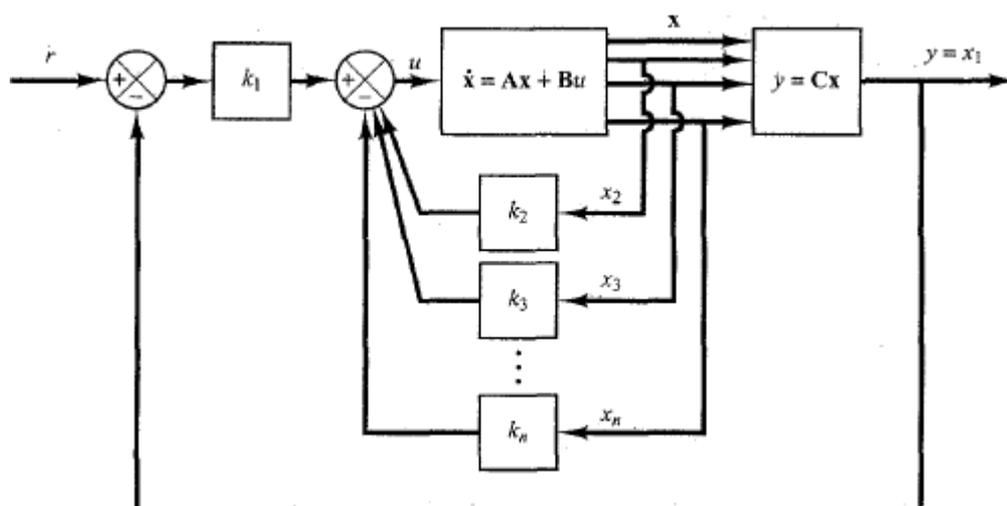


Figura 3.21 Diagrama de bloque de realimentación de estados, tomado de [24]

Aplicando el diagrama de realimentación de estados a nuestro sistema se tiene el diagrama de bloques de la Figura 3.22, teniendo en cuenta que los parámetros fueron determinados previamente y que es aplicable tanto para el control de la posición en el eje Y (Pitch) como en el eje X (Roll):

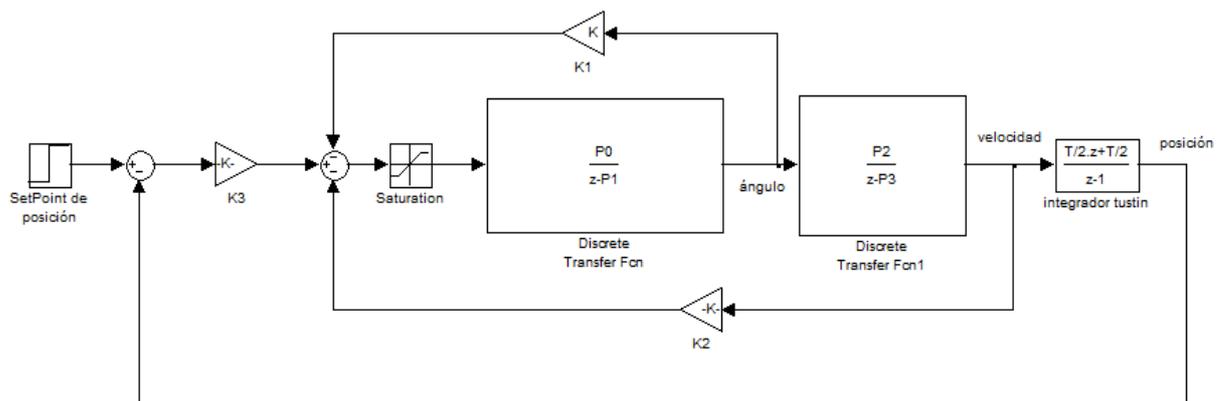


Figura 3.22 Diagrama de bloques de realimentación de estados para Roll o Pitch

El vector de estados del sistema se compone de tres variables: el ángulo, velocidad y posición.

$$x = \begin{pmatrix} \varphi \\ v \\ d \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

Las ecuaciones que gobiernan cada uno de estos estados se deducen del diagrama de bloques tanto de Pitch como de Roll, que de forma literal se obtiene:

$$\varphi[n + 1] = P_0 * U[k] + P_1 * \varphi[k] \quad (3.8)$$

$$v[n + 1] = P_2 * \varphi[n] + P_3 * v[n] \quad (3.9)$$

La posición del cuadricóptero se estima mediante la integración de la velocidad, al tener datos discretos el mejor método de integración es el Tustin o integración trapezoidal con lo que se determina la expresión de la posición así:

$$d(z) = \frac{Tm}{2} * \frac{z+1}{z-1} * v(z)$$

$$z * d(z) - d(z) = \frac{Tm}{2} * (z * v(z) + v(z))$$

Expresando en ecuaciones en diferencias:

$$d[n+1] - d[n] = \frac{Tm}{2} * (v[n+1] + v[n])$$

Reemplazando la expresión (3.9):

$$d[n+1] = d[n] + \frac{Tm}{2} * (P_2 * \varphi[n] + P_3 * v[n] + v[n]) \quad (3.10)$$

Colocando las expresiones (3.8), (3.9) y (3.10) en función del vector de estados se tiene:

$$x_1[n+1] = P_0 * U[n] + P_1 * x_1[n]$$

$$x_2[n+1] = P_2 * x_1[n] + P_3 * x_2[n]$$

$$x_3[n+1] = \frac{Tm}{2} * P_2 * x_1[n] + \frac{Tm * (P_3 + 1)}{2} * x_2[n] + x_3[n]$$

Planteando el sistema en función de sus variables de estado resulta:

$$\begin{pmatrix} x_1[n+1] \\ x_2[n+1] \\ x_3[n+1] \end{pmatrix} = \begin{pmatrix} P_1 & 0 & 0 \\ P_2 & P_3 & 0 \\ \frac{Tm * P_2}{2} & \frac{Tm * (P_3 + 1)}{2} & 1 \end{pmatrix} * \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} P_0 \\ 0 \\ 0 \end{pmatrix} \quad (3.11)$$

$$y[n] = (0 \quad 0 \quad 1) * \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad (3.12)$$

El sistema planteado en las expresiones (3.11) y (3.12) es aplicable tanto a Pitch como a Roll donde lo único que cambia son los parámetros de cada planta. Para obtener los sistemas en función de sus variables de estados se reemplaza los parámetros indicados en las Figuras 3.15 y 3.16 en la expresión (3.11).

$$\begin{pmatrix} x_1[n+1] \\ x_2[n+1] \\ x_3[n+1] \end{pmatrix} = \begin{pmatrix} 0.8883 & 0 & 0 \\ -10.6388 & 0.9576 & 0 \\ -0.3564 & 0.0656 & 1 \end{pmatrix} * \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} 0.0947 \\ 0 \\ 0 \end{pmatrix} \quad (3.13)$$

$$\begin{pmatrix} x_1[n+1] \\ x_2[n+1] \\ x_3[n+1] \end{pmatrix} = \begin{pmatrix} 0.9832 & 0 & 0 \\ 7.7467 & 0.9428 & 0 \\ 259.5136 & 0.0651 & 1 \end{pmatrix} * \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} 0.0124975 \\ 0 \\ 0 \end{pmatrix} \quad (3.14)$$

La expresión (3.13) corresponde a las matrices de estado de la planta de Pitch mientras que la expresión (3.14) a las matrices de estado de la planta de Roll.

Una vez determinadas las matrices de estado es necesario determinar las ganancias de realimentación. Uno de los métodos más sencillos es de la asignación de polos el cual consiste en añadir 3 polos al sistema, uno por cada estado, con los cuales la respuesta del sistema pueda cumplir ciertas condiciones de máximo sobre impulso y tiempo de establecimiento, una vez asignados los polos las ganancias se pueden calcular mediante la fórmula de Ackerman la cual se encuentra implementada en Matlab, siendo el comando “acker(a,b,j)” el que permite determinar las ganancias donde los requisitos indispensables para su uso son las matrices A y b de los sistemas expresados en variables de estados y el vector “j” que contiene los polos deseados para los sistemas.

Para emplear el método de la ubicación de polos es requisito que el sistema se encuentre en la forma canónica controlable. Los sistemas definidos por las ecuaciones (3.13) y (3.14) se pueden transformar a la forma canónica controlable con una matriz de transformación [24]:

$$T = MW$$

Donde: $M = [B : BA : \dots : A^{n-1}B]$

$$Y \quad W = \begin{bmatrix} a_{n-1} & a_{n-2} & \dots & a_1 & 1 \\ a_{n-2} & a_{n-3} & \dots & 1 & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ a_1 & 1 & \dots & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 \end{bmatrix}$$

Los elementos mostrados en la matriz W , a_i , son los coeficientes de la ecuación característica de la matriz A de la planta de Roll o de Pitch y n viene a ser el orden del sistema que en este caso sería de orden 3, los coeficientes de la ecuación característica se determinan mediante la ecuación (3.15).

$$|\lambda I - A| = \lambda^n + a_1\lambda^{n-1} + \dots + a_{n-1}\lambda + a_n = 0 \quad (3.15)$$

Para realizar la transformación a la forma canónica se emplea la matriz de transformación como lo indica la ecuación (3.16) [24].

$$T^{-1}AT = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ -a_n & -a_{n-1} & -a_{n-2} & \dots & -a_1 \end{bmatrix} \quad (3.16)$$

La forma canónica controlable para la planta de Pitch y de Roll se indica en las expresiones (3.17) y (3.18) respectivamente.

$$\begin{pmatrix} x_1[n+1] \\ x_2[n+1] \\ x_3[n+1] \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 2.8459 & -2.6965 & 0.8506 \end{pmatrix} * \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad (3.17)$$

$$\begin{pmatrix} x_1[n+1] \\ x_2[n+1] \\ x_3[n+1] \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 2.926 & -2.8529 & 0.9269 \end{pmatrix} * \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad (3.18)$$

El cálculo de las ganancias necesarias para la realimentación será desarrollado en el Capítulo 4.

3.2.1.3 Diagrama de flujo de los controladores de Pitch y Roll

El control diseñado es un controlador de posición es decir que si el usuario proporciona una coordenada en el plano X e Y el cuadricóptero deberá moverse hacia allá partiendo de una posición inicial. Las coordenadas serán proporcionadas automáticamente por el algoritmo de detección explicado en el punto 3.1.3 haciendo que el cuadricóptero se dirija hacia las coordenadas del faro y se posicione sobre el mismo. Si el faro infrarrojo se mueve sus coordenadas cambian y el A.R Drone lo seguirá hasta posicionarse sobre él cumpliendo así el objetivo de seguir a un móvil en tierra.

Para implementar el algoritmo de seguimiento es necesario realizar una convención en cuanto a los ejes de movimiento del cuadricóptero. El A.R Drone proporciona toda la información de ángulos y velocidades basándose en los ejes dispuestos como lo indica la Figura 3.23.

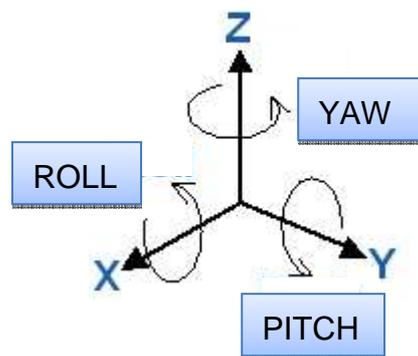


Figura 3.23 Roll, Pitch y Yaw en ejes cartesianos

El lazo de control de seguimiento para el eje Y permite las opciones de manejar al cuadricóptero tanto en modo manual como en automático desde LabView,

teniendo en cuenta que los datos resultantes serán escritos en una librería dinámica DLL para ser utilizados por el software actuador como se lo explicará más adelante.

El controlador deberá tener en cuenta los siguientes aspectos:

1. En modo manual es posible setear el ángulo de Roll, Pitch, Yaw y Gaz (ascensión) en un rango de 1 a -1 que es como reconoce el cuadricóptero A.R. Drone a las órdenes de navegación, el cual en su procesador los traduce de 12° a -12° según sea su configuración.
2. En modo automático se podrá realizar un control de posición ordenado por el usuario, es decir que las coordenadas hacia donde se moverá el cuadricóptero podrán ser ingresadas manualmente, o un control de seguimiento en donde las coordenadas son proporcionadas por el algoritmo de detección del faro infrarrojo.
3. Antes de iniciar el modo automático es necesario posicionar al cuadricóptero sobre el faro para proporcionar un punto de origen al sistema de coordenadas, de esta manera éste podrá recorrer una cierta distancia dependiendo de cuanto se aleje el faro. Cada vez que el cuadricóptero alcance su objetivo será necesario reiniciar el sistema de coordenadas y el integrador de la velocidad para que la posición a la cual deba moverse no se altere debido a sus valores previos, es decir cuando el cuadricóptero se posicione sobre el faro estará en la posición (0,0).

El control por realimentación de estados permite un movimiento relativamente suave hacia cualquier posición, compensando anomalías que se presenten durante el vuelo de la mejor manera posible. El diagrama de flujo del controlador implementado en LabView se muestra en la Figura 3.25.

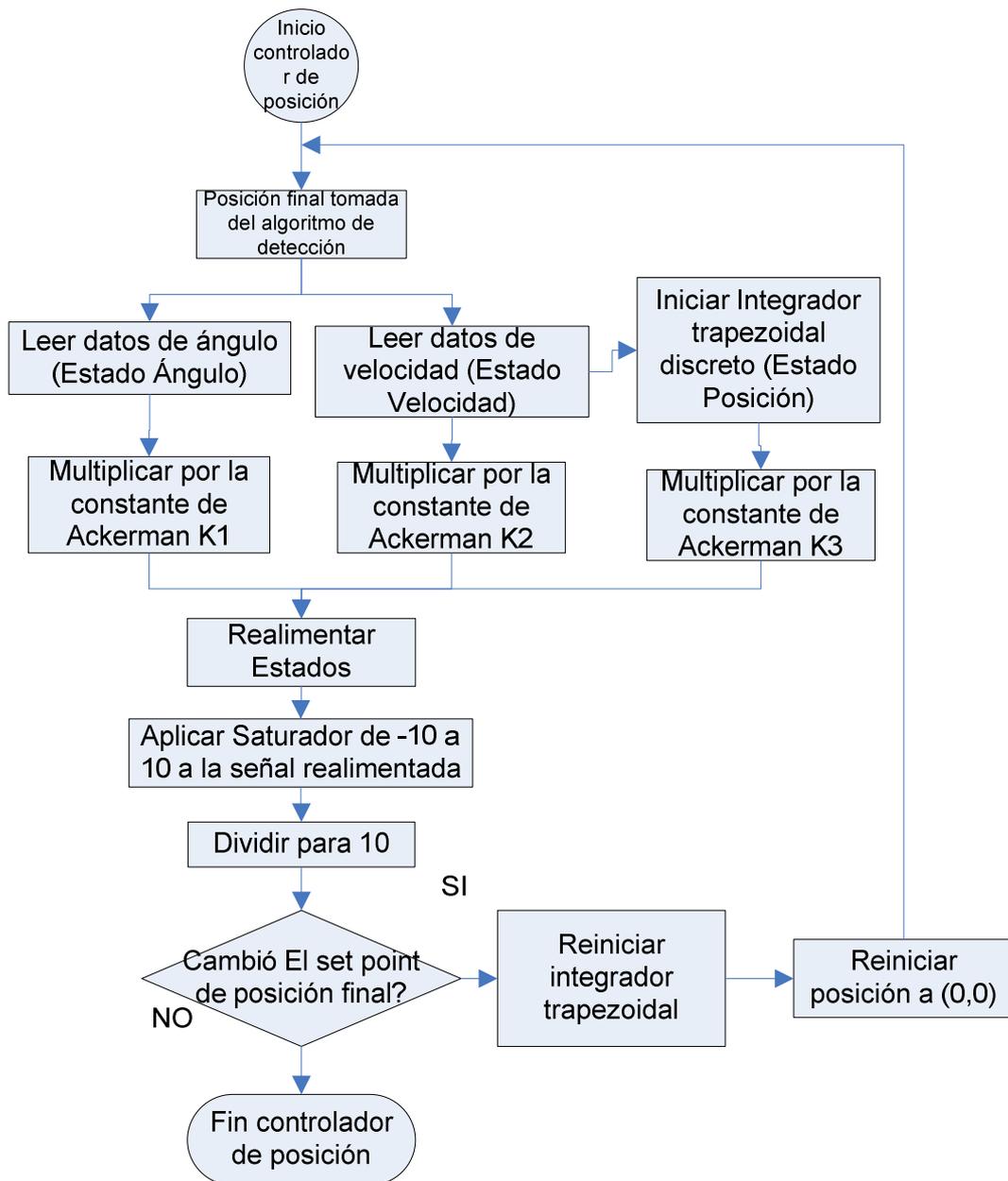


Figura 3.24 Diagrama de flujo de la realimentación de estados

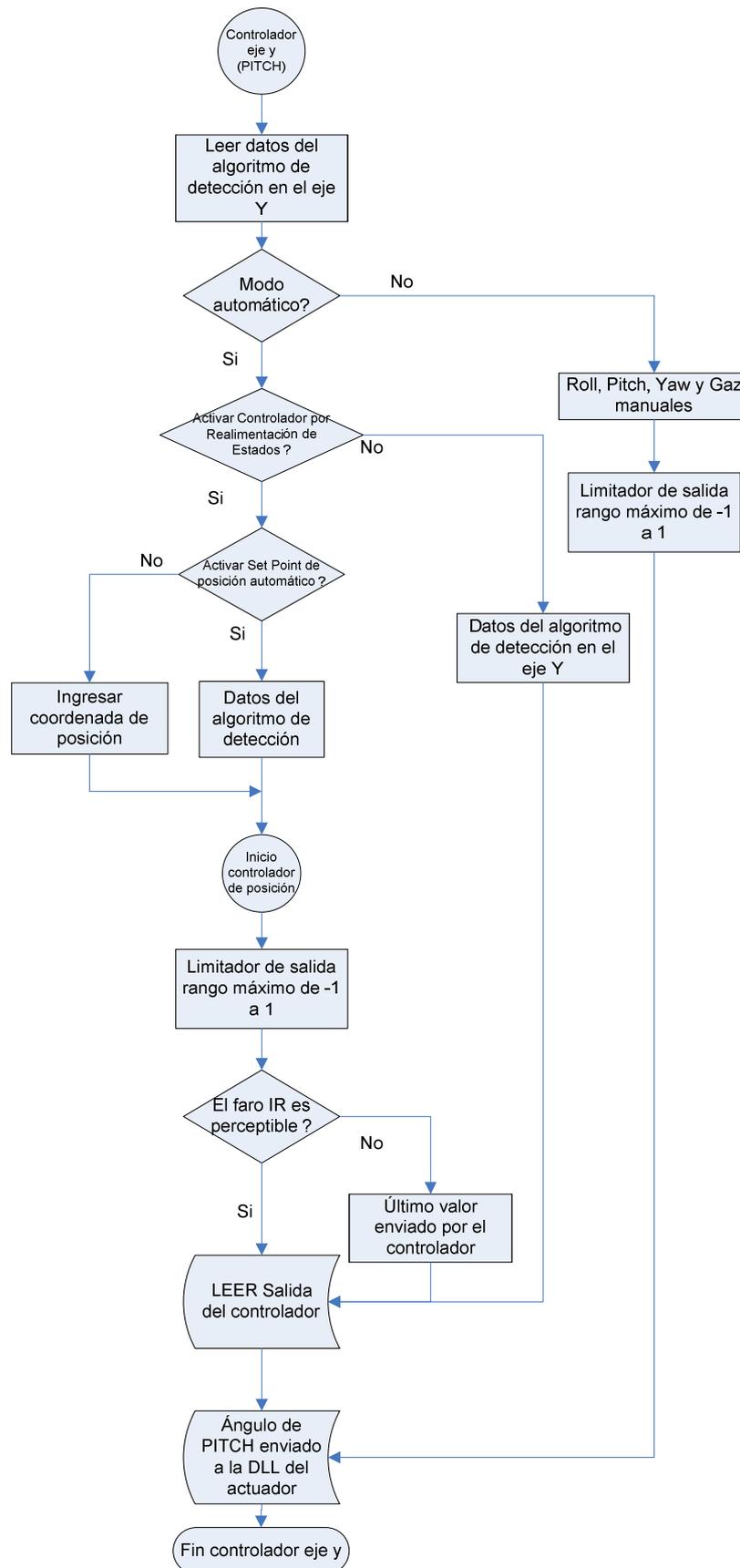


Figura 3.25 Diagrama de flujo del controlador para PITCH

Para contrarrestar el efecto de la inercia debido a las velocidades que puede alcanzar el A.R. Drone, se limitará la salida del controlador a un valor inferior de su salida máxima, es decir valores menores que 1 y mayores que -1.

En caso de un comportamiento errático del cuadricóptero se cuenta con el modo manual el cual se lo puede utilizar tanto desde LabView como desde C#. Para el pilotaje manual del AR Drone en C#, se hace uso del teclado.

Análogamente como se hizo en el eje Y, el control realizado para el eje X (Roll) resulta ser prácticamente idéntico, se necesitará leer los datos proporcionados por el algoritmo de seguimiento en el eje X y procesarlos para hacer uso de otro controlador para este eje y deberá actuar sobre el ángulo de Roll limitando su salida a valores inferiores a 1 y superiores a -1.

Los controladores tanto del eje X como del eje Y actuarán en conjunto para proporcionar al cuadricóptero una dirección adecuada, de tal modo que trate de no perder de vista al faro infrarrojo y limitando en lo posible comportamientos anómalos.

Ambos controladores empiezan a integrar la velocidad instantánea de cada eje para determinar la posición en un instante determinado, cabe resaltar que el integrador empieza a funcionar por orden del operador una vez que se haya logrado estabilizar manualmente al cuadricóptero sobre el faro. Al momento del despegue el vuelo es bastante errático hasta que se pueda vencer la inercia, este comportamiento depende fundamentalmente del estado aerodinámico de las hélices, las cuales tienden a deformarse después de un determinado número de horas de vuelo. El controlador para el eje X se lo muestra en el diagrama de flujo de la Figura 3.26.

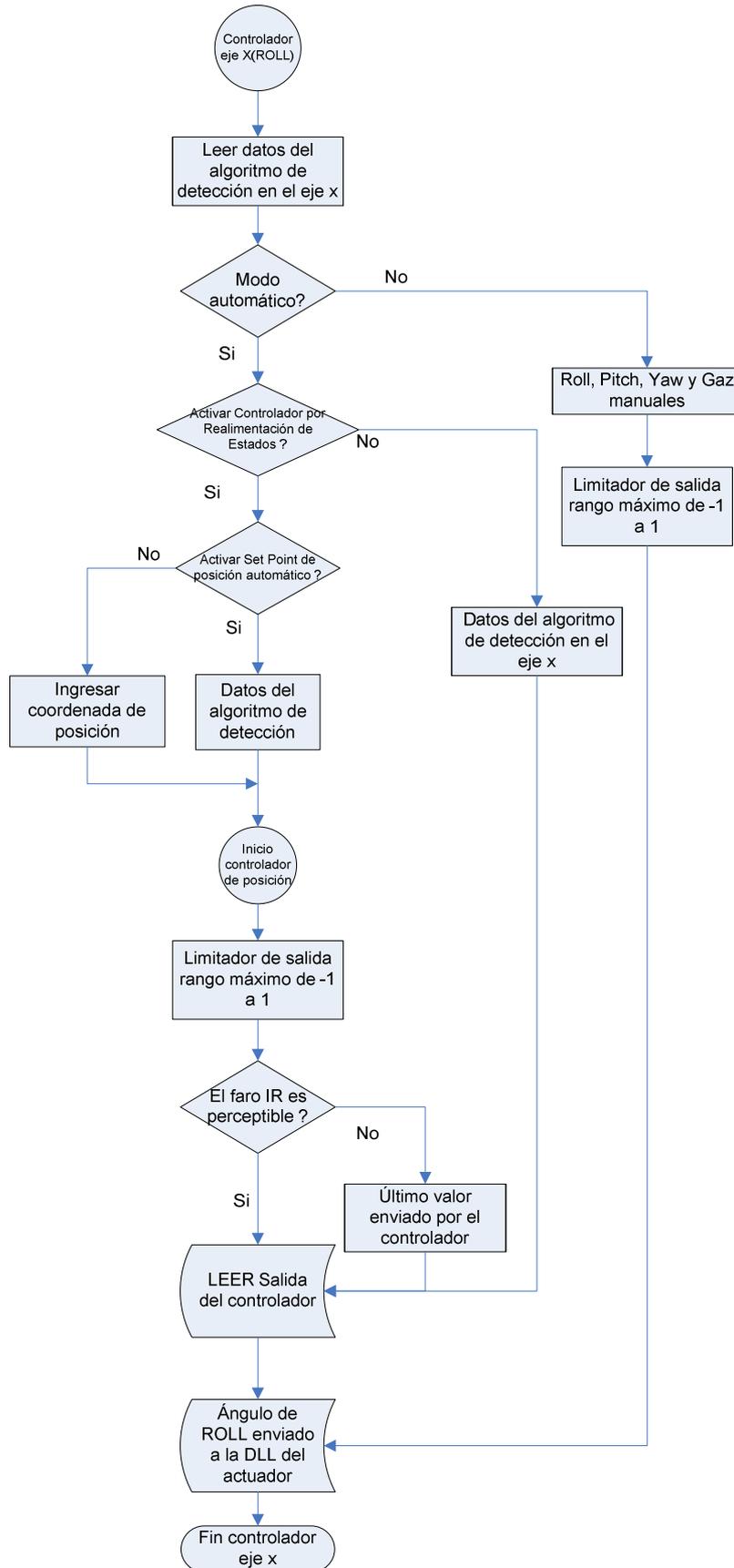


Figura 3.26 Diagrama de flujo del controlador de seguimiento en el eje x

3.3 SOFTWARE TRANSMISOR DE DATOS PARA EL SEGUIMIENTO

Como se explicó en las características del cuadricóptero AR Drone, éste se comunica con el medio externo mediante el uso de un chip WiFi que crea su propia red y permite conectarlo con cualquier dispositivo que realice este tipo de comunicación.

Cuando el AR Drone está encendido crea un punto de acceso local ad – hoc WiFi que debe ser unido al computador para poder realizar cualquier acción de control, en esta red el AR Drone se asigna a sí mismo una dirección IP estática y abre un puerto de comunicaciones predefinido; en la Figura 3.27 se muestra los principales canales de comunicación. La trama de video y los datos de los sensores son enviados del cuadricóptero hacia el computador y los comandos de control para el movimiento desde la PC hacia el AR Drone.

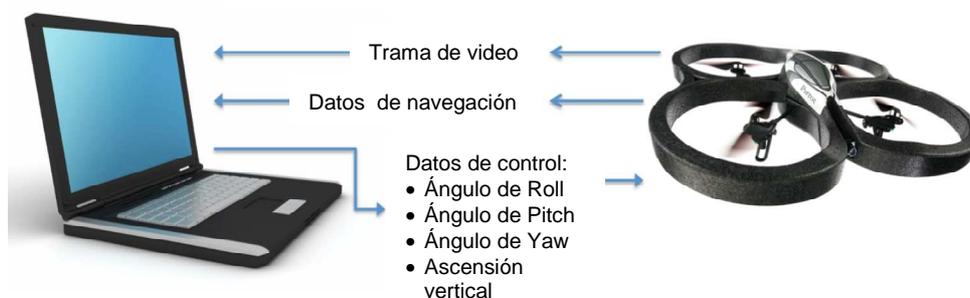


Figura 3.27 Flujo de información entre el AR Drone y el computador

Los controladores tanto del eje X como del eje Y proporcionan una salida que se encuentra en el mismo rango, teniendo como máximos los valores de -1 a 1, estos valores deben ser enviados vía WiFi hacia el AR Drone.

Realizar un algoritmo para implementar una comunicación WiFi con el cuadricóptero mediante LabView resulta complejo es por ello que se decidió tomar como referencia un software denominado “ARDroneControl.Net” elaborado en C#;

una plataforma de programación de Visual Studio, y desarrollado por Thomas Endres, Steve Hopley y Julien Vinel, el cual será el encargado de enviar los datos del control y recibir los datos de navegación vía WiFi.

3.3.1 DESCRIPCIÓN DEL ARCHIVO DLL DEL TRANSMISOR DE DATOS

El controlador y el actuador están programados en software aparentemente incompatibles, con interfaces y plataformas de programación totalmente diferentes sin embargo la manera más sencilla para interconectarlos es haciendo uso de un archivo DLL.

Un archivo DLL (Dynamic Library Link) es un módulo componente de un programa que ejecuta alguna función, los archivos DLL ejecutan acciones o rutinas de uso frecuente en Windows, y un mismo archivo DLL puede ser usado por varios programas al mismo tiempo, por ejemplo el procesador de palabras, la hoja de cálculo y otros programas pueden usar un mismo archivo DLL para desplegar el cuadro de diálogo “Abrir”, cada vez que use este comando.

Gracias a ese esquema modular, hay muchas funciones que los creadores de software no tienen que incluir en sus programas, cuando un programa necesita enviar un documento a la impresora, simplemente llama el archivo DLL respectivo (que ya Windows instaló) para que éste cargue y ejecute la tarea. De esa forma, los programas son más pequeños y se ahorra espacio en el disco duro.

El hecho de que estos módulos de rutinas (Archivos DLL) no sean parte de programas, sino que se guardan como archivos independientes, también optimiza el uso de la memoria RAM. Una DLL se carga en la memoria RAM y se ejecuta únicamente cuando un programa lo llama para que realice una función.

3.3.2 CREACIÓN DEL ARCHIVO DLL EN LABVIEW

El archivo DLL consta de algunas funciones, las cuales deberán ser principalmente la de llamar a la interfaz de LabView, enviar y recibir los datos desde el programa en C#.

La función “LAUNCH” será la encargada de llamar al VI principal, el cual contiene todos los procesos necesarios para realizar el control del cuadricóptero, su programación se describe en la Figura 3.28 y se ejecuta una sola vez en caso de ser llamada.

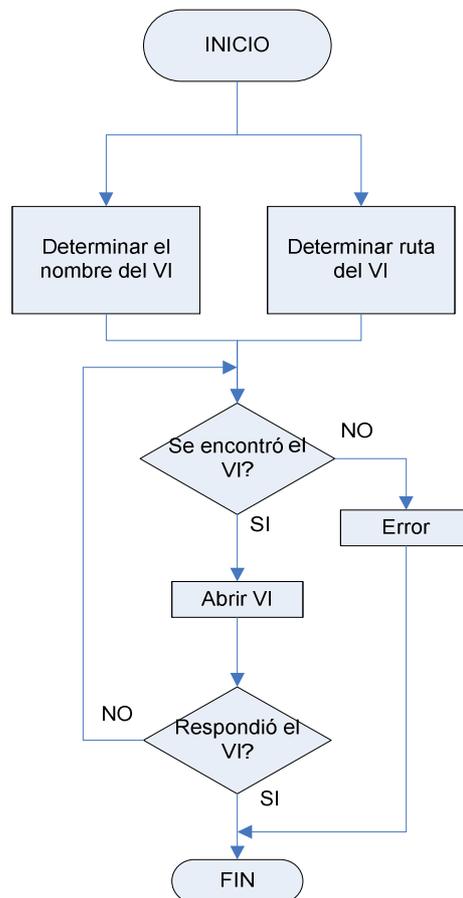


Figura 3.28 Diagrama de flujo de la función LAUNCH

Para que los datos resultantes del VI principal se escriban en la DLL se realizará la función “GET DATA”, la cual mediante el uso de una variable global almacena todos los resultados en un arreglo matricial y los escribe en la DLL de donde el ARDroneControl.Net los tomará y enviará hacia el cuadricóptero.



Figura 3.29 Función GET DATA

Es posible que el programa ARDroneControl.Net también se comunique con LabVIEW, por ejemplo se podría enviar los datos de navegación (altura, velocidad, ángulos de Roll, Pitch y Yaw, etc.) que constantemente son enviados vía WiFi desde la placa madre del AR Drone, luego los escribiría en el archivo DLL y posteriormente serían leídos en LabVIEW. La función “SET PARAMETERS” será la encargada de leer dichos datos desde la DLL y enviarlos hacia LabVIEW para ello deberá conocer la ruta del VI y en que variables se van a almacenar tales datos, la Figura 3.30 indica el algoritmo de la función.

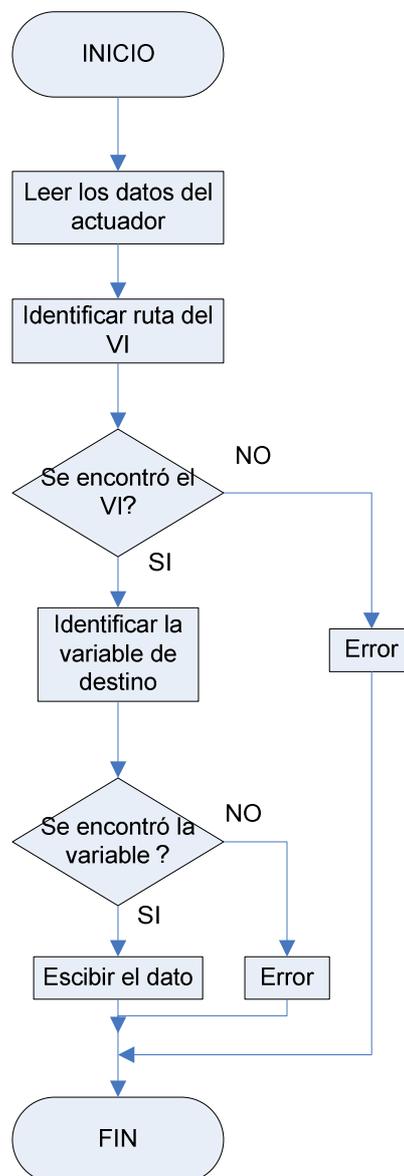


Figura 3.30 Diagrama de flujo de la función SET PARAMETERS

El VI principal tiene el nombre de “Sensores” es el encargado de procesar la información transmitida por la placa emisora del cuadricóptero, realizar el algoritmo de seguimiento e implementar los controladores para cada eje, para evitar cualquier conflicto de datos este VI al igual que las demás funciones que componen la DLL deben formar parte de un solo proyecto.

Antes de crear la DLL es necesario identificar los VIs que van a ser exportados, los mismos que el usuario puede acceder como funciones de una librería compartida como son las funciones “Launch”, “SetParameters” y “GetData”, además de éstos, los VIs dinámicos y archivos de soporte siempre estarán incluidos en la DLL, como es el caso del VI principal llamado “Sensores”, subVIs o los archivos de imagen que corresponden a los botones del programa. Finalmente se da una dirección y un nombre a la DLL para poder ser llamada desde el ARDroneControl.Net.

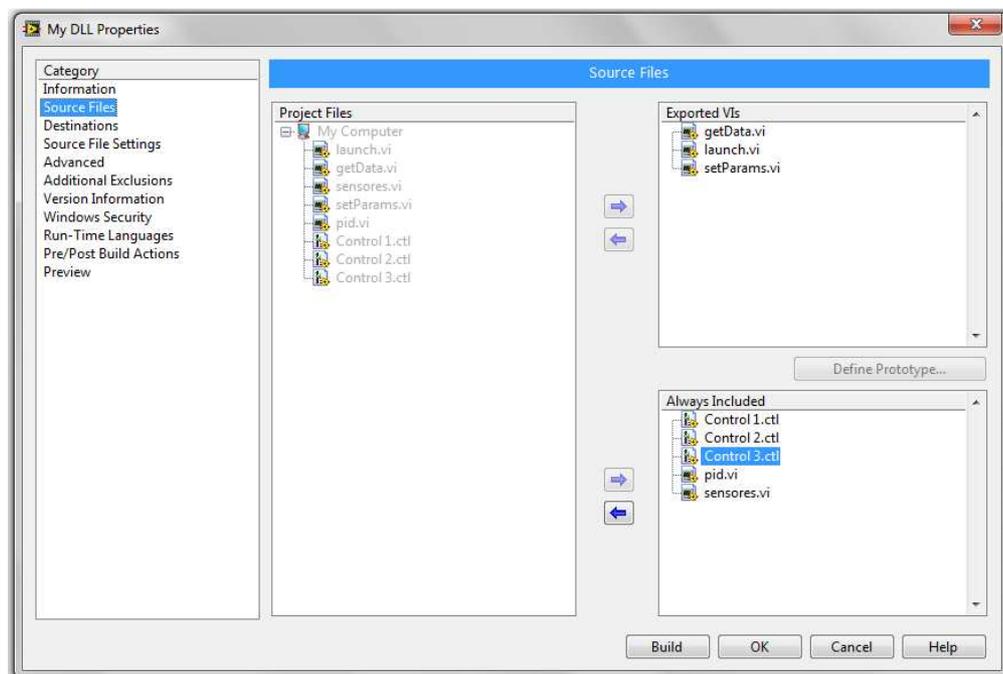


Figura 3.31 Ventana de configuración de la DLL en LabView

3.3.3 DESCRIPCIÓN DE LA INTERFAZ DEL CONTROLADOR

El controlador por realimentación de estados, que es el que hace posible el seguimiento del objeto móvil se programó en LabView 2012. Este software es llamado desde el programa ARDrone Control.NET a través de la DLL mediante la función “launch”, la cual ejecuta su acción al dar clic en un botón con el mismo nombre y que se encuentra localizado en la pantalla principal de este último programa.

La interfaz es extensa como para indicarla en un solo gráfico por lo que se la va a describir por secciones las cuales se detallan a continuación:

- **Selección de modo Automático / Manual:** el cual permite realizar el control de seguimiento o un control manual mediante pulsadores. La Figura 3.32 indica la parte de la interfaz que consta de un botón para seleccionar el modo de control y los controles manuales, cada vez que se da clic sobre el botón “automático” se activa un indicador.

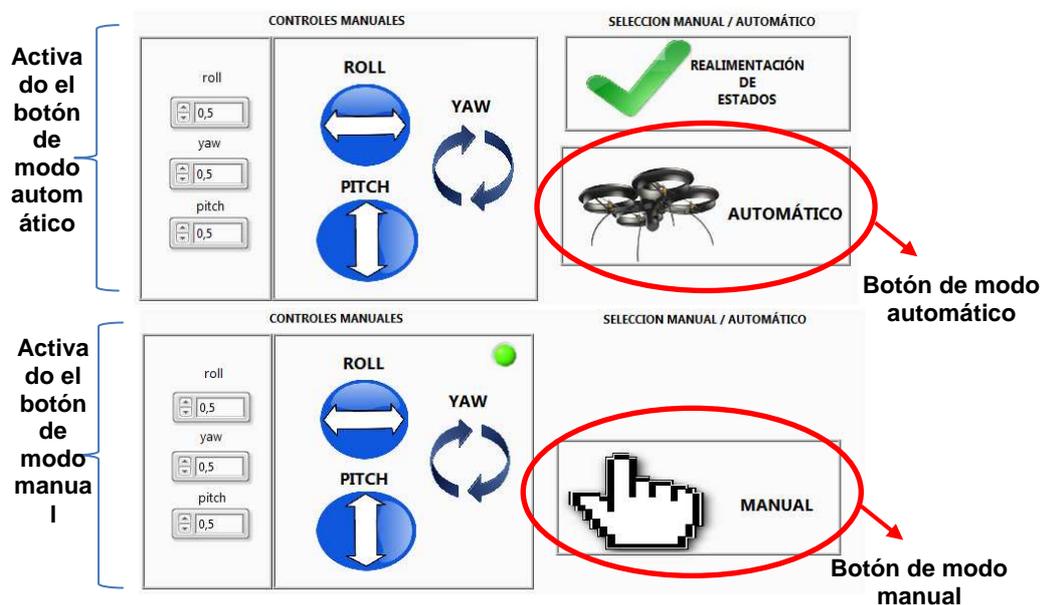


Figura 3.32 Selector de modo automático / manual.

- **On / Off del setpoint automático:** El botón de Setpoint automático permite ingresar una coordenada en mm a la cual el cuadricóptero debe moverse o

cambiar el setpoint de posición automáticamente dependiendo del movimiento del faro infrarrojo, además que consta de un indicador dependiendo del estado (Figura 3.33).



Figura 3.33 Setpoint manual / automático

- **Sección de indicadores:** En esta sección se indica cuáles sensores han detectado la presencia del faro infrarrojo mostrándose una alarma visible en caso de que el cuadricóptero esté fuera de rango, se muestra también la magnitud y dirección hacia donde éste se está moviendo y se grafica un vector en el recuadro "movimiento de cuadricóptero".

Se puede visualizar el setpoint de posición en cm que se está enviando a Roll o a Pitch y la salida en lazo abierto que corresponde a la salida del algoritmo de detección (Figura 3.34).

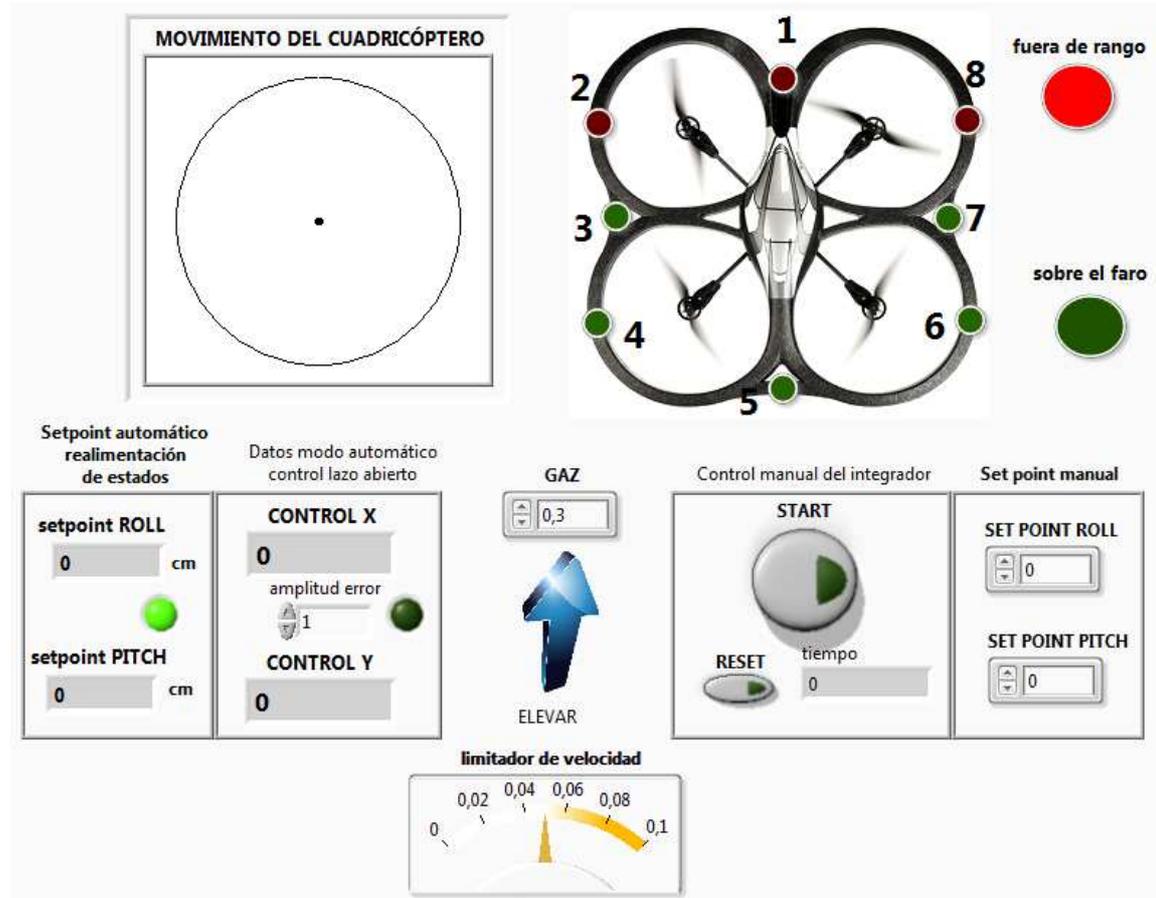


Figura 3.34 Sección de indicadores y control del integrador

- **Control del integrador:** en la Figura 3.34 se tiene dos controles para iniciar y resetear el proceso de integración de velocidad, el botón reset borra registros de velocidad anteriores y empieza la integración desde cero. Se puede ingresar los setpoints de posición manualmente, es decir se puede ingresar una coordenada en mm a la cual el cuadricóptero debe moverse. En caso de que el A.R. Drone empiece a disminuir su altura durante el vuelo se lo puede elevar con el botón del mismo nombre.

Es necesario limitar la velocidad del cuadricóptero de lo contrario se moverá demasiado rápido y será más inestable, el limitador de velocidad es una constante de 0 a 1 para limitar la salida del controlador.

- **Indicadores de los datos de navegación:** En esta sección se tiene los datos de navegación que envía el cuadricóptero vía WiFi los cuales son: las

velocidades de los tres ejes, los ángulos de Euler y la altitud como se ve en la Figura 3.35.

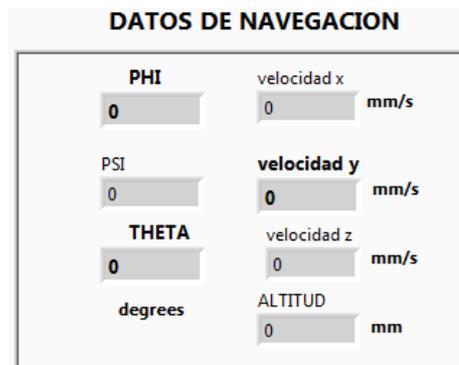


Figura 3.35 Sección de datos de navegación

- **Sección de gráficas:** Las gráficas de la interfaz permite observar el Setpoint de posición así como la posición actual del cuadricóptero, permitiendo visualizar la respuesta paso de la posición hasta llegar a la estabilidad, además de las perturbaciones que se presentan y como responde el controlador ante las mismas, además de la respuesta de velocidad y de ángulo en los dos ejes.

Además es posible configurar las ganancias de la realimentación de estados en caso de que se presente comportamientos anómalos debido al mal funcionamiento de las hélices.

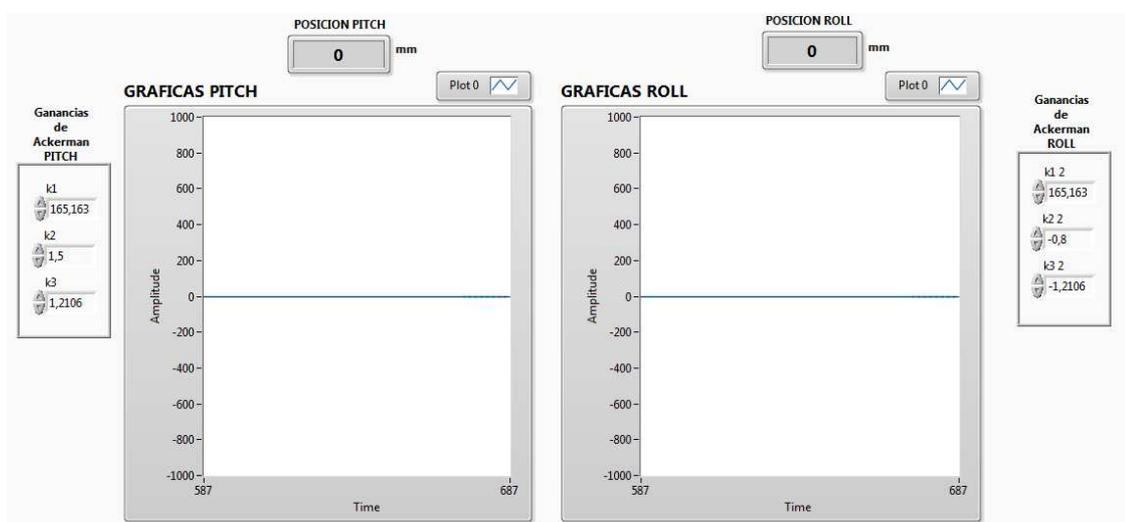


Figura 3.36 Gráficas de la salida de los controladores vs setpoint

Al llamar a la interfaz de LabView, éste empieza a ejecutarse con ciertos valores que se ponen por defecto, es decir se inicia en modo automático, con Set Point automático para realizar el seguimiento del móvil y con ganancias predefinidas para la realimentación de estados.

3.3.4 PROGRAMACIÓN EN LA INTERFAZ DE C#

C# es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, su sintaxis básica deriva de C/C++, es similar a Java, aunque incluye mejoras derivadas de otros lenguajes. Su creador Anders Heljsberg fue también el creador de muchos otros lenguajes y entornos como Turbo Pascal, Delphi o Visual J++.

La idea principal detrás del lenguaje es combinar la potencia de lenguajes como C++ con la sencillez de lenguajes como Visual Basic. La interfaz AR.DRONECONTROL.NET empleada es de licencia pública general GNU, es decir que su código fuente es de libre acceso para el público. Al disponer del código fuente se requiere realizar modificaciones puntuales para tener acceso a los datos de la DLL creada en LabView.

3.3.4.1 Programación de los movimientos de datos desde la DLL

Una vez creado el archivo DLL en LabVIEW con todas las funciones necesarias para el envío y recepción de datos, ésta debe ser asociada a los archivos de la interfaz AR.DRONECONTROL.net, para ello se lo guarda en su carpeta "Bin" la cual se crea una vez que se ha compilado todo el programa.

Las modificaciones se realizan en el programa principal del proyecto, las cuales tienen que ver con la declaración de las funciones de la DLL, éstas son:

- SetParameters []

Esta función permite enviar datos desde C# hacia la DLL, los cuales pueden ser leídos en la interfaz de LabVIEW.

- GetData []

Permite obtener datos desde la DLL, específicamente en este proyecto los comandos de movimiento roll, pitch, yaw y gaz proporcionados por los controladores en LabView.

- Launch []

Llama a la interfaz de LabView en donde se realiza todo el control para el seguimiento.

Se creó un botón cuyo evento al producirse llama a la función Launch de la DLL, en ese instante aparece una ventana con el HMI de LabVIEW, una vez abierto se tiene acceso a todas sus funcionalidades como lectura de datos del puerto serial, los algoritmos de control, gráficas y configuración de algunos parámetros importantes como son los de la comunicación y las ganancias de realimentación.

Mediante la utilización de un checkbox se habilita la lectura de los comandos de movimiento desde la DLL mediante el uso de un "THREAD", el cual es básicamente un hilo de ejecución, y puede ser ejecutado en paralelo con otra tarea. El thread se encarga de llamar a la función "movimientoDesdeDLL()" (creada en el programa principal del proyecto) la cual a su vez llama a la función "GetData()" la misma que convierte los datos leídos en un formato que pueda manejar el AR Drone, todos los datos adquiridos son indicados en pantalla para poder visualizarlos.

Los datos ya procesados deben ser enviados hacia el cuadricóptero vía WiFi y usando la función "Navigate(roll, pitch, yaw, gaz)" construida en el proyecto original del "ARDRONECONTROL.NET" se realiza esta acción. Los diagramas de flujo de las Figuras 3.37 y 3.38 resumen el funcionamiento del thread y de la función movimientoDesdeDLL().

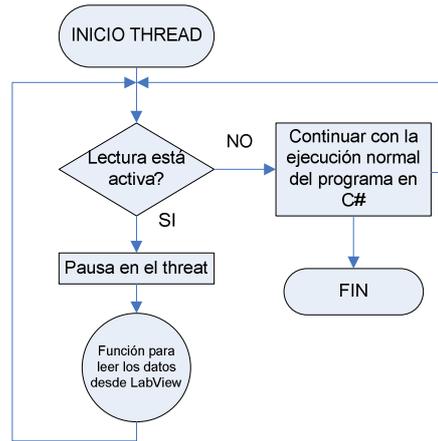


Figura 3.37 Diagrama de flujo del funcionamiento de thread

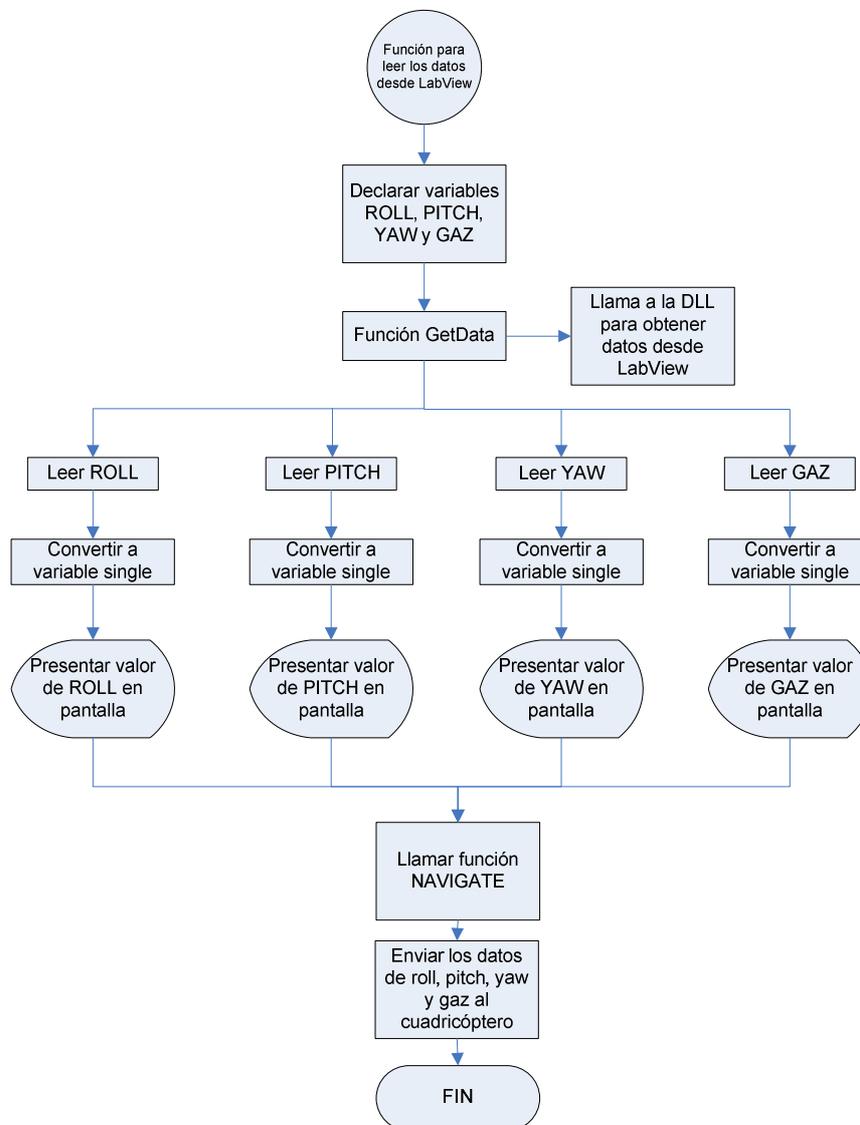


Figura 3.38 Diagrama de flujo del funcionamiento de la función movimientos desde DLL

Una vez que la instrumentación montada, junto con el ARDrone ha sido encendida y se ha establecido la comunicación WiFi con el computador se inicia el programa Ardrone Control.NET, mientras este programa se está ejecutando, mediante el botón Launch se llama a la interfaz de LabVIEW en donde es necesario comprobar que la comunicación entre los circuitos a bordo del cuadricóptero y la PC esté totalmente funcional.

Se procede a despegar y maniobrar al cuadricóptero para posicionarlo sobre el faro infrarrojo mediante el uso del teclado de acuerdo con los objetivos planteados, una vez que se lo ha posicionado se puede activar el checkbox para habilitar la recepción de los comandos de movimiento entregados por los controladores en LabView y el envío de los mismos por parte del programa escrito en C# hacia el cuadricóptero AR.Drone. Mientras el checkbox está activo únicamente se reciben los datos desde la DLL, en caso de presentarse un comportamiento anómalo se puede rápidamente desactivar este checkbox para hacer uso del teclado y controlar manualmente al cuadricóptero. El diagrama de flujo de la Figura 3.39 describe el funcionamiento global de las características añadidas al proyecto.

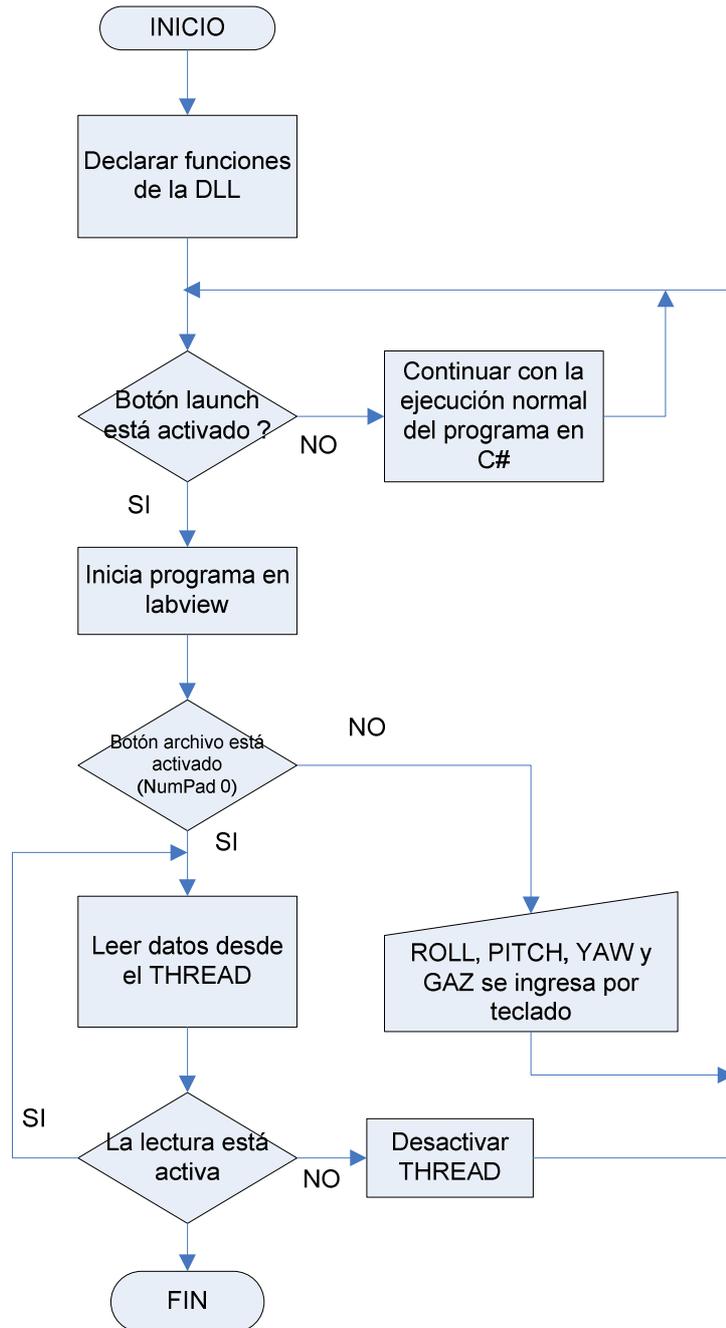


Figura 3.39 Diagrama de flujo del funcionamiento de las características añadidas al proyecto

CAPÍTULO 4

PRUEBAS Y RESULTADOS

Una vez que se ha ensamblado, diseñado, e implementado tanto el hardware como el software que conforman el conjunto de elementos necesarios para realizar el seguimiento del faro infrarrojo, se pueden realizar las pruebas necesarias para evaluar los resultados que se han obtenido al finalizar el proyecto, el mismo que consta de algunas partes que se probarán primero por separado y luego en conjunto, es decir el resultado final.

4.1 PRUEBAS DE LA PLACA DEL FARO INFRARROJO

La placa del faro infrarrojo está constituida por un conjunto de 20 emisores infrarrojos dispuestos en serie y en forma de dos círculos concéntricos tal y como se lo indicó en la etapa de diseño, y se observa en la Figura 4.1, los cuales se encuentran oscilando a una frecuencia de 38 KHz.

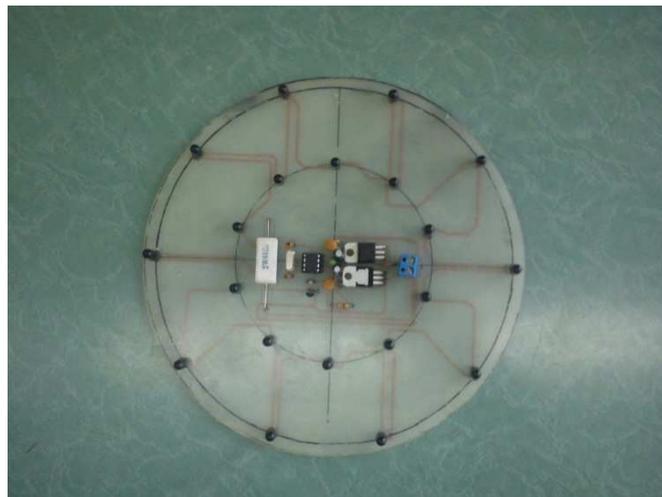


Fig. 4.1 Placa del faro infrarrojo

La placa del faro infrarrojo al energizarse forma un cono de luz infrarroja que funciona como faro para poder ser seguido por el cuadricóptero AR Drone, las

dimensiones de este cono de luz infrarroja van a cambiar dependiendo de la altura a la que se encuentre el cuadricóptero AR Drone, por lo cual hace falta saber cómo se produce este cambio de dimensiones del faro, entonces se ha diseñado un procedimiento para, dependiendo de la altura, poder saber hasta dónde es efectiva la señal infrarroja del faro.

Este procedimiento consiste en energizar la placa del faro infrarrojo para luego a ciertas alturas determinadas medir la efectividad de la señal del faro mediante uno de los sensores infrarrojos montados en el cuadricóptero AR Drone.

Tabla 4.1 Pruebas captación señal infrarroja

ALTURA	RADIO DE CAPTACIÓN EFECTIVA DE LA SEÑAL INFRARROJA
0,5 m	0,42 m
0,75 m	0,46 m
1 m	0,53 m
1,25 m	0,585 m
1,50 m	0,615 m
1,75 m	0,715 m

4.2 PRUEBAS DE LA PLACA DE LOS RECEPTORES INFRARROJOS

La placa de los receptores infrarrojos consta de un microcontrolador ATmega 164P el cual es el encargado de captar las 8 señales de los receptores y enviarlas hacia el computador mediante los módulos XBee, todo debidamente especificado en la etapa de diseño de la placa.

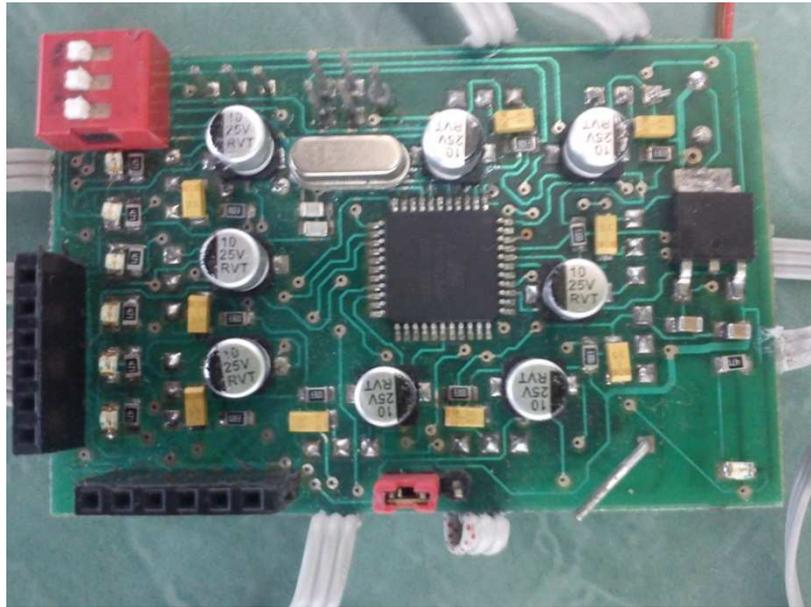


Figura 4.2 Placa de los receptores infrarrojos

Lo que se probará en este caso es qué tan bien funcionan los receptores con y sin el filtro pasabajos que se ha diseñado para mejorar la respuesta de los sensores infrarrojos. Las señales del osciloscopio de la respuesta de los sensores se presentan a continuación:

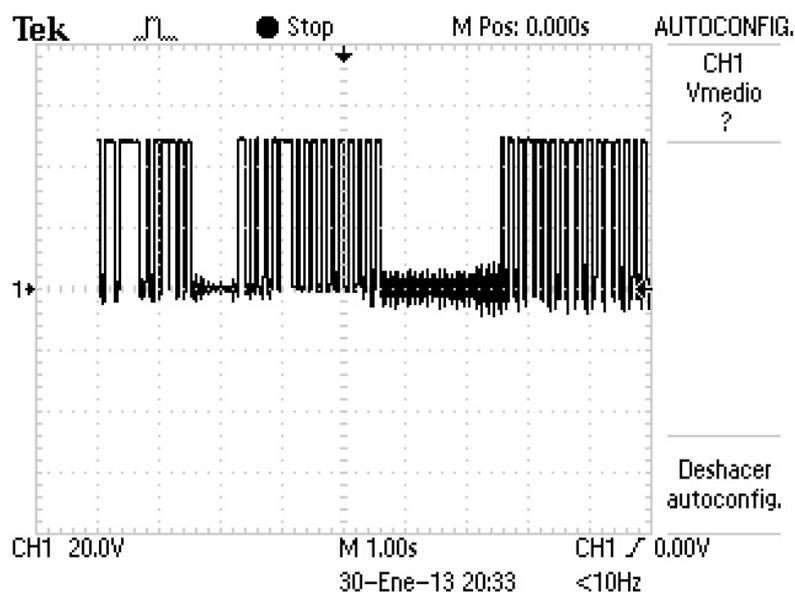


Figura 4.3 Respuesta del receptor infrarrojo en el umbral de detección sin el filtro pasabajos

Se observa en la Figura 4.3 que existe oscilación en el umbral de detección, es decir un cambio rápido entre el estado de detección y no detección, y se tiene un estado bajo cuando el sensor detecta completamente al faro infrarrojo. La

oscilación es un problema perjudicial para el algoritmo de control ya que ésta será tomada como una señal intermitente que hará que exista incertidumbre en el momento en el que el faro infrarrojo se aleje del cuadricóptero AR Drone causando que se vuelva ineficaz el algoritmo de seguimiento, es por esta razón que se implementó un filtro pasabajos para optimizar esta respuesta, el comportamiento de la señal del sensor después del filtro se lo muestra en la Figura 4.4.

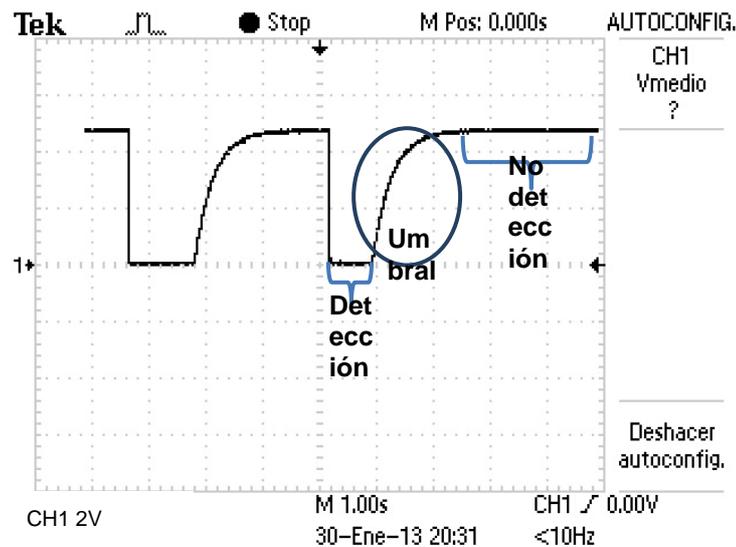


Figura 4.4 Respuesta del receptor infrarrojo en el umbral de detección con el filtro pasabajos

El filtro pasabajos es un filtro pasivo y permite el paso únicamente a frecuencias inferiores a la de diseño. Como se observa en la Figura 4.4 cuando el sensor detecta al faro se tiene 0 V, en el umbral de detección y no detección cuando se presentan las oscilaciones, las cuales tienen una frecuencia superior a la de diseño el filtro empieza a actuar y atenúa dichas variaciones y se tiene una curva de tipo exponencial propia de la carga del capacitor y finalmente cuando el receptor deja de detectar el faro, se tiene una salida de 5 V.

4.3 PRUEBAS DE VUELO DEL CUADRICÓPTERO AR DRONE ENSAMBLADO.

El cuadricóptero AR Drone de Parrot originalmente está diseñado para realizar vuelos mediante un iPod, iPhone o un dispositivo que disponga del sistema operativo Android y con conexión WiFi, el mismo ha sido modificado para que pueda ser comandado desde un computador y que cumpla las órdenes dadas por

un algoritmo para de esta forma poder realizar el seguimiento de un faro infrarrojo.

Al realizarse el ensamblaje del AR Drone, como se puede observar en la Figura 4.5, se puede notar que se tuvo que economizar mucho tanto en espacio como en peso para que el cuadricóptero pueda elevarse, cabe destacar que las placas fueron realizadas con elementos de montaje superficial para lograr reducir el tamaño de la misma para de esta manera lograr lo indicado anteriormente.



Figura 4.5 Partes constitutivas del cuadricóptero AR Drone.

El peso del cuadricóptero con las placas de instrumentación adicionales, es decir la placa de los receptores infrarrojo y la placa del módulo XBee contrastado con el peso del cuadricóptero sin estas placas se muestra en la Tabla 4.2:

Tabla 4.2 Diferencia entre los pesos del cuadricóptero original y el acondicionado.

PESO DEL CUADRICÓPTERO SIN PLACAS ADICIONALES	320 g
PESO DEL CUADRICÓPTERO CON LAS PLACAS ADICIONALES	465 g

Otra prueba que se realizó es del tiempo de vuelo durante el cual las hélices del cuadricóptero conservan su forma original sin deformarse ni pandearse, éste es un punto crítico para el vuelo del AR Drone ya que si las hélices no se encuentran en un estado óptimo el cuadricóptero no podrá elevarse, en ese estado las hélices son totalmente rectas y sin protuberancias en sus bordes.

Después de un cierto número de horas de vuelo es necesario dar un tratamiento a las hélices para ayudarlas a recuperar su forma original para lo cual se debe calentar con un secador de mano en modo caliente al máximo y el aire al mínimo durante 5 minutos cada uno de los pasos que se detallan a continuación [25].

Primero hay que dar forma a la punta de las hélices (sólo el primer centímetro de las 2 puntas de las hélices), para esto se debe utilizar un cilindro de un diámetro de pulgada y media.



Figura 4.6 Modo de enderezar la curvatura de las hélices

Luego hay que sujetar las hélices en algo plano y firme, se puede utilizar una regla y unas pinzas para fijar las hélices a la regla.



Figura 4.7 Forma de enderezar las hélices

Se debe realizar este procedimiento inclusive a las hélices nuevas ya que por el viaje de la importación se deforman debido a los paquetes que suelen poner encima.

El tiempo de vuelo óptimo acumulado de un juego de 4 hélices muestreado en 4 diferentes juegos de hélices se detalla en la Tabla 4.3.

Tabla 4.3 Tiempo de vuelo útil de las hélices

	TIEMPO ÚTIL DE VUELO ACUMULADO
JUEGO 1	1h 5 min
JUEGO 2	56 min
JUEGO 3	1h 13 min
JUEGO 4	1h 2 min

Cada juego de hélices fue nuevo, alineado y balanceado con el cual el cuadricóptero volaba de la mejor manera y luego del tiempo de vuelo que se indica en la Tabla 4.3 se presentaron anomalías en el comportamiento del mismo, de esta tabla se puede estimar un tiempo medio de vida útil de las hélices del A.R Drone.

Durante un tiempo de vida útil estimado de 1 hora 4 minutos de vuelo el AR Drone se comporta de una forma óptima en la cual se puede realizar el seguimiento del faro infrarrojo, ya que de otra forma el AR Drone tiene comportamientos anómalos que no permiten el correcto desenvolvimiento del algoritmo de control.

4.4 PRUEBAS DEL ALGORITMO DE POSICIONAMIENTO DEL AR DRONE.

El algoritmo de control usado para el posicionamiento del AR Drone es la realimentación de estados ya que se puede realimentar el ángulo de pitch tanto como el de roll para cada eje respectivamente, la velocidad de pitch y la de roll, y la posición en pitch y en roll que se obtiene de la integración de las velocidades respectivamente como se explicó en el Capítulo 3.

Se usó Matlab para encontrar las ganancias de realimentación de estados mediante el método de Ackerman, para lo cual se deben asignar los 3 polos que se desea que tenga el sistema.

Al ser la planta de cuadricóptero AR Drone un sistema de tercer orden se tienen 3 polos, pero para el diseño de una respuesta requerida se lo hará presumiendo que se trata de un sistema de segundo orden y luego se colocará el tercer polo en un lugar que haga que los polos calculados sean los dominantes.

Para poder calcular los polos dominantes deseados se tienen las condiciones de un máximo sobreimpulso M_p de 5% y un tiempo de establecimiento t_s con el criterio del 2% de 0,5 s.

$$p_{1,2} = -\xi\omega_n \pm j\omega_n\sqrt{1-\xi^2}$$

$$\xi = \pm \frac{\ln M_p}{\sqrt{\pi^2 + (\ln M_p)^2}}$$

$$\xi = \pm \frac{\ln 0,05}{\sqrt{\pi^2 + (\ln 0,05)^2}}$$

$$\xi = 0.6901$$

$$Wn = \frac{4}{\xi ts}$$

$$Wn = \frac{4}{\xi 0.5}$$

$$Wn = 11.5925$$

Con estos parámetros se tiene este punto de diseño:

$$PD = -8 \pm j8.38965$$

Estos polos dominantes conjugados hay que llevarlos al plano Z y para esto se usa un tiempo de muestreo de 67 ms, dado que éste es el tiempo que se tarda el computador en adquirir los datos de navegación desde el AR Drone.

$$z_{1,2} = e^{(PD)}$$

$$z_{1,2} = e^{(-8 \pm j8.38965)}$$

$$z_{1,2} = 0.495059 \pm j0.31183205$$

El tercer polo se lo escoge dentro del círculo unitario en el plano z en 0.85 ya que fuera de éste se pueden obtener resultados inestables.

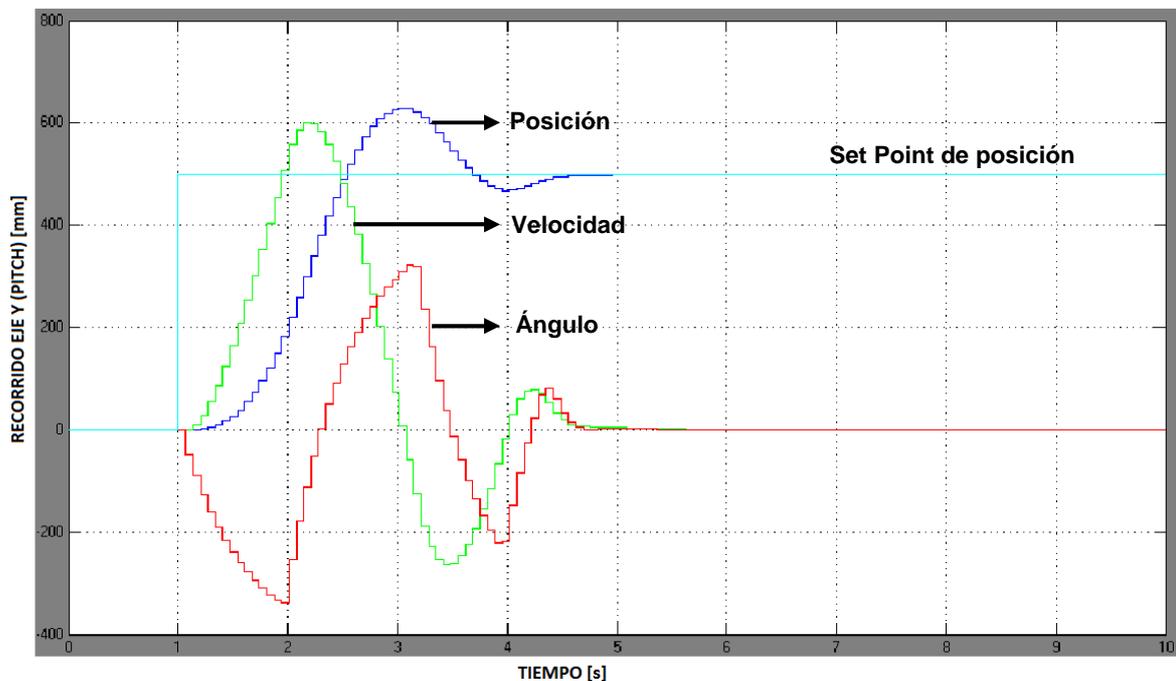
$$z_3 = 0.85$$

Usando estos polos complejos conjugados como polos dominantes para el cálculo de las ganancias de realimentación del sistema de tercer orden y como tercer polo el ubicado en 0.85, mediante el algoritmo de Ackerman disponible en Matlab para la planta de pitch se tiene:

Tabla 4.4 Valores de ganancia de realimentación calculadas para pitch

GANANCIA DE REALIMENTACIÓN	VALOR
K1	10.6188
K2	0.4266
K3	0.7825

Con estos valores de ganancias de realimentación de estados se tiene la respuesta de la Figura 4.8 a una señal paso, en la simulación realizada en Matlab para la planta de pitch.

**Figura 4.8** Respuesta simulada a una señal paso de 500mm en el eje de pitch

La señal celeste es el setpoint, la azul es la salida de posición en pitch, la verde es la velocidad en pitch y la roja es el ángulo de pitch multiplicado por una constante para que pueda ser apreciable en la gráfica.

Como se puede observar el tiempo de establecimiento está entre 3 y 4 segundos y el máximo sobreimpulso está en el orden del 20%, esto se tiene ya que los

parámetros escogidos de respuesta se ven reflejados en los polos dominantes deseados pero también existe un tercer polo que hace que la respuesta cambie un poco, además también se ve que la velocidad aumenta cuando el ángulo aumenta negativamente ya que el cuadricóptero debe cabecear hacia adelante para movilizarse en esta dirección, luego para frenarse el ángulo se hace positivo con lo que la velocidad disminuye, esto se mantiene hasta que ángulo y velocidad se vuelven cero, al llegar al setpoint.

Ahora usando los mismos 3 polos que se usaron en la planta de pitch como polos deseados en el algoritmo de Ackerman para la planta de roll, se obtienen las siguientes ganancias:

Tabla 4.5 Valores de ganancia de realimentación calculadas para roll

GANANCIA DE REALIMENTACIÓN	VALOR
K1	86.8861
K2	4.6694
K3	2.7

Con estos valores de ganancias de realimentación de estados se tiene la respuesta de la Figura 4.9 a una señal paso en la simulación realizada en Matlab para la planta de roll.

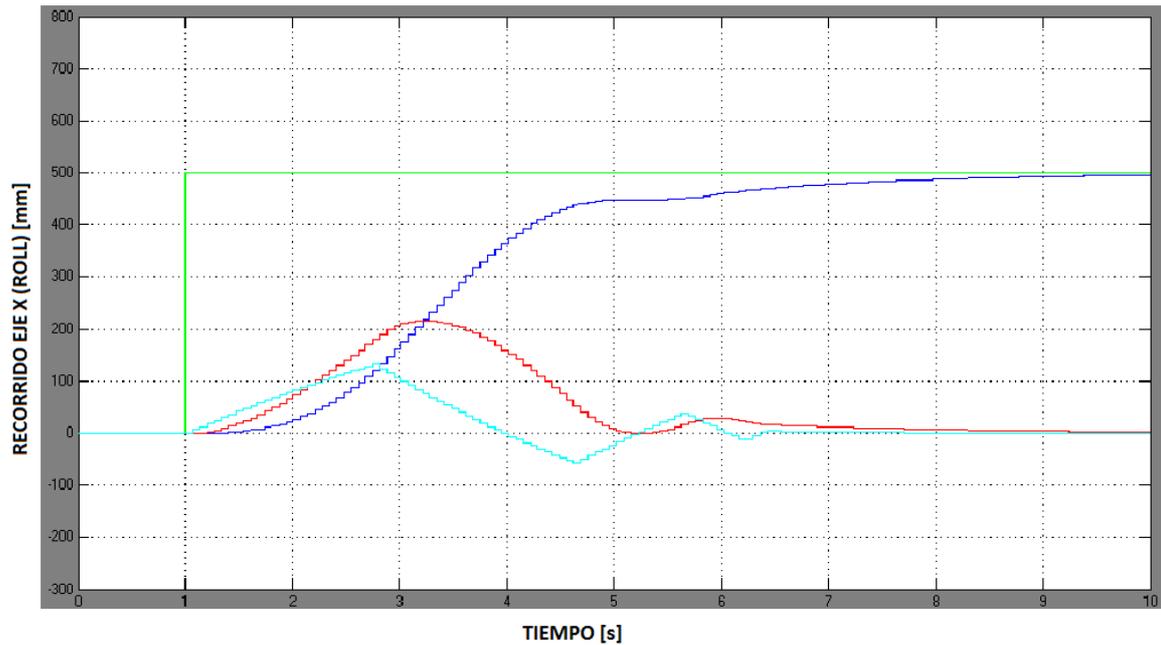


Figura 4.9 Respuesta simulada a una señal paso de 500 mm en el eje roll

La señal verde es el setpoint, la azul es la salida de posición en roll, la roja es la velocidad en roll y la celeste es el ángulo de roll. El tiempo de establecimiento en este caso está en los 7 segundos y no existe sobre pico, esto se debe a que los 2 polos escogidos son los dominantes pero no los únicos, se observa claramente que cuando el ángulo se incrementa la velocidad también lo hace dando como consecuencia un movimiento positivo en el eje roll hasta que se logre alcanzar al setpoint, en este caso también el ángulo se hace negativo con el fin de frenar y compensar la velocidad para que no exista sobrepico.

Implementando los valores de ganancias de realimentación que se calcularon con el algoritmo de Ackerman en Matlab para cada planta, en pitch y roll respectivamente, pero ahora en el algoritmo implementado realmente en LabVIEW para comandar al cuadricóptero se obtienen las respuestas mostradas en las Figuras 4.10 y 4.11 a una señal paso de 500 mm pitch y roll respectivamente.

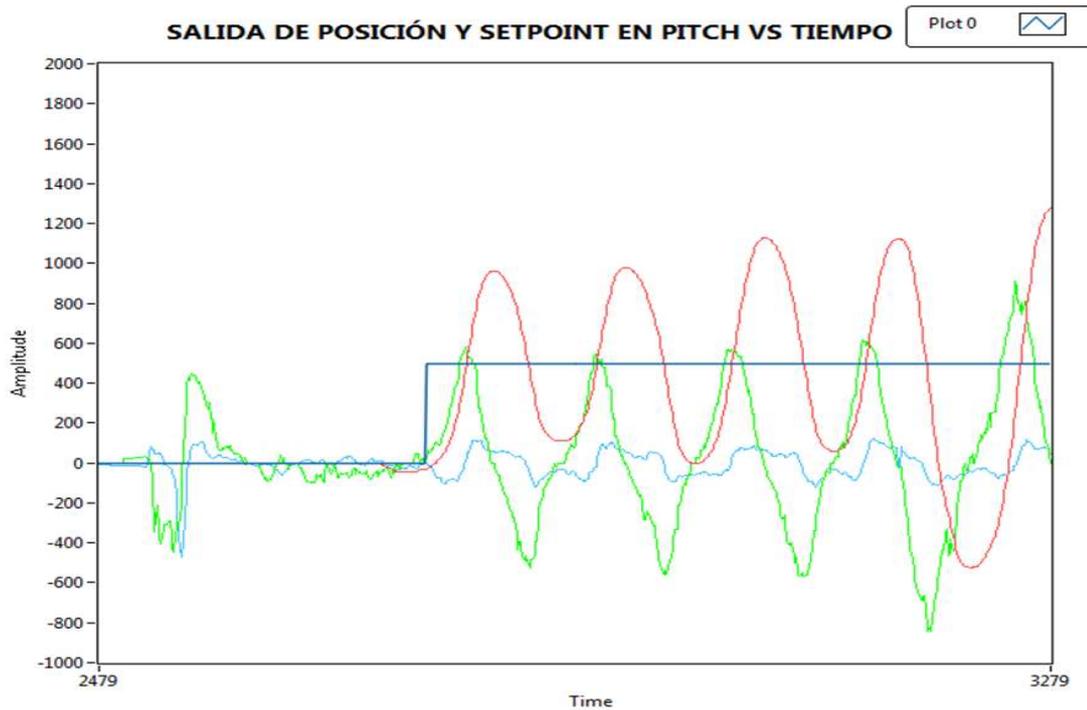


Figura 4.10 Respuesta de posición del cuadricóptero en el eje Y (pitch) ante una entrada paso de 500 mm

La señal azul representa el setpoint, la señal roja la salida de posición en el eje Y (pitch), la verde es la velocidad en el eje Y y la celeste la velocidad en el eje X (roll), se puede observar que la respuesta del sistema es oscilante e inestable con las ganancias de realimentación calculadas por el algoritmo de Ackerman en Matlab, por lo cual va a ser necesario usar el método de prueba y error para encontrar las ganancias adecuadas con las que pueda funcionar de una manera adecuada el posicionamiento del cuadricóptero.

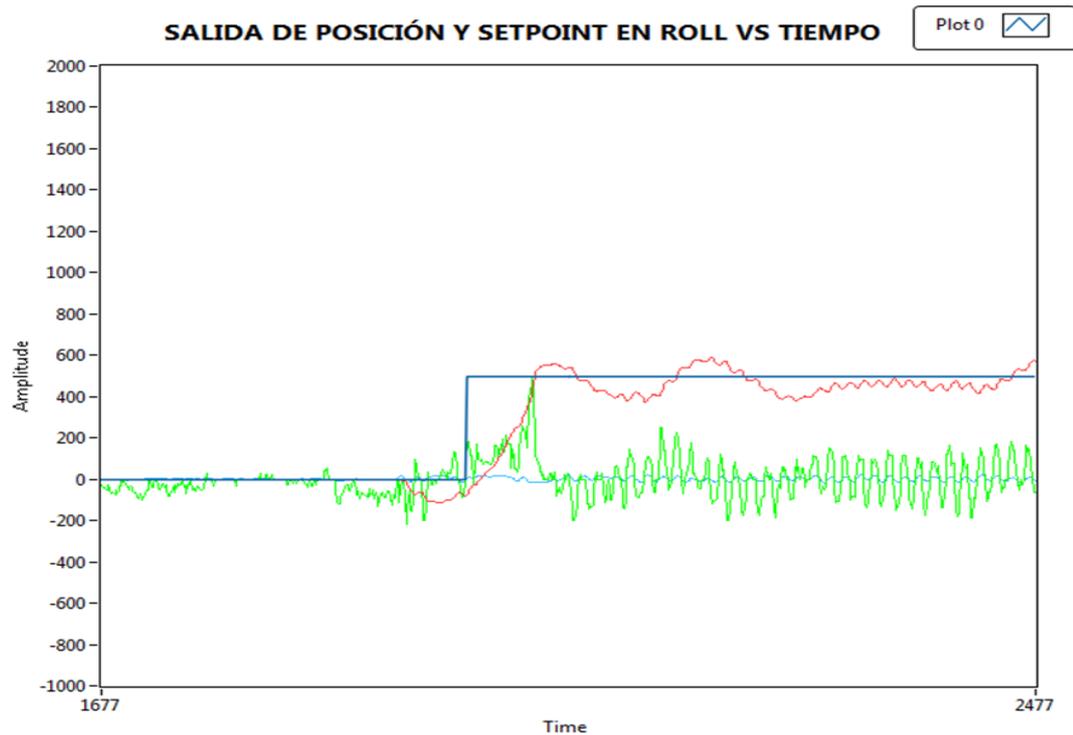


Figura 4.11 Respuesta de posición del cuadricóptero en el eje X (roll) ante una entrada paso de 500 mm

Al igual que en la gráfica anterior la señal azul es el setpoint, la roja es la salida de posición en roll, la verde la velocidad del cuadricóptero en roll y la celeste la velocidad en pitch, esta respuesta es mucho más estable que la respuesta en Pitch pero aun así debe ser mejorada con el mismo método que se usará para la planta de pitch, el de prueba y error.

De la experimentación, prueba y error con diferentes valores de ganancias de realimentación de estados en el algoritmo implementado en el cuadricóptero se puede concluir que al aumentar K_1 el error de posición disminuye, al aumentar K_2 la estabilidad aumenta y el tiempo de establecimiento también aumenta, y al aumentar K_3 el tiempo de establecimiento disminuye pero la estabilidad también disminuye, por lo que se han realizado las pruebas necesarias para encontrar las mejores ganancias experimentalmente para poder obtener un buen desempeño del cuadricóptero AR Drone.

Entonces partiendo con la primicia de que las planta de Roll y Pitch son muy parecidas, en base a las ganancias de Roll se encontraron las ganancias de Pitch

y posteriormente se los han ajustado para que el control sea más estable, con lo que se han escogido ciertos valores de ganancias de realimentación que se muestran en la Tabla 4.6, partiendo como ya se dijo de la experimentación del mejor comportamiento del cuadricóptero Parrot AR Drone y el comportamiento de ambas plantas con dichas ganancias son muy aceptables.

Tabla 4.6 Valores de ganancias de realimentación experimentales para la plantas de pitch y roll

GANANCIA DE REALIMENTACIÓN	Valor en Pitch	Valor en Roll
K1	165.163	165.163
K2	3.5	-3.5
K3	1.5	-1.5

Con estas ganancias de realimentación implementados en la simulación realizada en Matlab se tiene las siguientes respuestas a una señal paso de 500 mm para pitch y roll respectivamente.

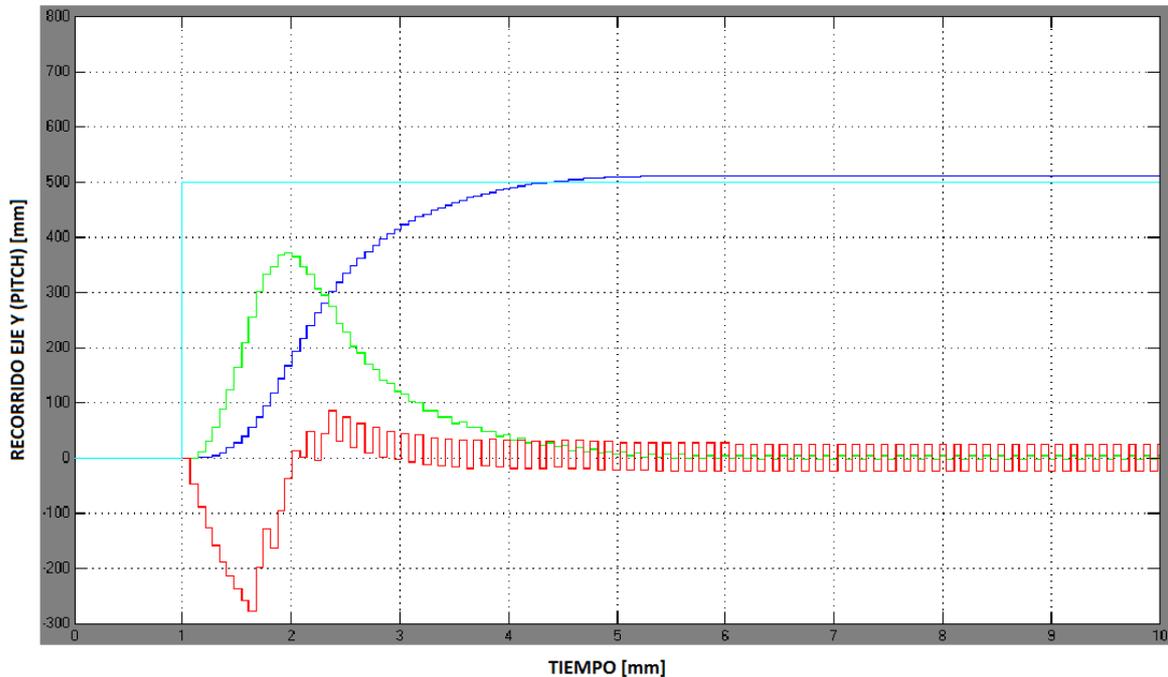


Figura 4.12 Respuesta simulada a una señal paso de 500mm de la planta de pitch

La señal celeste representa el setpoint, la azul es la salida de posición en el eje de pitch, la verde es la velocidad en pitch y la roja el ángulo de pitch multiplicado por una constante para que sea apreciable en la gráfica.

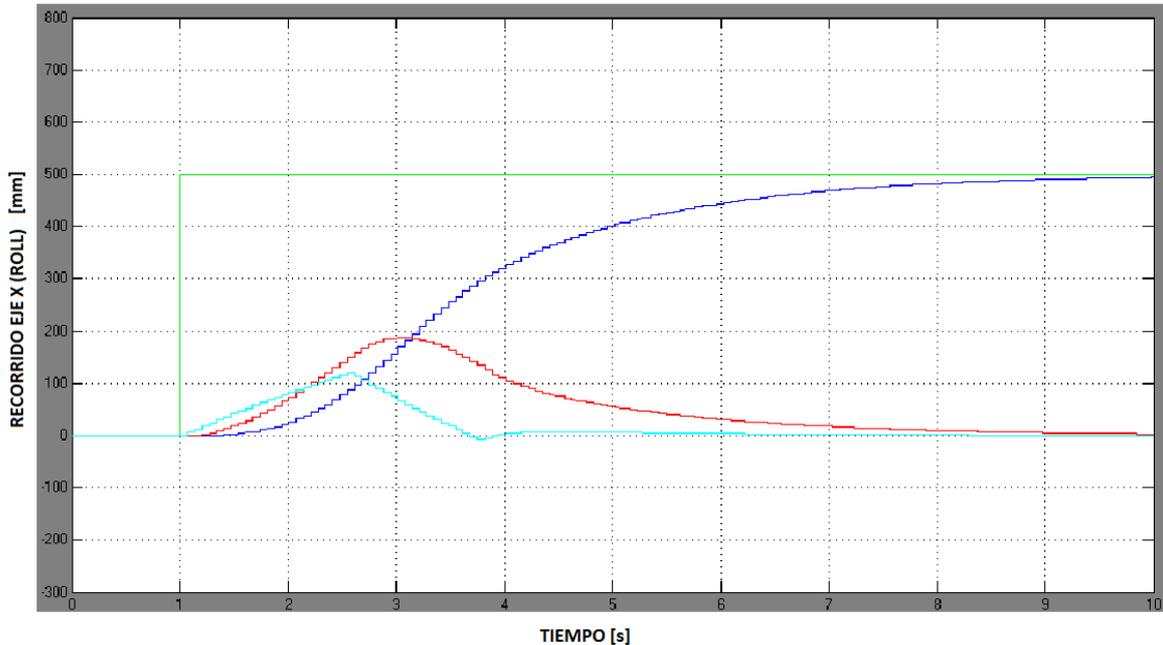


Figura 4.13 Respuesta simulada a una señal paso de 500 mm de la planta roll

En este caso la señal verde es el setpoint, la azul la salida de posición en el eje roll, la roja es la velocidad en roll y la celeste es el ángulo de roll.

La respuesta tanto de pitch como de roll son satisfactorias, aunque el tiempo de establecimiento se nota que es algo prolongado en la simulación, de todos modos se usarán éstos valores de ganancias de realimentación experimentales.

Con estos mismos valores de ganancias de realimentación experimentales se realiza una prueba en el cuadricóptero real insertándolos en el algoritmo implementado en el computador, dando como resultado las respuestas de las Figuras 4.14 y 4.15 a una entrada paso de 500 mm.

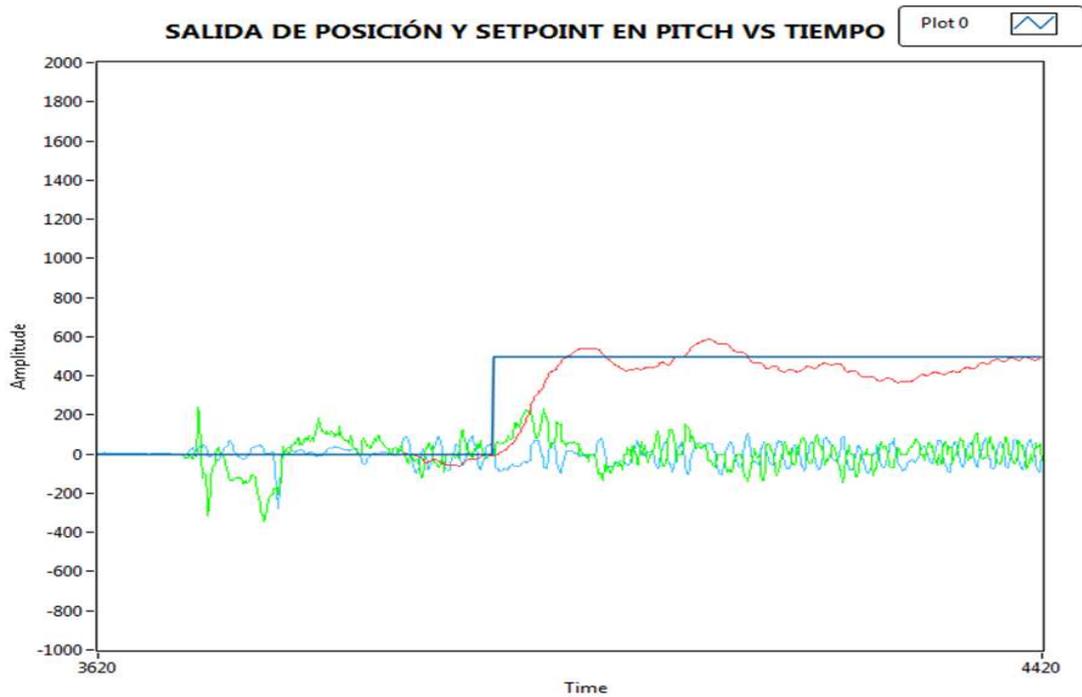


Figura 4.14 Respuesta en el eje Y (pitch) del cuadricóptero con las ganancias de realimentación experimentales ante una entrada paso de 500 mm.

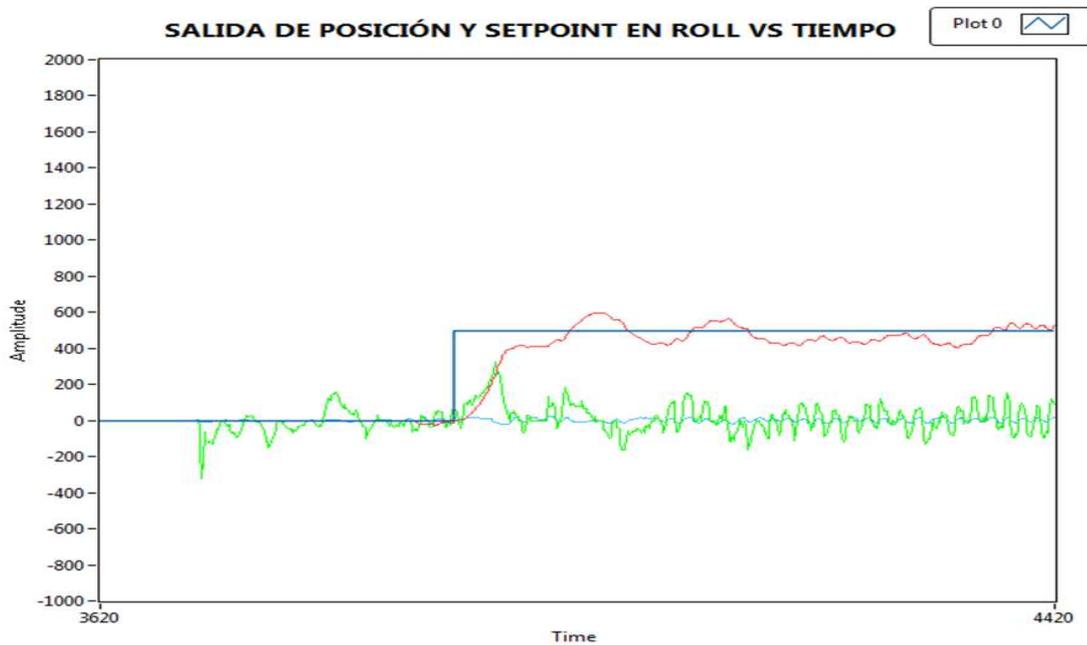


Figura 4.15 Respuesta en el eje X (roll) del cuadricóptero con las ganancias de realimentación experimentales ante una entrada paso de 500 mm.

Como se observa en las figuras la respuesta de las plantas de pitch y roll del cuadricóptero son aceptables para el posicionamiento sobre una posición dada, es por esto que para el seguimiento del faro infrarrojo se usarán las ganancias de realimentación que se obtuvieron de la experimentación, es decir se usarán las ganancias de la Tabla 4.6.

4.4.1 PRUEBAS DE LA INCERTIDUMBRE EN EL POSICIONAMIENTO DEL CUADRICÓPTERO AR DRONE

Se ha diseñado esta prueba para poder medir la fiabilidad de los movimientos del AR Drone al enviarle un set point de posición en el plano XY ya que según el algoritmo de control y su visualización en el computador el cuadricóptero alcanza el setpoint, pero lo que no se sabe es que tan preciso fue este logro contrastándolo con la realidad, es decir si en el algoritmo de posicionamiento se envía un setpoint de 1m en X y 1m en Y, se debe saber si en la realidad el cuadricóptero se movió esa distancia de lo contrario conocer cuál es el porcentaje de error en ese movimiento. Para realizar todas estas pruebas se usaron las ganancias de la Tabla 4.5 ya que con ellas se lograron los mejores resultados.

4.4.1.1 Pruebas en el eje Y (pitch).

Primero se pondrá a prueba el posicionamiento del cuadricóptero en el eje Y para esto se solicitará al cuadricóptero un set point de 0.5m en el eje Y (pitch).

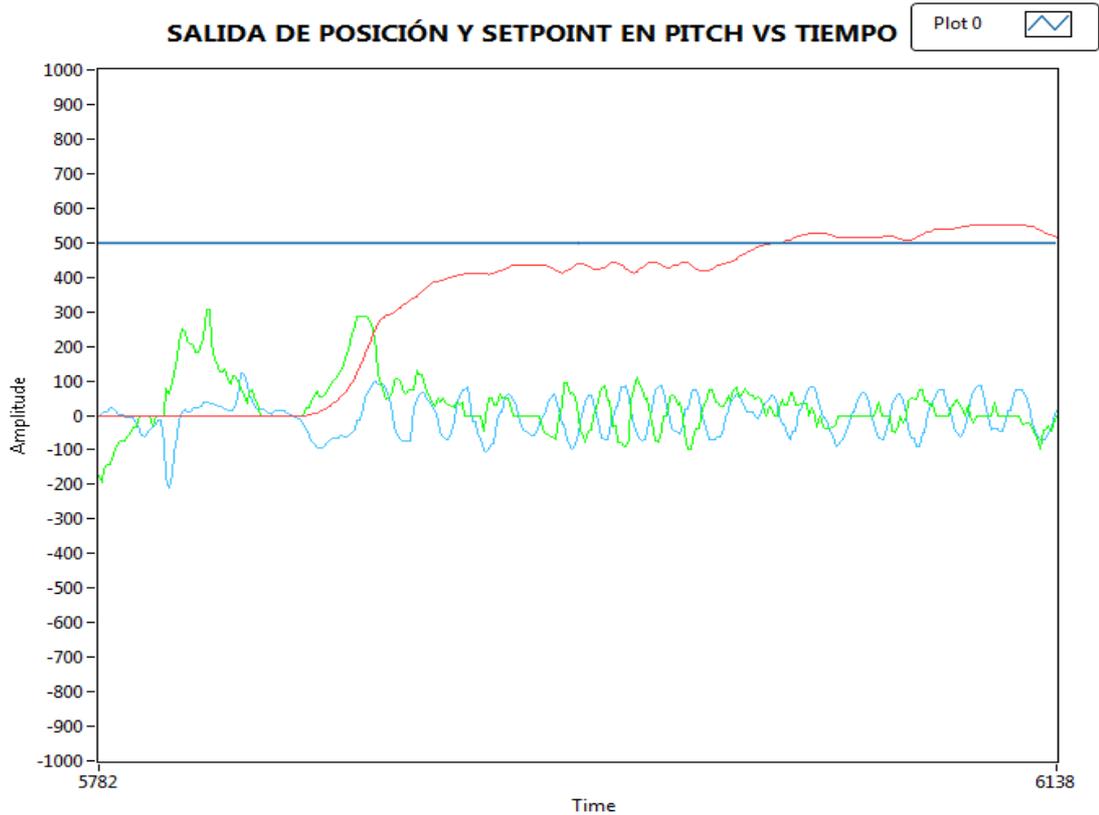


Figura 4.16 Respuesta del cuadricóptero ante una entrada paso de 500 mm en el eje de pitch

La señal azul representa el setpoint, la roja la posición en Y, la verde la velocidad en pitch y la celeste la velocidad en roll. Se puede observar que el tiempo de establecimiento es de 3.12 s y su sobrepico es del 16 %, valores aceptables como resultado lo que indica que los valores escogidos para las ganancias de realimentación son los apropiados para la planta de pitch, además la distancia real medida que recorrió el cuadricóptero en el eje Y (pitch) fue de 633 mm, dando un error de posicionamiento del 26.6 %.

Luego se le solicitó al cuadricóptero un setpoint de 2m en el eje Y (pitch).

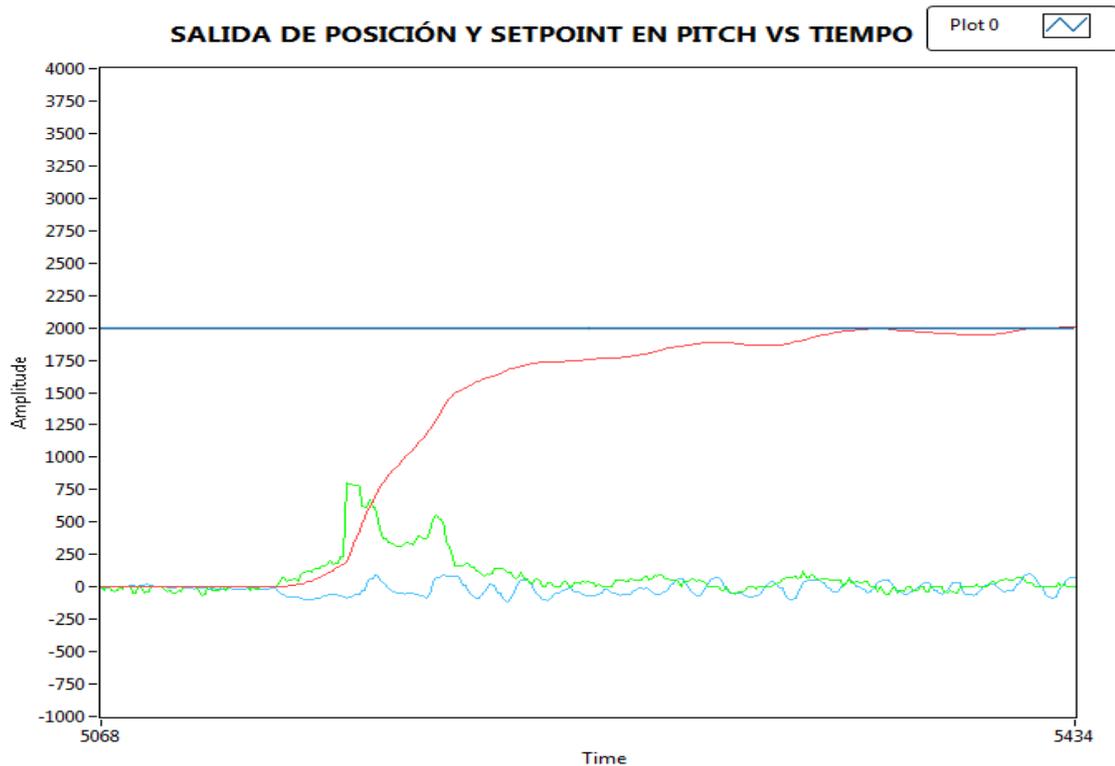


Figura 4.17 Respuesta del cuadricóptero ante una entrada paso de 2000 mm en el eje Y (pitch)

La señal azul representa el setpoint, la señal roja el recorrido en pitch, la señal verde la velocidad en pitch y la señal celeste corresponde a la velocidad en roll. En este caso el tiempo de establecimiento es de 3.16 s y no existe sobrepico, estos valores de respuesta a una entrada de 2000 mm son muy aceptables para la planta de pitch, el recorrido real del cuadricóptero en el eje Y (pitch) fue de 2300 mm, dando un error de posicionamiento del 15%.

Y finalmente se le solicitará al cuadricóptero un setpoint de 5m en el Y (pitch).



Figura 4.18 Respuesta del cuadricóptero ante una entrada paso de 5000 mm en el eje Y (pitch)

La señal azul representa el setpoint, la señal roja el recorrido en pitch, la señal verde la velocidad en pitch y la señal celeste corresponde a la velocidad en roll. El tiempo de establecimiento para este caso es de 5.33 s y existe un sobrepico del orden del 5%, estos valores de respuesta a una entrada de 5000 mm son aceptables para la planta de pitch, el recorrido real del cuadricóptero en el eje Y (pitch) fue de 5525 mm, dando un error de posicionamiento del 10.5%.

4.4.1.2 Pruebas en el eje X (roll).

Para las gráficas siguientes de respuesta ante una señal paso, la señal azul representa el setpoint, la señal roja la posición en el eje X (roll), la señal verde la velocidad en el eje X (roll) y la señal celeste la velocidad en el eje Y (pitch).

Se probará el posicionamiento del cuadricóptero en el eje X (roll), para esto se solicita un setpoint de 0.5m en el eje X (roll).

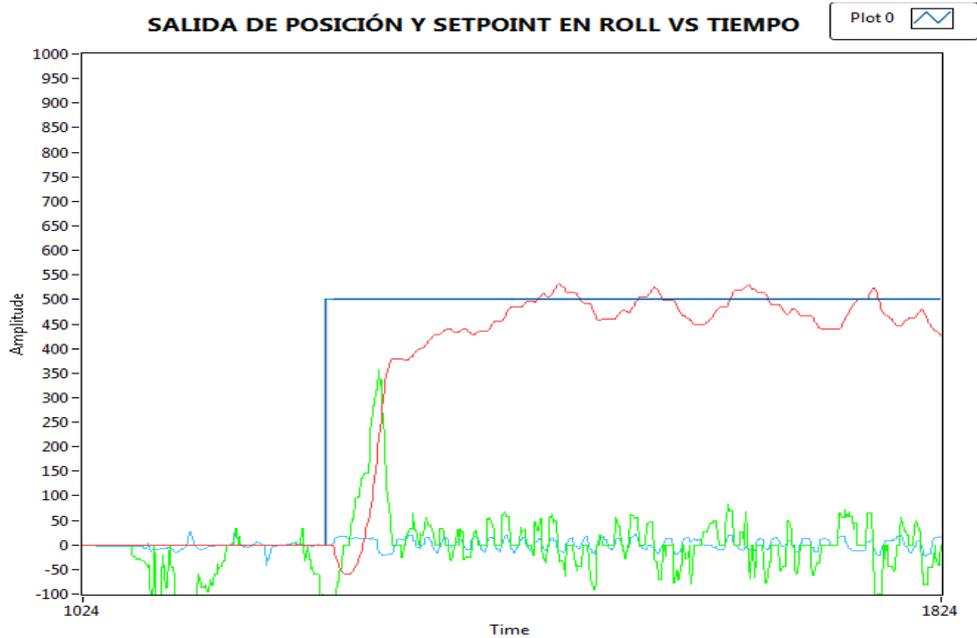


Figura 4.19 Respuesta en el eje X (roll) ante una señal paso de 500 mm

El tiempo de establecimiento para este caso es de 2.9 s, existe un pequeño sobrepico del orden del 5%, también se puede observar cierta oscilación al llegar al setpoint debido a factores externos como lo es una brisa leve, el recorrido real del cuadricóptero hasta el setpoint fue de 595 mm, es decir se tuvo un error del 19%.

Luego se le solicitará al cuadricóptero un set point de 2m en el eje X (roll).

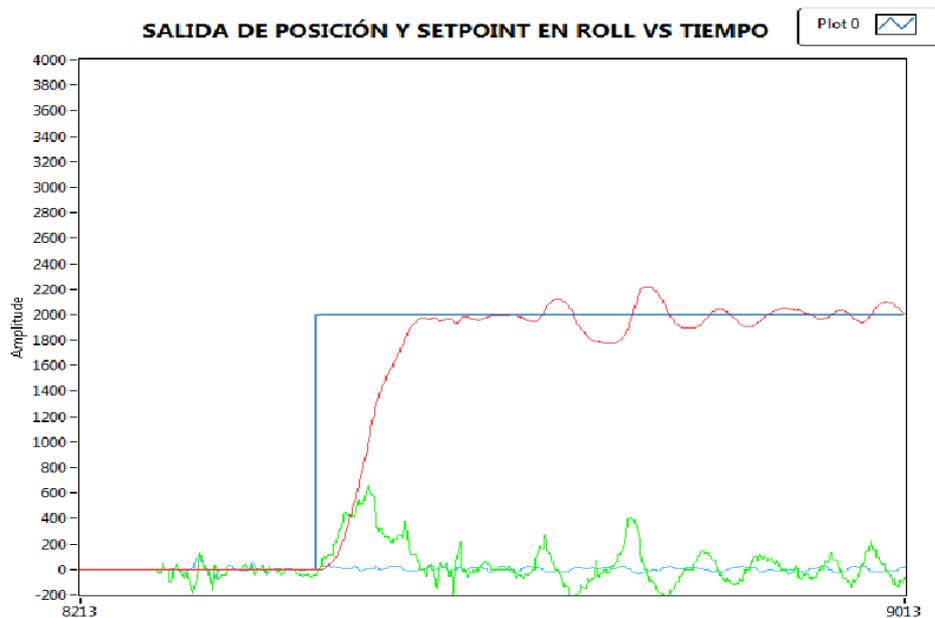


Figura 4.20 Respuesta en el eje X (roll) ante una señal paso de 2000 mm

Aquí el tiempo de establecimiento es de 2.1 s, sin sobrepico, lo que se observa luego es la respuesta de la acción compensadora ante una muy leve brisa, el recorrido real en el eje X (roll) del cuadricóptero hasta alcanzar el setpoint fue de 2100 mm teniéndose un error de posicionamiento del 5%.

También se le solicitará al cuadricóptero un setpoint de 5m en el eje X (roll).

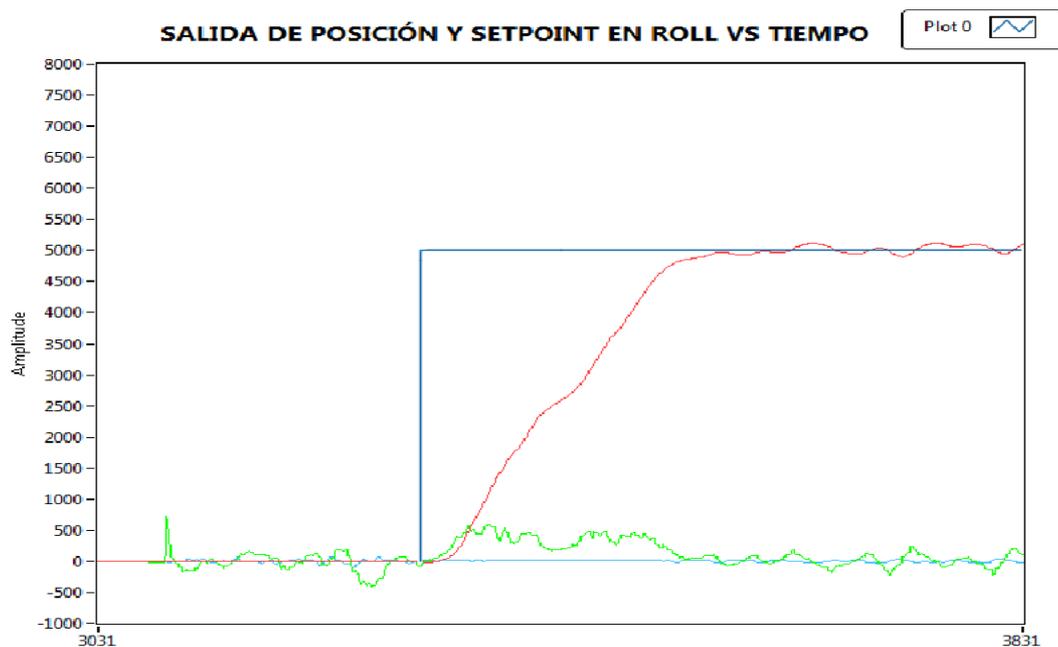


Figura 4.21 Respuesta en el eje X (roll) ante una señal paso de 5000 mm

El tiempo de establecimiento de la señal de posición en este caso es de 5.2 segundos, existe un sobrepico del orden del 2% y la respuesta compensadora es muy buena, el recorrido real del cuadricóptero en el eje X (roll) hasta alcanzar el setpoint de 5000 mm fue de 6980 mm, dando un error del 39.6%.

4.4.1.3 Pruebas simultáneas en el eje X (roll) y Y (pitch).

Una vez que se han realizado pruebas por separado en cada eje X e Y (roll y pitch) se van hacer pruebas simultáneamente en ambos ejes.

Primero se solicita un setpoint de 0.5m en X (roll) y 0.5m en Y (pitch)

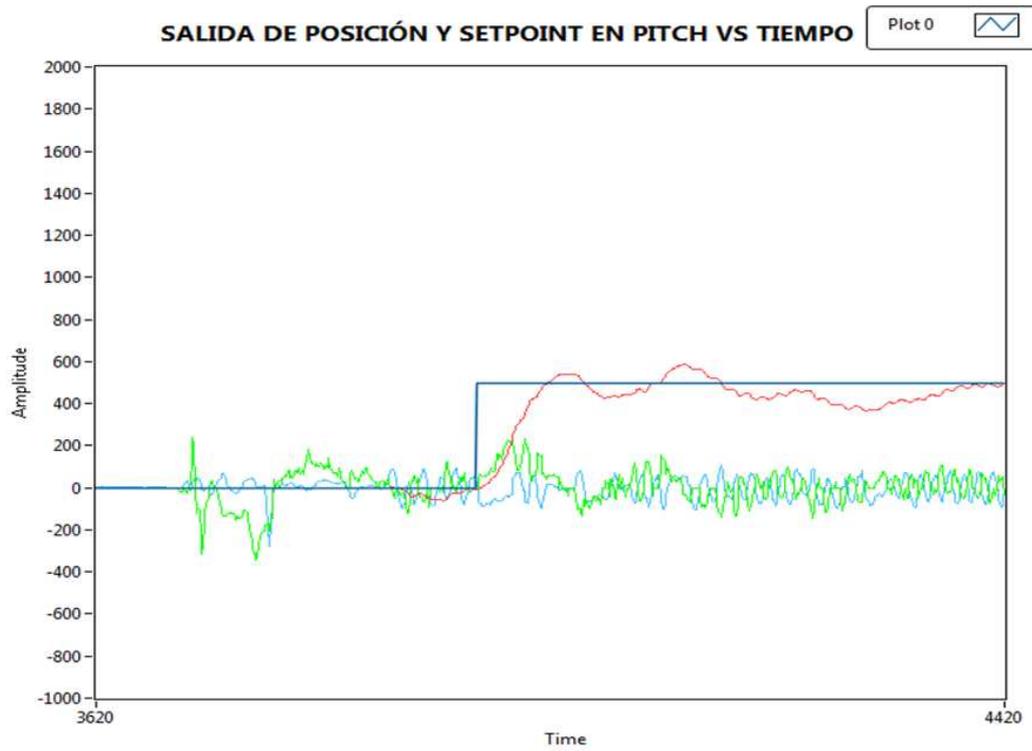


Figura 4.22 Respuesta en el eje Y (pitch) ante una señal paso de 500 mm

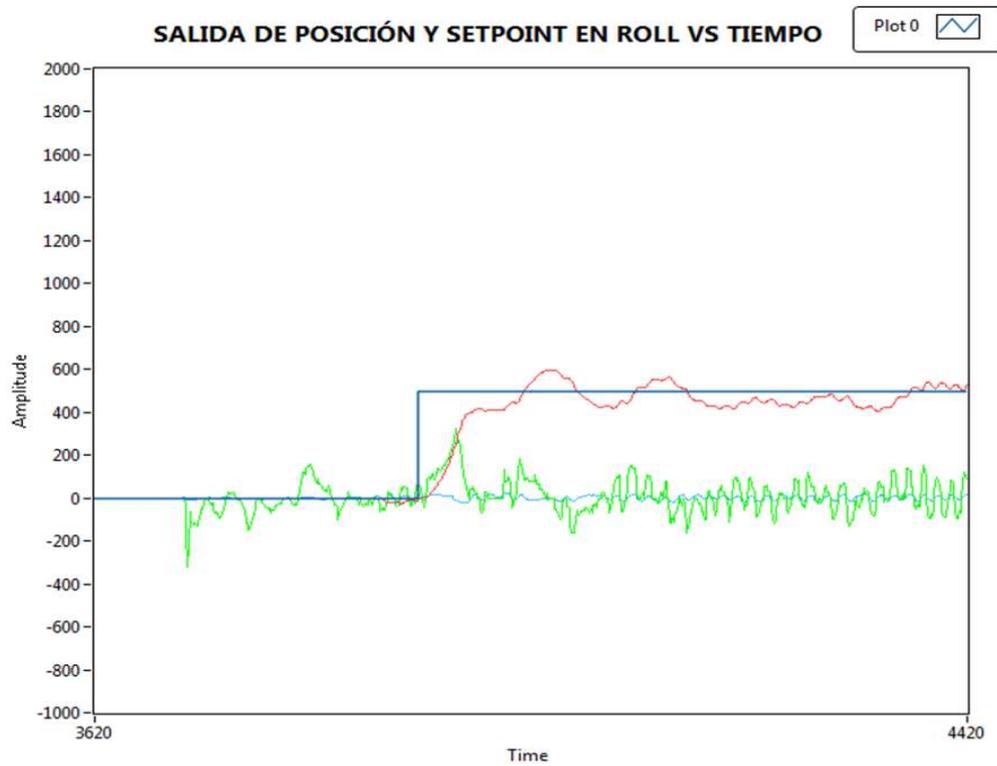


Figura 4.23 Respuesta en el eje X (roll) ante una señal paso de 500 mm

Las respuestas (señal roja) tanto de pitch como de roll al setpoint (señal azul) de 500 mm son muy aceptables, y suficientes para funcionar en el seguimiento del faro infrarrojo, el comportamiento de la velocidad se puede observar en la señal verde y el del ángulo en la señal celeste el mismo que varía constantemente con el afán de realizar la acción compensadora, en el eje de pitch el cuadricóptero se movió realmente 530 mm y en el eje de roll 590 mm hasta llegar al setpoint, es decir en el eje de pitch se tuvo un error del 6% y en el eje de roll un error del 18%.

Luego se le solicitará al cuadricóptero un set point de 2m en el eje X (roll) y 2 m en el eje Y (pitch).

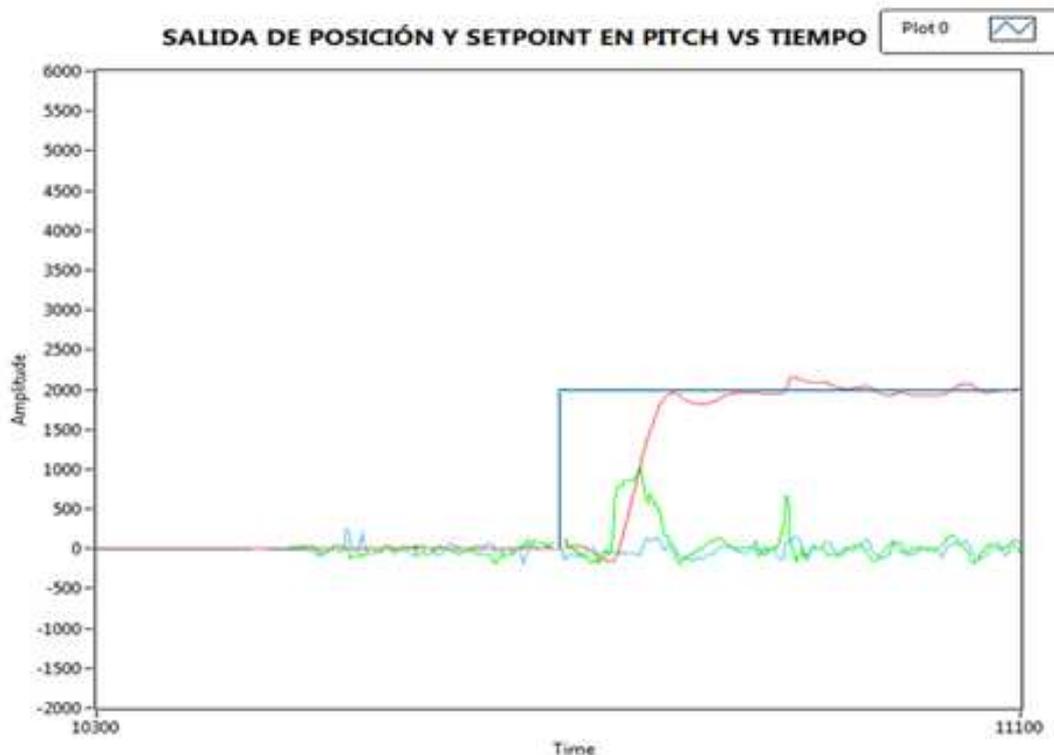


Figura 4.24 Respuesta en el eje Y (pitch) ante una señal paso de 2000 mm



Figura 4.25 Respuesta en el eje X (roll) ante una señal paso de 2000 mm

En este caso el convenio de colores en las señales es el mismo que en las 2 gráficas anteriores, las respuestas son un poco menos oscilantes, por lo tanto más satisfactorias que en el último caso, también se puede observar el comportamiento de la velocidad y del ángulo en las compensaciones que deben hacerse para llegar al setpoint deseado.

La posición final en el eje de pitch fue de 2370 mm y en el eje de roll fue de 2510, es decir, en el eje de pitch se tuvo un error de 18.5% y en el eje de roll un error del 25.5%.

En la Tabla 4.7 se compararán los resultados en las pruebas de incertidumbre en la posición para los diferentes casos.

Tabla 4.7 Cuadro resumen de pruebas de incertidumbre de posición.

TIPO DE PRUEBA	DISTANCIA [mm]	Error en Pitch [%]	Error en Roll [%]
NO SIMULTÁNEA	500	26,6	19
	2000	15	5
	5000	10,5	39,6
SIMULTÁNEA	500	6	18
	2000	18,5	25,5

4.5 PRUEBAS DEL ALGORITMO DE SEGUIMIENTO IMPLEMENTADO

Por último como prueba final se realizará el seguimiento del faro, utilizando el algoritmo de posicionamiento modificado para que el integrador se resetee cada vez que cambie el setpoint según lo explicado en el Capítulo 3, en las gráficas siguientes tanto de pitch como de roll respectivamente, se puede observar el comportamiento y cambios del set point en la señal azul, el comportamiento de la salida de posición en la señal roja, la velocidad de cada eje es la señal verde en cada gráfica y la señal celeste representa el ángulo en cada caso, observando las gráficas se puede decir que la salida tiende a seguir al setpoint en cada eje y como el vuelo se realiza a una altura relativamente constante, el eje Z no se toma en cuenta, es decir la tercera dimensión no se usa en el algoritmo ya que es controlado manualmente, estas gráficas se traducen en el seguimiento del faro infrarrojo, tal y como se lo planteó en los alcances de este proyecto de titulación.

El setpoint siempre tiende a cero ya que cuando el setpoint es cero quiere decir que el cuadricóptero debe dejar de moverse porque se encuentra dentro del faro infrarrojo, cuando el setpoint es un valor diferente de cero quiere decir que el cuadricóptero debe moverse esa distancia para que siga dentro del faro infrarrojo.

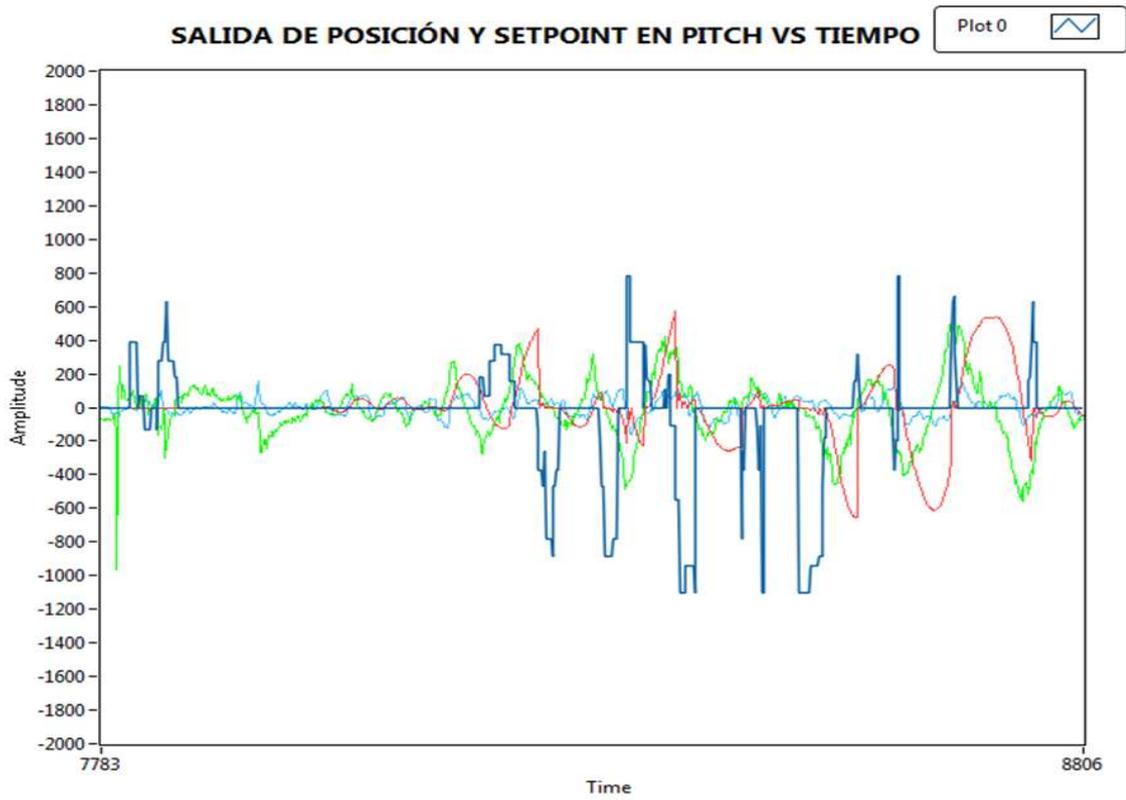


Figura 4.26 Prueba de seguimiento al faro infrarrojo en el eje Y (pitch)

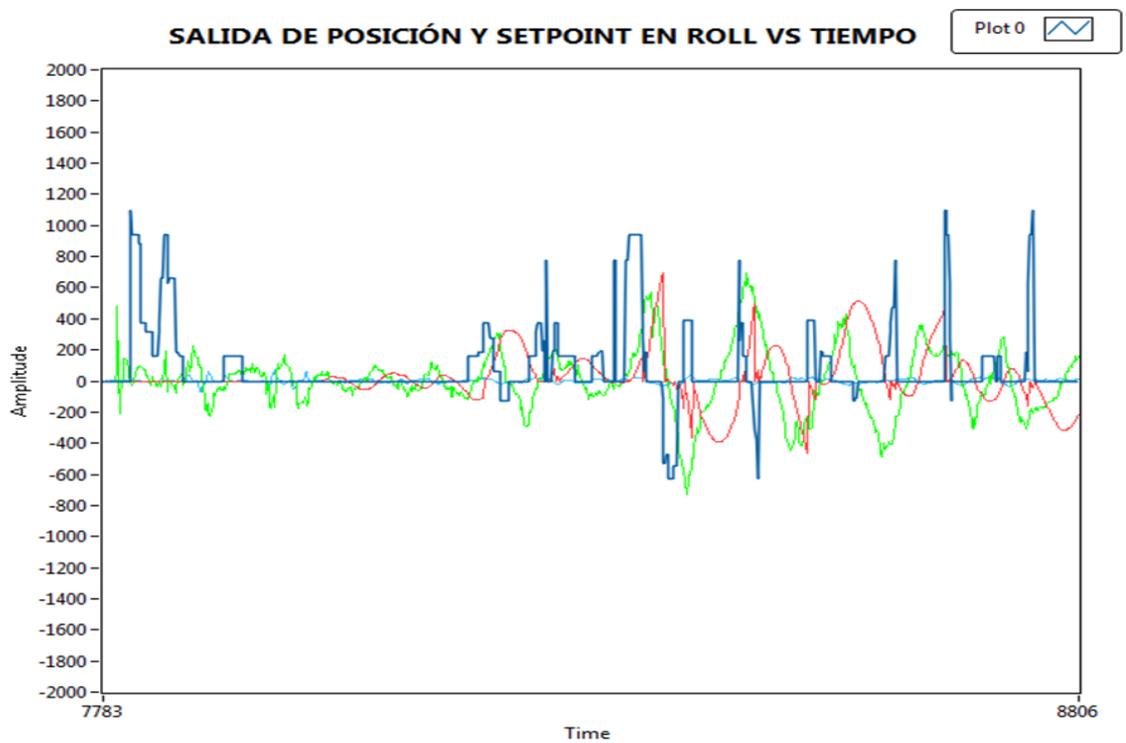


Figura 4.27 Prueba de seguimiento al faro infrarrojo en el eje X (roll)

4.6 LISTA DE ELEMENTOS Y COSTOS

El costo del proyecto de titulación se lo detalla a continuación.

Tabla 4.8 Costos de elementos varios

ELEMENTOS VARIOS			
DESCRIPCIÓN	CANTIDAD	PRECIO UNITARIO	PRECIO TOTAL
CUADRICOPTERO AR DRONE*	1	\$ 575,00	\$ 575,00
JUEGO DE 4 HÉLICES	3	\$ 20,00	\$ 60,00
BATERÍAS LiPo 2000 mAh	4	\$ 35,00	\$ 140,00
CABLE SERIAL USB	1	\$ 25,00	\$ 25,00
CAUTÍN DE PRECISIÓN	1	\$ 20,00	\$ 20,00
ROLLO DE ESTAÑO PARA SOLDADURA SMD RADIOSHACK	1	\$ 20,00	\$ 20,00
REPUESTO CRUZ ESTRUCTURA FIBRA DE CARBONO	1	\$ 40,00	\$ 40,00
REPUESTO MOTOR CUADRICÓPTERO	1	\$ 50,00	\$ 50,00
			\$ 930,00

Tabla 4.9 Costos de la placa del Faro Infrarrojo

PLACA FARO			
DESCRIPCIÓN	CANTIDAD	PRECIO UNITARIO	PRECIO TOTAL
MICROCONTROLADOR ATTINY85	1	\$ 1,55	\$ 1,55
EMISORES INFRARROJOS	20	\$ 0,70	\$ 14,00
CAPACITORES CERÁMICOS DE 22 Pf	2	\$ 0,12	\$ 0,24
CAPACITORES CERÁMICOS DE 0.10uF	2	\$ 0,12	\$ 0,24
REGULADOR DE VOLTAJE 7812	1	\$ 0,55	\$ 0,55
REGULADOR DE VOLTAJE 7805	1	\$ 0,55	\$ 0,55
RESISTENCIA DE POTENCIA 60 OHMIOS 5 W	1	\$ 0,20	\$ 0,20
RESISTENCIAS DE 330 OHMIOS	1	\$ 0,02	\$ 0,02
RESISTENCIAS DE 1 KOHMIOS	1	\$ 0,02	\$ 0,02
BORNERA AZUL DE DOS TERMINALES	1	\$ 0,25	\$ 0,25
TRANSISTOR 2N3904	1	\$ 0,08	\$ 0,08
CRISTAL RESONADOR PIEZOELÉCTRICO 4Mhz	1	\$ 0,90	\$ 0,90
BAQUELITA FIBRA DE VIDRIO A4	1	\$ 2,50	\$ 2,50
			\$ 21,10

Tabla 4.10 Costos de la placa del Receptor Infrarrojo

PLACA RECEPTOR			
DESCRIPCIÓN	CANTIDAD	PRECIO UNITARIO	PRECIO TOTAL
MICROCONTROLADOR ATMEGA 164P SMD	1	\$ 8,32	\$ 8,32
CAPACITORES ELECTROLÍTICOS 10 uF 25V	9	\$ 0,13	\$ 1,17
LEDS SMD	7	\$ 0,18	\$ 1,26
RESISTENCIAS 330 OHMIOS SMD	9	\$ 0,01	\$ 0,09
RESISTENCIAS 470 OHMIOS SMD	7	\$ 0,01	\$ 0,07
CAPACITORES DE TANTALIO 1uF SMD	9	\$ 0,12	\$ 1,08
DEEP SWITCH DE 3 INTERRUPTORES	1	\$ 0,25	\$ 0,25
REGULADOR DE VOLTAJE A 3.3V 7833 SMD	1	\$ 0,76	\$ 0,76
REGULADOR DE VOLTAJE A 5V 7805 SMD	1	\$ 0,55	\$ 0,55
CRISTAL OSCILADOR PIEZOELÉCTRICO 16 Mhz	1	\$ 0,90	\$ 0,90
CAPACITORES CERÁMICOS 0.1 uF SMD	4	\$ 0,12	\$ 0,48
CAPACITORES CERÁMICOS 22 pF SMD	2	\$ 0,12	\$ 0,24
JUMPER DE DOS POSICIONES	1	\$ 0,25	\$ 0,25
SENSORES INFRARROJOS	9	\$ 1,05	\$ 9,45
BUS DE DATOS 0,50 cm	1	\$ 0,40	\$ 0,40
ELABORACIÓN CIRCUITO EMPRESO CNC	1	\$ 30,00	\$ 30,00
RECEPTOR INFRARROJO TSOP32438	9	\$ 1,05	\$ 9,45
			\$ 64,72

Tabla 4.11 Costos de la placa del emisor Xbee

PLACA EMISOR XBEE			
DESCRIPCIÓN	CANTIDAD	PRECIO UNITARIO	PRECIO TOTAL
Xbee series 1	1	\$ 34,00	\$ 34,00
LEDS SMD	2	\$ 0,15	\$ 0,30
RESISTENCIAS 470 OHMIOS SMD	2	\$ 0,01	\$ 0,02
ELABORACIÓN CIRCUITO EMPRESO CNC	1	\$ 7,50	\$ 7,50
			\$ 41,82

Tabla 4.12 Costo de la placa del receptor Xbee

PLACA RECEPTOR XBEE			
DESCRIPCIÓN	CANTIDAD	PRECIO UNITARIO	PRECIO TOTAL
Xbee series 1	1	\$ 34,00	\$ 34,00
REGULADOR DE VOLTAJE A 3.3V 7833 SMD	1	\$ 0,76	\$ 0,76
LEDS SMD	3	\$ 0,15	\$ 0,45
RESISTENCIAS 470	3	\$ 0,01	\$ 0,03
MAX 232	1	\$ 1,60	\$ 1,60
CAPACITORES ELECTROLÍTICOS 1uF	4	\$ 0,10	\$ 0,40
LED	1	\$ 0,15	\$ 0,15
CONECTOR DB9 HEMBRA	1	\$ 0,50	\$ 0,50
CAJA DE PROYECTO	1	\$ 2,36	\$ 2,36
ELABORACIÓN CIRCUITO EMPRESO CNC	1	\$ 7,50	\$ 7,50
			\$ 47,75

Tabla 4.12 Costo del prototipo del proyecto

DESCRIPCION	PRECIO
CUADRICÓPTERO Y ELEMENTOS VARIOS	\$ 930
PLACA DEL FARO INFRARROJO	\$ 21.10
PLACA DEL CIRCUITO RECEPTOR	\$ 64.72
PLACA EMISOR XBEE	\$ 41.82
PLACA RECEPTOR XBEE	\$ 47.75
TOTAL	\$ 1101.39

Para la consecución de este proyecto de titulación se ha realizado un trabajo de investigación y desarrollo durante 11 meses entre dos personas, dedicándole por cuestiones laborales un promedio de 3 horas diarias por persona sin contar los fines de semana, resultando 120 horas mensuales por 11 meses (ya que se inició el trabajo desde que llegó la plataforma comercial importada) un total de 1320 horas de ingeniería invertidas en el proyecto a un costo de 15 dólares la hora nos da un total de 19800 dólares, los cuales añadidos al costo del prototipo resultaría que el proyecto tuvo un costo total de USD 20901.39 .

4.7 RESULTADOS

Como resultado del presente proyecto de titulación se tiene un cuadricóptero que cumple con todos los objetivos planteados, es decir, la plataforma comercial adquirida se la ensambló y acondicionó para que sea capaz de seguir un móvil terrestre mediante un sistema emisor-receptor, su autonomía de vuelo tiene una media de 20 minutos con una batería de 2000 mAh y 12 minutos con una batería de 1000 mAh, la velocidad de seguimiento al móvil terrestre es relativamente baja ya que el sistema emisor-receptor usado constituye un limitante en este aspecto.



Figura 4.28 Ensamblaje y Control de un Cuadricóptero

El proyecto consta de las siguientes partes principales:

- Un cuadricóptero Parrot A.R. Drone con un juego de 4 hélices y una estructura de fibra de carbono, 4 motores brushless, escudo y casco para vuelos externos.
- Una batería LiPo 11V 2 A/h con su respectivo cargador.
- Una placa de instrumentación con 8 receptores infrarrojos y un microprocesador para procesar las señales.
- Una placa de transmisión de datos con un módulo Xbee.
- Vehículo móvil con un faro infrarrojo alimentado con 28V, 0.06 A.
- Circuito de recepción de datos XBee explorer cable adaptador USB / MiniUSB.
- Software de Instalación.
- Software de implementación del Algoritmo de Control implementado en LabVIEW y ejecutado a través de C# mediante una dll.

El controlador por facilidad fue implementado en LabView el cual se comunica con el programa A.R Drone Control.NET a través de una librería DLL, este último software es el encargado de comunicarse con el cuadricóptero vía WiFi. El móvil

consta de un faro infrarrojo que emite una radiación detectable hasta una altura de 10 m, la cual es sensada por el circuito montado sobre el cuadricóptero, mediante este método se estima la distancia y dirección que el cuadricóptero debe seguir para posicionarse sobre el faro.

El controlador permite comandar al cuadricóptero en modo manual, en donde éste simplemente responde a órdenes remotas de Roll, Pitch o Yaw. En modo automático es posible realizar el seguimiento del móvil en tierra en cualquier dirección a una velocidad moderada, en este caso el móvil le proporciona una ruta o una coordenada a la cual moverse, en esta modalidad también es posible posicionar al cuadricóptero en una coordenada específica, partiendo desde un determinado punto de origen, teniendo en cuenta que ambos aspectos son proporcionados por el usuario.

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- El cuadricóptero AR Drone es una plataforma comercial que se adecuó perfectamente a las necesidades del proyecto, lo que permite concentrarse en el sistema de control de seguimiento autónomo y la instrumentación a implementar, ya que su estabilidad de vuelo es buena debido a la tecnología de navegación embebida en el cuadricóptero.
- El cuadricóptero Parrot A.R Drone tiene un diseño tanto electrónico como mecánico lo suficientemente robusto como para soportar golpes y caídas moderadas, que se presentan al ser manejados por usuarios sin la suficiente experiencia. Cualquier anomalía que se presente, sin haber puesto en riesgo la integridad del mismo, es producto de fallas aerodinámicas resultado de choques, caídas o deformación de las hélices.
- La propulsión del cuadricóptero depende en su totalidad de las hélices, el constante uso, golpes y altas temperaturas las van deformando, por ende su capacidad para formar una columna uniforme de aire con el suficiente empuje para elevarlo disminuye, una hélice en buenas condiciones es flexible, recta y sin ningún tipo de protuberancia en sus curvas.
- El cuadricóptero tiende a comportarse de manera errática cuando sus hélices necesitan de mantenimiento, en condiciones óptimas éste es capaz de despegar perpendicularmente, alcanza alturas de hasta 3 m y se mantiene estable a una altura indicada por el usuario sin necesidad de tomar el control, con un desgaste moderado el despegue es perpendicular con pequeñas oscilaciones, se mantiene estable a cualquier altura pero con el paso del tiempo tiende a descender suavemente y con un desgaste severo el cuadricóptero no despegue fácilmente, alcanza una altura máxima de 50 cm y no es capaz de conservar tal altitud sino que desciende bruscamente a nivel del suelo.

- El controlador implementado es un controlador de posición el cual busca disminuir la posición del cuadricóptero con respecto al faro infrarrojo, con un error variable de aproximadamente 20% dependiendo de la altura de vuelo. A mayor altura incrementa el radio del cono de luz infrarroja, y por ende el cuadricóptero detecta al faro antes de posicionarse sobre el centro del mismo.
- La red Wifi creada por el cuadricóptero para establecer comunicación con una estación de control no es de gran potencia, en un ambiente plagado de contaminación radiante, principalmente con redes WiFi existe una alta probabilidad de que alguna de estas interfieran con el funcionamiento del mismo. En estos ambientes el alcance de la señal se reduce a menos de 50 m, que es el alcance máximo, además dificulta que la estación de control, sea ésta un iPhone, iPad, Pc, etc., detecte a la red del A.R. Drone.
- El control por realimentación de estados resulta más eficiente y sencillo de implementar que un controlador PID siempre y cuando se puedan medir todos los estados del sistema.
- El cuadricóptero es un sistema no lineal e inestable, especialmente si se desconoce el modelo matemático del mismo. Lo único que se puede medir y observar son sus salidas por lo tanto implementar un controlador lineal y particularmente un PID resulta inefectivo, gracias a las salidas se puede conocer los estados del sistema por lo tanto el control por realimentación de estados ofrece un camino efectivo para realizar el control del cuadricóptero, por otro lado, éste es un método de control no lineal que permite linealizar al sistema y analizar la interdependencia de sus variables.
- La información sensorial del cuadricóptero se adquiere cada 3 ms sin embargo existen retardos debido a la transmisión vía WiFi, la decodificación de las tramas de video, procesamiento de información, etc. hacen que se pueda tomar una lectura cada 67 ms, este tiempo de muestreo impide que el seguimiento se pueda realizar a una velocidad moderada, mientras los datos recibidos no se refresquen rápidamente, los estados del sistema no se actualizarán y la salida resultará imprecisa.

- El cuadricóptero es incapaz de mantenerse estable en una determinada posición por mucho tiempo, posteriormente se pueden presentar oscilaciones aleatorias o movimientos involuntarios provocados principalmente por la deformación de las hélices, viento, etc. Lo que hace imposible el uso de un control en lazo abierto.
- La inercia producto del movimiento del cuadricóptero en el aire se reduce considerablemente gracias a que el control por realimentación de estados calcula de una manera adecuada los ángulos de Roll y Pitch para trasladarlo de un punto a otro, su efecto se observa en la velocidad ya que los ángulos correctos hacen que la velocidad aumente al inicio y conforme el cuadricóptero se acerca a su destino, ésta disminuya.
- El Software ARDroneControl.Net fue una pieza clave en la consecución del proyecto ya que puede manejar la comunicación WiFi con el cuadricóptero junto con los datos de navegación y la posibilidad de enviar los comandos de vuelo después de ser recibidos de la librería dinámica creada en LabVIEW, que facilitó la implementación, visualización y procesamiento de los sistemas de adquisición de datos y el algoritmo de control implementado.
- Es indispensable que los emisores infrarrojos que forman el faro a ser seguido por el cuadricóptero AR Drone estén oscilando siempre a una frecuencia fija de 38 KHz para que el desempeño del cuadricóptero en el seguimiento del faro no se vea afectado por el nivel de iluminación ni por fuentes de luz infrarroja existentes en el ambiente donde se realiza el seguimiento.
- Las placas de la instrumentación adicional montada en el cuadricóptero junto con los sensores infrarrojos tuvieron que ser rediseñadas 2 veces, la última con elementos de montaje superficial para aprovechar mejor el espacio con el afán de reducir en forma significativa el peso que soporta el cuadricóptero en pleno vuelo tratando también de no influir en la aerodinámica del cuadricóptero.
- El buen uso de las baterías LiPo, tanto en su carga como descarga, asegura una vida útil prolongada de lo contrario éstas empiezan a inflarse y reducir su

capacidad de carga, tornándose en un punto nocivo para la salud y para el ambiente.

5.2 RECOMENDACIONES

- En los dispositivos que generan redes WiFi de baja potencia, su señal fácilmente es anulada o distorsionada por la contaminación radiante del ambiente, por ello para lograr un óptimo funcionamiento del mismo es recomendable trabajar en ambientes controlados como subsuelos, garajes o sitios con paredes gruesas que reducen al mínimo la radiación del ambiente.
- Es importante dar mantenimiento preventivo a las partes constitutivas del cuadricóptero para asegurar un vuelo estable, este mantenimiento se puede enfocar en el enderezamiento de las hélices y en poner aceite en el eje del motor para evitar que la fricción desgaste la guía del mismo.
- Se recomienda que proyectos futuros usen al AR Drone como plataforma de vuelo para la investigación e implementación de algoritmos de desarrollo para UAVs dado que esta plataforma comercial provee de comunicación WiFi con el computador, además de que se puede tener acceso relativamente fácil a sus datos de navegación.
- Para mejorar los tiempos de procesamiento de los datos al usar dos programas comunicados con una dll, se puede usar el toolkit de LabVIEW “ar_drone_toolkit” para comunicarse y controlar al AR Drone, esto no se realizó en este proyecto porque el toolkit se encuentra aún en desarrollo y presenta incompatibilidades con ciertas versiones y herramientas de LabView, teniendo en cuenta que está diseñado exclusivamente para Windows 7, además el enlace de descarga contiene un toolkit incompleto en el cual no funcionan los comandos para la dirección del cuadricóptero .

REFERENCIAS BIBLIOGRÁFICAS

- [1] R. MAYORGA, “Sistema de Navegación para Vehículos Aéreos Cuadricópteros”, Proyecto de titulación en Ingeniería Técnica Aeronáutica, Universidad Politécnica Superior de Cataluña. España. 2009.
- [2] Defense Program Acquisition News “Unmanned Aerial Vehicles News”, [Online]. Disponible: <http://www.defenseindustrydial.com>
- [3] The Forum of the Americas, Diálogo, “Vehículos Aéreos no tripulados para vigilar las costas ecuatorianas”, [Online]. Disponible: http://www.dialogo-americas.com/en_GB/articles/rmisa/features/regional_news/2011/10/17/ecuador-unmanned-aircraft
- [4] Nadales C., “Control de un Quadrotor mediante la plataforma Arduino”, Proyecto de titulación en Ingeniería Técnica de Telecomunicaciones, Universidad Politécnica Superior de Cataluña. España. 2009.
- [5] FlightGlobal, “Blog acerca de UAVs micro”, [Online]. Disponible: <http://www.flightglobal.com>.
- [6] International Defence Update Magazine ,”Blog acerca UAVs mini”, [Online]. Disponible: <http://defense-update.com>.
- [7] Foro Defensa Chile, “Chile Adquiere UAV Pioner de combate”, [Online]. Disponible: <http://foro-chiledefensa.foro-chile.net>.
- [8] Techmez, “Foro nuevas tecnologías en UAV”, [Online]. Disponible: <http://www.techmez.com>.
- [9] A. Barrientos, J Del Cerro, R. San Martín, C. Rossi, “Vehículos aéreos no tripulados para uso civil”, Proyecto de Titulación, Universidad Politécnica de Madrid. España. 2012.
- [10] A. Arévalo, “Vehículos aéreos no tripulados, descripción y capacidades para la obtención de información”, [Online]. Disponible: <http://www.escriint.cl/php/contenido/UAV%20CORREGIDO.pdf>

- [11] Giro Club Pamplona, "Historia del autogiro, máquina de los hermanos Richet", [Online]. Disponible: <http://www.autogiroescuela.com>
- [12] Nation States, "Foro de tecnología UAV", [Online]. Disponible: <http://forum.nationstates.net>
- [13] Moller International, "Diseño del Moller Skycar", [Online]. Disponible: <http://www.moller.com>
- [14] Brito J., Vissierre D., "Design of a Quadrotor Prototype", Proyecto de titulación. Universidad Técnica de Lisboa, Portugal. 2009
- [15] Pierre J., Callou F. "The Navigation and Control technology inside the AR.Drone micro UAV", [Online]. Disponible: <http://cas.ensmp.fr/~petit/papers/ifac11//PJB.pdf>
- [16] IAPPS. SA, "Características del Parrot A.R. Drone", [Online]. Disponible: <http://iapps.scenebeta.com/noticia/ar-drone-el-cuadrocoptero-teledirigido-por-tu-iphone-ipod-touch-ipad>
- [17] E-Radicontrol, "Foro de circuitos electrónicos de radio frecuencia", [Online]. Disponible: <http://www.e-radiocontrol.com.ar>
- [18] Ascending Technologies, "Cuadricóptero Hummingbird", [Online]. Disponible: <http://www.asctec.de/uav-applications/research/products/asctec-hummingbird/>
- [19] Draganfly Innovation, "Cuadricóptero Dragan Flyer", [Online]. Disponible: <http://www.draganfly.com/uav-helicopter/draganflyer-x4p/>
- [20] Dty Drones, "El Arducóptero", [Online]. Disponible: <http://code.google.com/p/arducopter/wiki/ArduCopter>
- [21] Parrot, "Especificaciones Cuadricóptero Parrot A.R. Drone", [Online]. Disponible: <http://www.parrot.com>

- [22] Hoja de Datos , “Receptor infrarrojo TSOP32438”, [Online]. Disponible:
<http://ec.mouser.com/ProductDetail/Vishay-Semiconductors/TSOP32438SS1V/?qs=sGAEpiMZZMsj2w6QBC0lctknjYSQm97y>
- [23] Krajnik T., Vonasek V, Fiser D., Faijl G., “AR-Drone as a Platform for Robotic Research and Education”, In: Research and Education in Robotics: EU ROBOT 2011, Heidelberg, Springer. 2011.
- [24] Ogata K., “Sistemas de Control en Tiempo Discreto”, Tercera Edición. Editorial Prentice Hall Hispanoamericana, México, 2002.
- [25] Ar Drone Flyers, “Foro arregla tus propias hélices”, [Online]. Disponible:
<http://www.ar-drone-flyers.com/forum/viewtopic.php?f=13&t=3375>
- [26] Hoja de Datos, “Atmel Atmega 164P”, [Online]. Disponible:
<http://www.atmel.com/Images/doc7674.pdf>
- [27] Hoja de Datos, “Atmel ATtiny 85”, [Online]. Disponible:
<http://www.atmel.com/Images/doc2586.pdf>
- [28] Hoja de Datos, “Xbee Pro Series 1”, [Online]. Disponible:
http://www.digi.com/pdf/ds_xbeemultipointmodules.pdf.

ANEXOS

ANEXO A

MANUAL DE USUARIO

A.1 PARTES CONSTITUTIVAS

El proyecto consta de las siguientes partes para su óptimo funcionamiento:

- Un cuadricóptero Parrot A.R. Drone con un juego de 4 hélices y una estructura de fibra de carbono, 4 motores brushless, escudo y casco para vuelos externos.
- Una batería LiPo 11 V 2 A/h con su respectivo cargador.
- Una placa de instrumentación con 8 receptores infrarrojos y un microprocesador para procesar las señales.
- Una placa de transmisión de datos con un módulo Xbee.
- Vehículo móvil con un faro infrarrojo alimentado con 28 V, 0.06 A.
- Circuito de recepción de datos y cable adaptador RS232 / USB con alimentación de 5 V.
- Software de Instalación.



Figura A.1 Partes constitutivas del proyecto.

A.2 FUENTES DE ALIMENTACIÓN

A.2.1 BATERÍAS PARROT A.R. DRONE

Las baterías del cuadricóptero son de tipo LiPo, las cuales con reducido tamaño y peso las hace muy útiles para equipos pequeños que requieran potencia y duración. Las baterías originales tienen una capacidad de 1A/h que le da una autonomía de vuelo de 12 minutos cuando están completamente nuevas, en el mercado existen baterías de 2 A/h que ofrecen autonomía de 20 minutos aproximadamente, éstas no son originales pero tienen el peso y el tamaño adecuado para el cuadricóptero A.R. Drone.



Figura A.2 Baterías LiPo

Un indicador en el software indicará el porcentaje de batería, cuando éste indique una medida inferior al 20% el cuadricóptero no despegará. Las baterías son reutilizables y se las pueden cargar con un cargador exclusivo para las mismas. La Tabla A.1 indica el tiempo de carga para asegurar el buen funcionamiento de las baterías.

Tabla A.1 tiempo de carga y duración de las baterías LiPo

Tipo de batería	Tiempo de carga	Tiempo de Descarga
LiPo 1 A/h	1 hora	12 minutos
LiPo 2 A/h	3 horas	20 minutos

A.2.2 ALIMENTACIÓN DEL MÓVIL

El vehículo móvil es un carro con controles mando sujetos a él mediante un cable lo suficientemente largo para que se mueva una distancia a la cual se pueda apreciar el seguimiento del cuadricóptero. El carro se lo alimenta con 5V los cuales pueden ser provistos por una fuente de voltaje de laboratorio o por una batería.

**Figura A.3** Móvil y sus controles

El faro infrarrojo necesita de 28 V de alimentación puesto que consta de 20 leds en serie para disminuir el consumo de energía, la corriente que todo el circuito consume es de 60 mA, los 28 V se pueden conseguir con un arreglo de baterías de 9 V o bien mediante una fuente de laboratorio la cual también puede proporcionar 5 V al carro para su movimiento.

A.2.3 ALIMENTACIÓN DEL CIRCUITO DE RECEPCIÓN DE DATOS.

El circuito de recepción es un Xbee explorer que necesita de 5 V para su funcionamiento los cuales son proporcionados por un puerto USB de la PC, este módulo no es más que un adaptador de RS232 a USB.

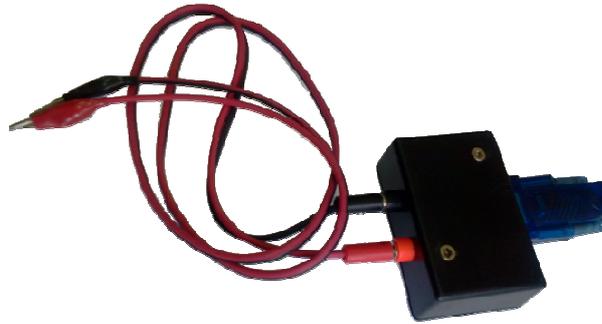


Figura A.4 alimentación circuito receptor.

A.3 ENCENDIDO DEL CUADRICÓPTERO PARROT A.R. DRONE.

Para encender el cuadricóptero se debe conectar la batería al terminal de alimentación del mismo, una vez conectado los 4 leds de la placa de los motores se pondrán de color rojo hasta cuando éste haya inicializado sus circuitos y genere la red WiFi a la cual se lo va a enlazar.

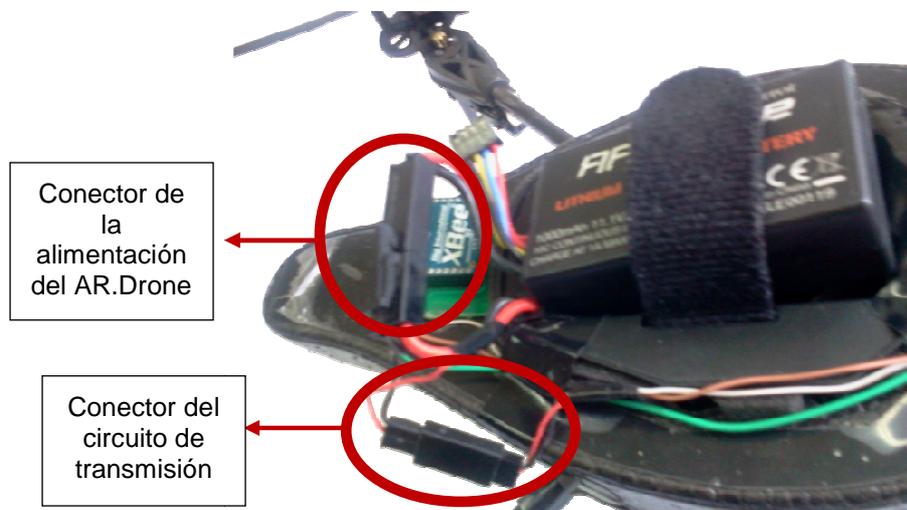


Figura A.5 encendido del cuadricóptero

Es posible que una vez conectado el cuadricóptero su red WiFi no sea visible, esto se debe a la contaminación radiante que existe en el ambiente, si esto sucede es recomendable cambiar de sitio y volver a encender el cuadricóptero.



Figura A.6 Red WiFi del Parrot A.R. Drone.

A.4 ENCENDIDO DE LA PLACA DE SENSORES.

La placa de sensores se enciende únicamente conectándola a la batería del cuadricóptero al encenderla se observará 1 led verde que indica que el circuito está alimentado.

Es necesario colocar al Dipswitch y al jumper en la posición que indica la Figura A.7, de no ser así algunos sensores dejarán de funcionar ya que estos elementos ayudan a programar al microprocesador de la placa.

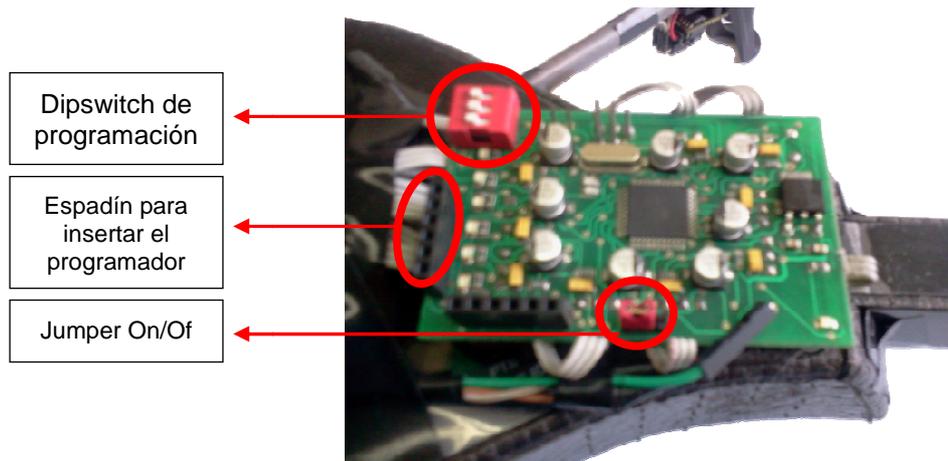


Figura A.7 descripción de la placa de sensores

A.5 MANEJO DEL SOFTWARE

A.5.1 MANEJO DEL SOFTWARE A.R.DRONE CONTROL .NET

El software A.R Drone Control .NET es el encargado de controlar el movimiento del cuadricóptero, este software es un paquete de subrutinas programadas en C#, las cuales es necesario compilar previo a la puesta en marcha para lo cual es necesario seguir los siguientes pasos:

1. Abrir el proyecto en Microsoft Visual Estudio 10.
2. En la ventana del explorador del proyecto buscar ARDroneUI_WPF, dar clic derecho y seleccionar "Set as start up Project".
3. Compilar el proyecto en Debug – Start Debugging, si la PC se encuentra conectado a la red WiFi del cuadricóptero aparecerá la ventana de la Figura A.8, de lo contrario aparecerá una ventana de error.

Nota: El paso número 2 permite seleccionar como proyecto de inicio a la sección donde se encuentra programada la interfaz principal, este paso se lo hace únicamente la primera vez que se va a compilar el programa ya que esta opción queda configurada automáticamente.

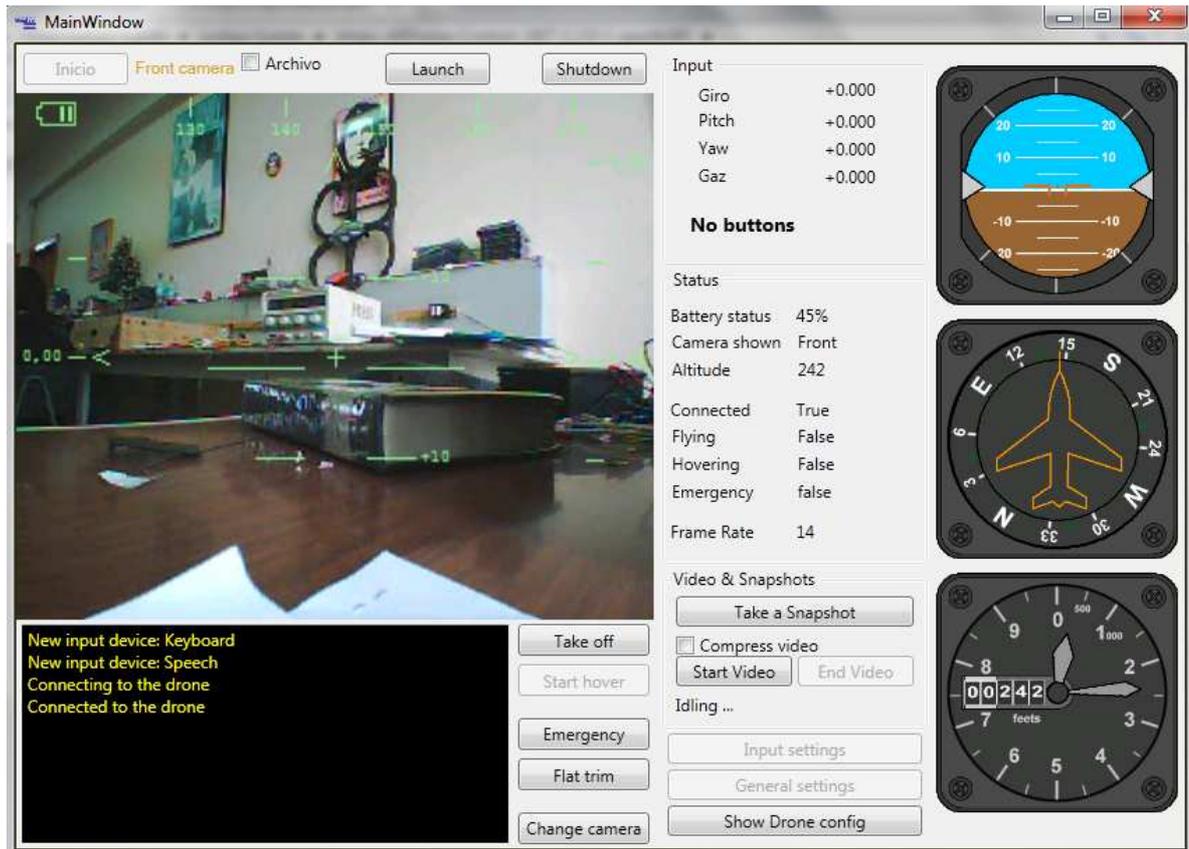


Figura A.8 Ventana de inicio del ARDrone Control .NET

Una vez compilado el proyecto se da clic en el botón “inicio” de la Figura A.8 e inmediatamente aparecerá el video de la cámara del cuadricóptero.

A continuación se detalla los principales botones que se encuentran en la Figura A.8:

- **Take off:** permite despegar el cuadricóptero, también puede emplearse la tecla “Shift”.
- **Landing:** aterriza el cuadricóptero, también puede emplearse la tecla “Shift” después de haber despegado.
- **Flat Trim:** encera los sensores inerciales a una posición horizontal determinada, antes de despegar es necesario encerar los sensores para que despegue perpendicularmente y se mantenga estable.
- **Emergency:** detiene el movimiento de los motores en caso de emergencias, es importante evitar pulsarlo durante el vuelo ya que el cuadricóptero caerá bruscamente y se puede presentar daños en el mismo.

- **Change camera:** cambia la vista de la cámara frontal a la inferior o una mezcla de las dos cámaras.
- **Archivo:** permite establecer la comunicación entre LabView y C# a través de la DLL, esta opción se la activa cuando se pone el cuadricóptero en modo automático.

La ventana del ARDrone Control .NET también tiene indicadores de estado de la batería, altura en cm, estado del cuadricóptero y otros que indican que valor de ángulo en el rango -1 a 1 se está enviando al cuadricóptero.

Para manejar el cuadricóptero, por defecto se utilizan las siguientes teclas:

- **W:** movimiento hacia adelante (Pitch).
- **S:** movimiento hacia atrás (Pitch).
- **A:** movimiento hacia la izquierda (Roll).
- **D:** movimiento hacia la derecha (Roll)
- **Flecha abajo**  : Eleva el cuadricóptero.
- **Flecha arriba**  : Desciende el cuadricóptero.
- **Flecha derecha**  : gira al cuadricóptero hacia la derecha en su propio eje (Yaw).
- **Flecha izquierda**  : gira al cuadricóptero hacia la izquierda en su propio eje (Yaw).

Si se desea es posible configurar las teclas para comodidad del usuario dando clic en Input settings, este botón aparece únicamente antes de clic en inicio es decir antes de empezar a volar al cuadricóptero, como se indica en la Figura A.9.



Figura A.9 Ventana de inicio del AR.Drone Control .NET

A.5.2 MANEJO DE LA INTERFAZ DE LABVIEW

Antes de abrir la interfaz es necesario verificar si no se ha realizado algún cambio en la misma, de lo contrario se debe actualizar la DLL para aplicar dichos cambios siguiendo los siguientes pasos:

1. Guardar todos los cambios realizados.
2. Abrir el proyecto de LabView llamado "untitled Project 1" que contiene la interfaz principal y las funciones de la DLL.
3. En la parte inferior en Build specifications / clic derecho en My DLL / Build.
4. En la carpeta donde se guardó la DLL copiar el archivo "sharedlib.dll" en bin/ debug / ARDroneUI_WPF de la carpeta que contiene el programa A.RDrone control .NET.

Desde la interfaz ARDrone Control.NET se da clic en Launch para llamar a la interfaz de LabView, ésta aparece con los valores que se colocan por defecto, es decir para que funcione en modo automático, con el setpoint también

automático listo para realizar seguimiento, parámetros de comunicación y los valores de las ganancias predeterminadas. La funcionalidad del resto de las opciones de la interfaz se describe en el punto 3.3.3

Después se despegan y se colocan al cuadricóptero sobre el faro infrarrojo, se da clic en Start para iniciar el controlador y ese lugar se toma como referencia del origen de coordenadas. Si se quiere iniciar el controlador en un punto diferente es necesario borrar los datos previamente almacenados, para lo que se da clic en el botón Reset antes de que éste empiece a funcionar.

A.5 PROCEDIMIENTO PARA REALIZAR EL SEGUIMIENTO DEL FARO

El procedimiento para poner en marcha el seguimiento automático del faro infrarrojo es el siguiente:

1. Encender el cuadricóptero y las placas a bordo y conectar la PC a la red WiFi del mismo.
2. Conectar el faro infrarrojo a 27V, el receptor Xbee a 5V, el móvil a 5V y el cable serial / USB a la PC.
3. Compilar el proyecto ARdroneUI_WPF y dar clic en inicio.
4. Clic en “launch” para abrir la interfaz en LabView.
5. Asegurarse que los controles estén fijos en modo automático y setpoint automático, con las ganancias de la realimentación de estados adecuada.
6. Clic en “flat trim” en la ventana del AR Drone control .NET, despegar el cuadricóptero y posicionarlo sobre el faro.
7. Una vez posicionado en la ventana del AR Drone control .NET clic en “archivo” y en la ventana de la interfaz de LabView clic en “start” para comenzar el seguimiento.
8. En caso de comportamientos anómalos desactivar la opción “archivo” para tomar el control, no es posible controlarlo manualmente desde la ventana del AR Drone control .NET cuando la opción “archivo” está activada en cuyo caso es preferible usar la opción manual de la interfaz de LabView.
9. Aterrizar suavemente el cuadricóptero después de haber cumplido su objetivo.

10. En caso de querer realizar nuevamente el seguimiento del móvil se debe dar clic en el botón “reset” de la interfaz de LabView para reiniciar el integrador.

Para posicionar al cuadricóptero en una coordenada específica se procede de la siguiente manera:

- a. Repetir los pasos del 1 al 4 explicados anteriormente.
- b. Asegurarse que los controles estén fijos en modo automático y setpoint manual, con las ganancias de la realimentación de estados adecuadas.
- c. Ingresar las coordenadas para eje x en el recuadro Set point Roll y para el eje y en el recuadro Set point Pitch, ambas en milímetros.
- d. Repetir los pasos del 7 al 10 explicados anteriormente.

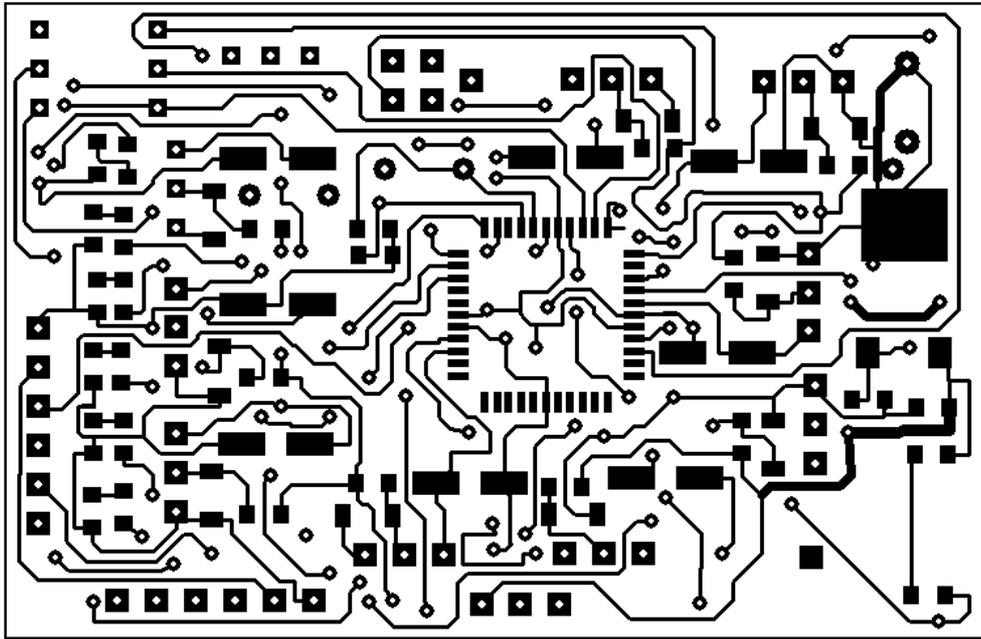


Figura B.2 Ruteado placa smd, vista superior

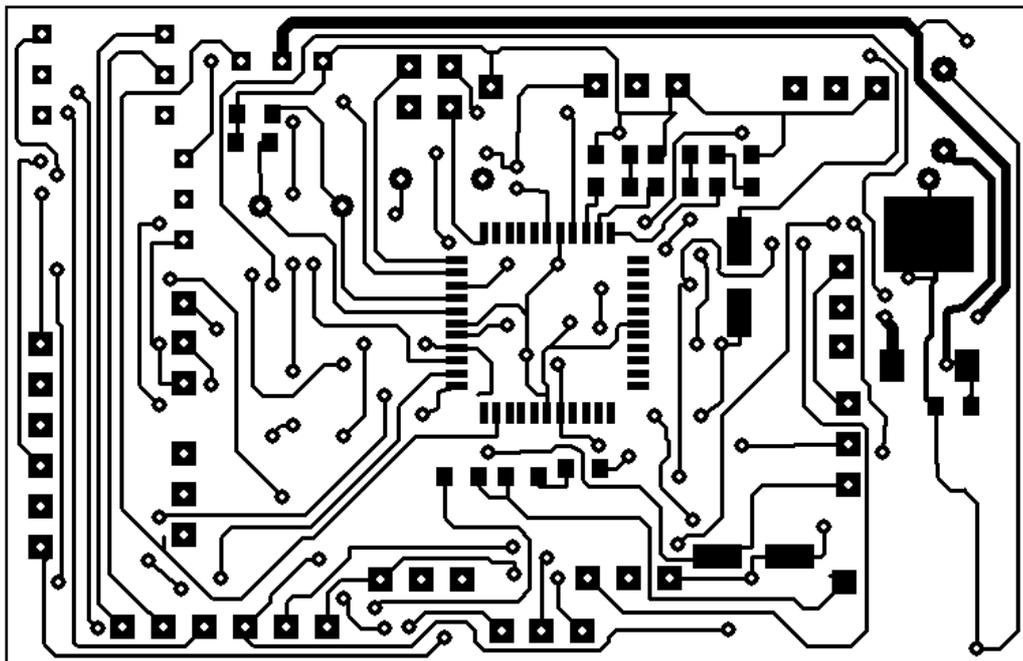


Figura B.3 Ruteado placa smd, vista inferior con microcontrolador sólo para el envío de datos

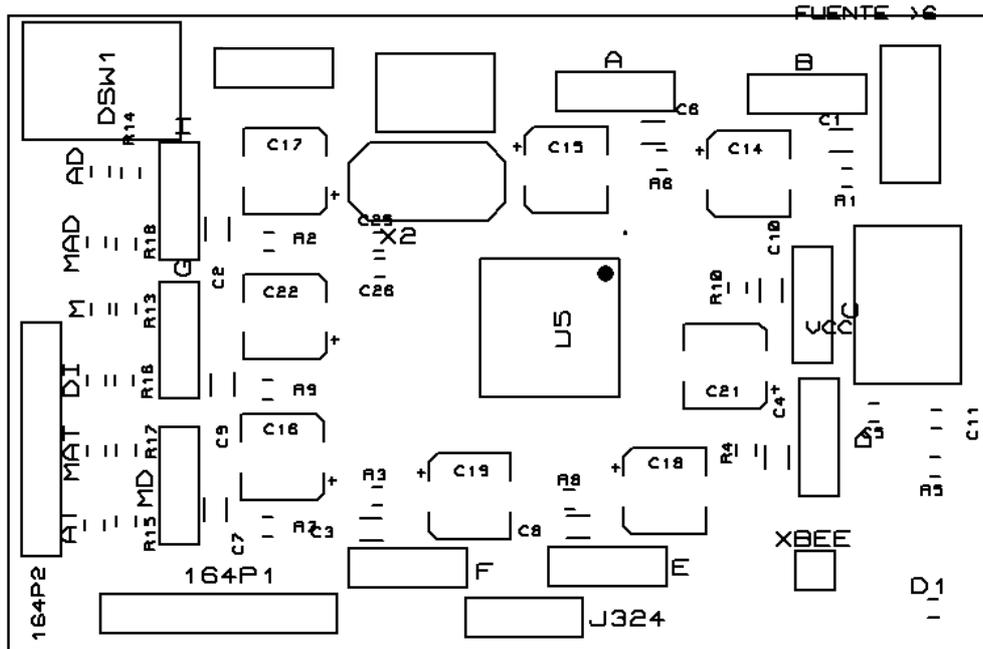


Figura B.4 Plano de posición de los elementos



Figura B.5 Circuito acondicionador de sensores y de transmisión de datos, vista superior

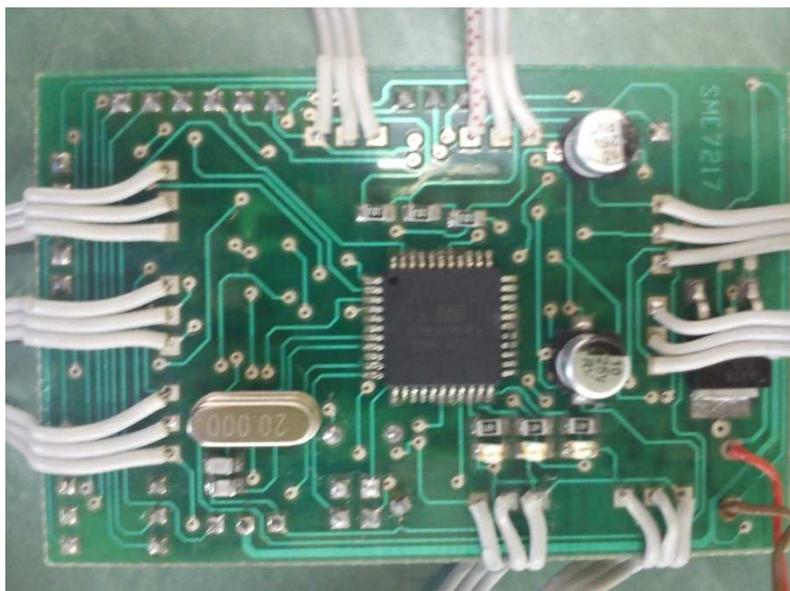


Figura B.6 Circuito acondicionador de sensores y de transmisión de datos, vista inferior

Tabla B.1 Lista de elementos de la tarjeta de instrumentación

DESCRIPCIÓN	ELEMENTO
U5	MICROCONTROLADOR ATMega 164P SMD
164P1	ESPADÍN HEMBRA
164P2	ESPADÍN HEMBRA
AT	LED AZUL SMD
MAT	LED AZUL SMD
DI	LED AZUL SMD
MAT	LED AZUL SMD
MAD	LED AZUL SMD
AD	LED AZUL SMD
C2-C9	CAPACITORES DE TANTALIO 1 μ F SMD
C14-C22	CAPACITORES ELECTROLÍTICOS DE 10 μ F 25 V
VCC	REGULADOR DE VOLTAJE A 5V 7805 SMD
J324	JUMPER DE 2 POSICIONES
XBEE	CONECTOR XBEE
DSW1	DEEP SWITCH DE 3 INTERRUPTORES
A-H	RECEPTORES INFRARROJOS TSOP32428
R1-R9	RESISTENCIAS 330 OHMIOS SMD
R12-R18	RESISTENCIAS 470 OHMIOS SMD
VCC1	REGULADOR DE VOLTAJE A 3.3V 7833 SMD
X2	CRISTAL OSCILADOR PIEZOELÉCTRICO 6 MHz

B.2 FARO INFRARROJO

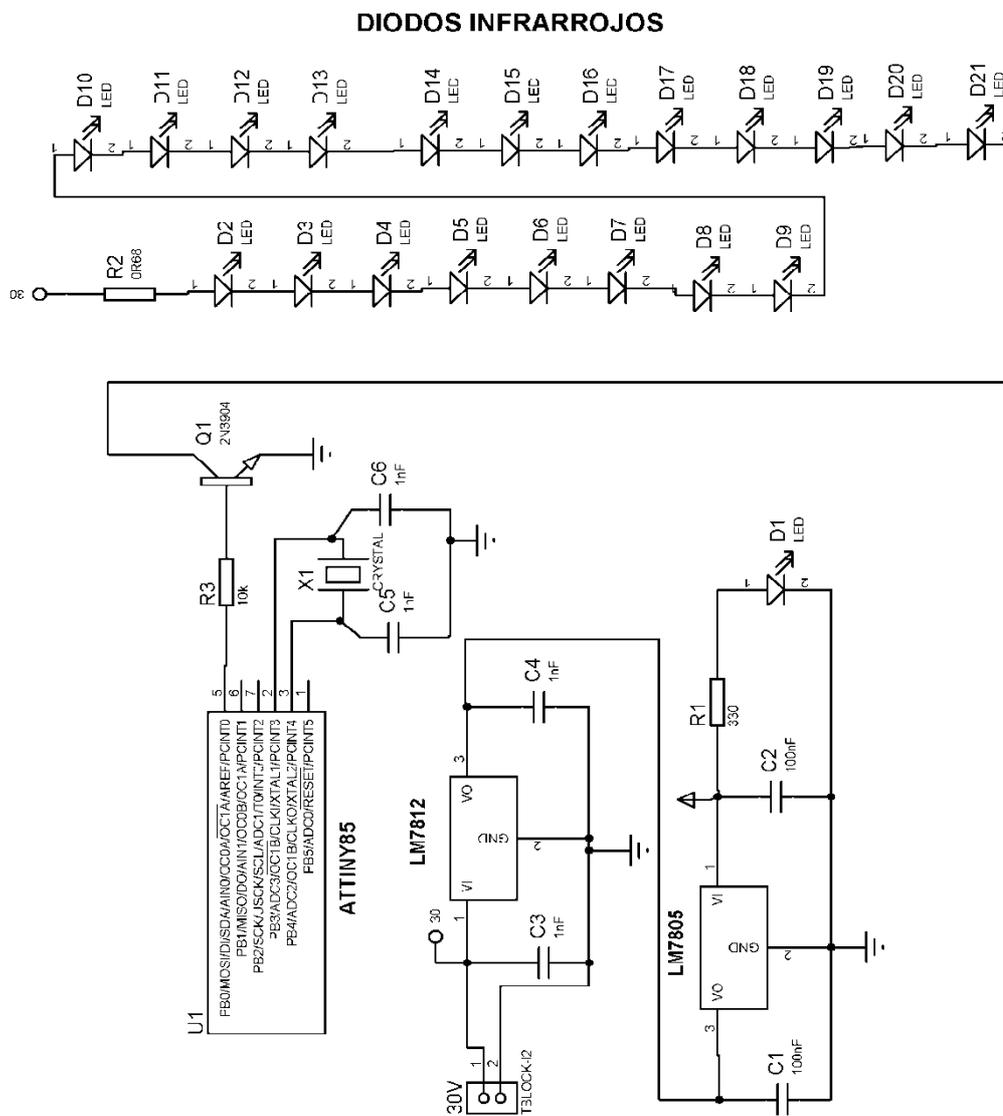


Figura B.7 Esquemático del circuito emisor infrarrojo

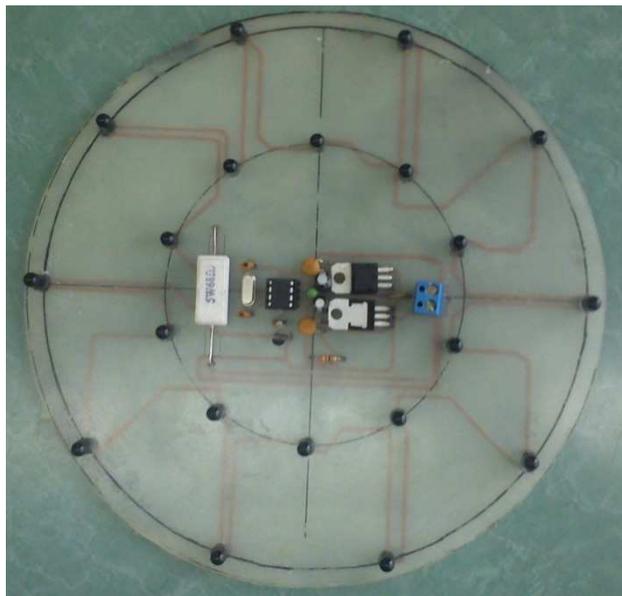


Figura B.10 Placa del emisor infrarrojo

Tabla B.2 Lista de elementos del Faro infrarrojo

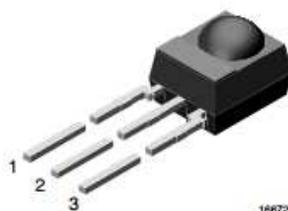
DESCRIPCIÓN	ELEMENTO
U1	MICROCONTROLADOR ATTINY85
D1-D20	EMISORES INFRARROJOS
C1-C2	CAPACITORES CERÁMICOS DE 22 Pf
C3-C4	CAPACITORES CERÁMICOS DE 0.10uF
U2	REGULADOR DE VOLTAJE 7812
U3	REGULADOR DE VOLTAJE 7805
R2	RESISTENCIA DE POTENCIA 60 OHMIOS 5 W
R1	RESISTENCIA DE 330 OHMIOS
R3	RESISTENCIAS DE 1 KOHMIOS
U4	BORNERA AZUL DE DOS TERMINALES
Q1	TRANSISTOR 2N3904
X1	CRISTAL RESONADOR PIEZOELÉCTRICO 4Mhz

ANEXO C

HOJAS DE DATOS

C.1 RECEPTOR INFRARROJO

IR Receiver Modules for Remote Control Systems

**MECHANICAL DATA**

Pinning for TSOP348..., TSOP344...:

1 = OUT, 2 = GND, 3 = V_S

Pinning for TSOP322..., TSOP324...:

1 = OUT, 2 = V_S , 3 = GND

FEATURES

- Very low supply current
- Photo detector and preamplifier in one package
- Internal filter for PCM frequency
- Improved shielding against EMI
- Supply voltage: 2.5 V to 5.5 V
- Improved immunity against ambient light
- Insensitive to supply voltage ripple and noise
- Material categorization: For definitions of compliance please see www.vishay.com/doc?99912

**DESCRIPTION**

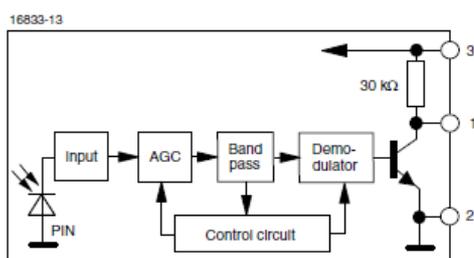
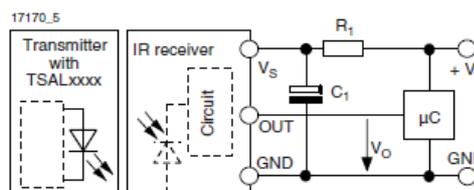
These products are miniaturized receivers for infrared remote control systems. A PIN diode and a preamplifier are assembled on a lead frame, the epoxy package acts as an IR filter.

The demodulated output signal can be directly decoded by a microprocessor. The TSOP348..., TSOP322... are compatible with all common IR remote control data formats. The TSOP324..., TSOP344... are optimized to suppress almost all spurious pulses from energy saving fluorescent lamps but will also suppress some data signals.

This component has not been qualified according to automotive specifications.

PARTS TABLE

CARRIER FREQUENCY	STANDARD APPLICATIONS (AGC2)		VERY NOISY ENVIRONMENTS (AGC4)	
	PINNING			
	1 = OUT, 2 = GND, 3 = V_S	1 = OUT, 2 = V_S , 3 = GND	1 = OUT, 2 = GND, 3 = V_S	1 = OUT, 2 = V_S , 3 = GND
30 kHz	TSOP34830	TSOP32230	TSOP34430	TSOP32430
33 kHz	TSOP34833	TSOP32233	TSOP34433	TSOP32433
36 kHz	TSOP34836	TSOP32236	TSOP34436	TSOP32436
38 kHz	TSOP34838	TSOP32238	TSOP34438	TSOP32438
40 kHz	TSOP34840	TSOP32240	TSOP34440	TSOP32440
56 kHz	TSOP34856	TSOP32256	TSOP34456	TSOP32456

BLOCK DIAGRAM**APPLICATION CIRCUIT**

R_1 and C_1 are recommended for protection against EOS. Components should be in the range of $33 \Omega < R_1 < 1 \text{ k}\Omega$, $C_1 > 0.1 \mu\text{F}$.

ABSOLUTE MAXIMUM RATINGS				
PARAMETER	TEST CONDITION	SYMBOL	VALUE	UNIT
Supply voltage		V_S	- 0.3 to + 6	V
Supply current		I_S	3	mA
Output voltage		V_O	- 0.3 to ($V_S + 0.3$)	V
Output current		I_O	5	mA
Junction temperature		T_j	100	°C
Storage temperature range		T_{stg}	- 25 to + 85	°C
Operating temperature range		T_{amb}	- 25 to + 85	°C
Power consumption	$T_{amb} \leq 85 \text{ °C}$	P_{tot}	10	mW
Soldering temperature	$t \leq 10 \text{ s, } 1 \text{ mm from case}$	T_{sd}	260	°C

ELECTRICAL AND OPTICAL CHARACTERISTICS ($T_{amb} = 25 \text{ °C}$, unless otherwise specified)						
PARAMETER	TEST CONDITION	SYMBOL	MIN.	TYP.	MAX.	UNIT
Supply current	$E_v = 0, V_S = 3.3 \text{ V}$	I_{SD}	0.27	0.35	0.45	mA
	$E_v = 40 \text{ klx, sunlight}$	I_{SH}		0.45		mA
Supply voltage		V_S	2.5		5.5	V
Transmission distance	$E_v = 0$, test signal see fig. 1, IR diode TSAL6200, $I_f = 250 \text{ mA}$	d		45		m
Output voltage low	$I_{OSL} = 0.5 \text{ mA, } E_e = 0.7 \text{ mW/m}^2$, test signal see fig. 1	V_{OSL}			100	mV
Minimum irradiance	Pulse width tolerance: $t_{pi} - 5/f_0 < t_{po} < t_{pi} + 6/f_0$, test signal see fig. 1	$E_e \text{ min.}$		0.1	0.25	mW/m ²
Maximum irradiance	$t_{pi} - 5/f_0 < t_{po} < t_{pi} + 6/f_0$, test signal see fig. 1	$E_e \text{ max.}$	30			W/m ²
Directivity	Angle of half transmission distance	$\Phi_{1/2}$		± 45		deg

TYPICAL CHARACTERISTICS ($T_{amb} = 25 \text{ °C}$, unless otherwise specified)

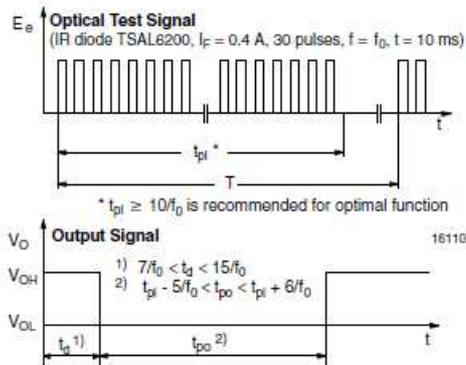


Fig. 1 - Output Active Low

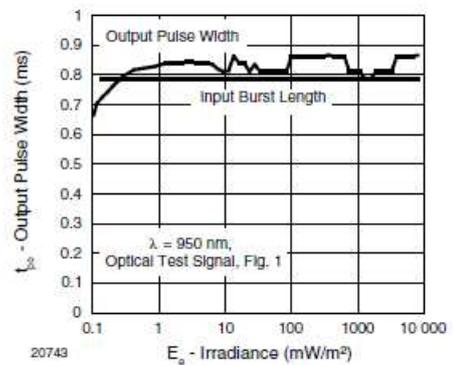


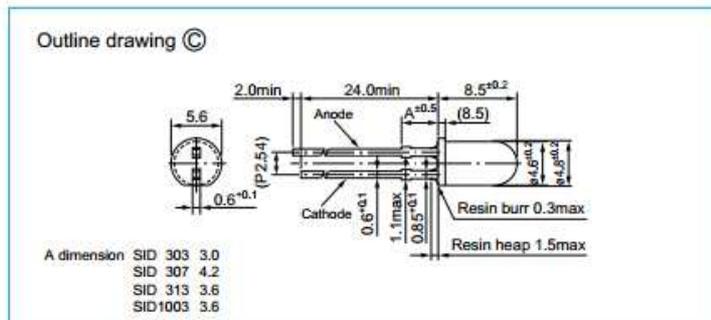
Fig. 2 - Pulse Length and Sensitivity in Dark Ambient

C.2 MÓDULO XBEE PRO

Platform	XBee® ZB	XBee-PRO® ZB
Performance		
RF Data Rate	250 kbps	
Indoor/Urban Range	Up to 133 ft (40 m)	Up to 300 ft (90 m) / 200 ft (60 m) Int'l variant
Outdoor/RF Line-of-Sight Range	Up to 400 ft (120 m)	Up to 2 miles (3200m) / 5000 ft (1500 m) Int'l variant
Transmit Power	1.25 mW (+1 dBm) / 2 mW (+3 dBm) boost mode	63 mW (+18 dBm) / Int'l version
Receiver Sensitivity (1% PER)	-96 dBm in boost mode	-102 dBm
Features		
Antenna Type	XBee ZB Adapters: Internal Antenna; XBee-PRO ZB Adapters: External RPSMA Antenna	
Frequency Band	2.4 GHz	
Serial Data Interface	RS-232 DB9M/DTE or RS-485 (6-position wire terminal block) switch selectable between RS-422 half-duplex, RS-422 full duplex and RS-485	
USB Data Interface	USB 2.0 Full Speed (with ESD protection); connects to host via captive 1-meter cable	
Analog IO (AIO) & Digital IO (DIO)	6-position wire terminal block; Analog IO: 0 - 10V, 4 - 20 mA, or +/- 2VDC Differential; Digital IO: Digital Input or sinking driver output	
Networking & Security		
Network Topologies	Point-to-point, Point-to-multipoint, Mesh	
Number of Channels	16	14
Spread Spectrum Type	Direct Sequence Spread Spectrum	
Filtration Options	PAN ID, 64-bit MAC, channel	
Addressing	65,000 available addresses for each channel	
Other		
LEDs	Power: Glows green when external power is supplied (not active when battery powered to limit power drain) Associate: Blinks when unit is connected and associated on an RF network	
Controls	Device reset (internal push button); Identification (internal push button) - relays ID to gateway	
Dimensions (L x W x H) & Weight	RS-232, RS-485, AIO, DIO models: 3.60 in x 1.90 in x 1.20 in (9.14 cm x 4.82 cm x 3.04 cm) 2.29 oz (64.92 g) USB models: 2.87 in x 1.80 in x 0.83 in (7.29 cm x 4.57 cm x 2.10 cm) 1.60 oz (45.36 g)	
Operating Temperature	-40° C to +70° C	
Power Requirements		
Input Voltage	RS-232, RS-485, AIO, DIO models: 3.7-6VDC and 9-30 VDC USB models: Bus powered (+5V)	
Power Consumption	USB: 70 mA (normal operation) / 200 uA (suspend mode)	USB: 70 mA Rx and 250 mA Tx (normal operation) / 200 uA (suspend mode)
Regulatory Approvals		
Emissions	FCC Part 15.247, IC, ETSI/CE, Telec, EN55022 (Class B)	
Safety	UL60950, CAN/CSA C22.2 No.60950	
RoHS	Yes	

C.3 DIODO EMISOR INFRARROJO

■SID300/1003 series (For remote control)



■External dimensions: Unit: mm Tolerance: ±0.3

Absolute maximum ratings (Ta=25°C)

Symbol	Unit	Ratings
P _D	mW	150
I _F	mA	100
ΔI _F	mA/°C	-1.33 (above 25°C)
I _{FP}	mA	1,000 (f = 1kHz, TW ≤ 10μs)
V _R	V	5
Top	°C	-30 to +85
Tstg	°C	-30 to +100

Type No.	Electro-optical characteristics (Ta=25°C)								Outline drawing	
	V _F (V)		Condition I _F (mA)	I _R (μA)	Condition V _R (V)	I _e (mW/sr)	Condition (Constant voltage)	λ _p (nm)		Δλ (nm)
	typ	max								
SID1010CXM (Wide viewing angle)	1.25	1.5	50	10	5	40	V _{CC} =3V Resistor R=2.2Ω	940	45	(A)
SID1010CM (Narrow viewing angle)						85				(B)
SID1K10CXM (High power output, wide viewing angle)	1.20	1.5	50	10	5	75	V _{CC} =3V Resistor R=2.2Ω	940	45	(A)
SID1K10CM (High power output, narrow viewing angle)						140				(B)
SID303C (Wide viewing angle)	1.25	1.4	50	10	5	100	V _{CC} =3V Resistor R=2.2Ω	940	50	©
SID303BR (Wide viewing angle)						220				
SID303BS (Wide viewing angle)						210				
SID307BR (Narrow viewing angle)						210				
SID313BP (Narrow viewing angle)						210				
SID1003BQ (High power output)	1.45	170								

C.3 MANUAL PARROT A.R.DRONE

Primera utilización

Batería

Recargar la batería

1. Seleccione el adaptador correspondiente a su país e introdúzcalo en el transformador. Tiene que escuchar el ruido que confirma la introducción.



2. Conecte la batería al cargador.



Advertencia: No fuerce la introducción del adaptador en el transformador.

3. Conecte el cable de alimentación al transformador y después a la red eléctrica.

Notas:

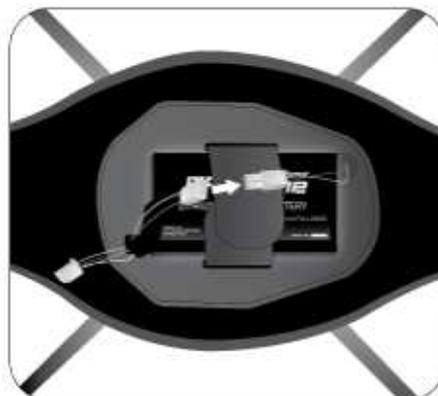
- El tiempo de carga de la batería de es 1 hora 30 minutos.
- Cuando la carga haya finalizado, el LED de estado pasará al verde.
- Los LEDs numerados del 1 al 3 parpadearán cuando las células correspondientes se estén equilibrando.

Comprobar el nivel de batería del iPhone

Para utilizar el AR.Drone en condiciones óptimas, asegúrese de que la batería del iPhone está bien cargada. Un nivel de batería bajo o medio puede hacer que el rendimiento del AR.Drone sea menor (en la reactividad de los mandos, la calidad del flujo de vídeo...).

Colocar la batería

1. Coloque la batería en el alojamiento previsto para ella.
2. Asegúrese de que quede correctamente sujeta utilizando el sistema de agarre.
3. Conéctela al AR.Drone.



Utilización en el interior

Advertencia: Evite que el AR.Drone vuele cerca de niños, animales domésticos u objetos frágiles.

- Instale la carcasa con anillas para proteger el AR.Drone en caso de que choque con algún objeto.



- Coloque el AR.Drone en el centro de una habitación (de 4m x 4m como mínimo), sin obstáculos. Manténgase 1 metro detrás del aparato (la parte delantera del aparato es fácilmente identificable gracias a la cámara).
- En los ajustes de la aplicación AR.Free Flight, desactive las opciones **Outdoor Shell** y **Outdoor Flight**.

Utilización en el exterior

- No utilice el Parrot AR.Drone con malas condiciones meteorológicas (lluvia, fuerte viento o nieve) o cuando las condiciones de visibilidad sean insuficientes (noche).
- Aunque el piloto automático compensa las turbulencias que se puedan producir por el viento, evite que el AR.Drone vuele si la velocidad de viento supera los 15Km/h.

Nota: Recuerde siempre que la fuerza del viento que nota en el lugar en el que está pilotando el AR.Drone puede ser muy diferente a la fuerza del viento que hay en el lugar en el que está el AR.Drone. Puede ser la razón de los cambios de trayectoria no deseados.

- Instale la carcasa sin anillas.



- Coloque el AR.Drone sobre una superficie plana y seca, en una zona sin obstáculos. Manténgase 1 metro detrás del aparato (la parte delantera del aparato es fácilmente identificable gracias a la cámara).
- En los ajustes de la aplicación AR.Free Flight, active las opciones **Outdoor Shell** y **Outdoor Flight**.

Conectar el iPhone al AR.Drone

Activar el modo avión

Le aconsejamos que ponga el iPhone en modo avión para utilizarlo más cómodamente. Para ello, seleccione **Ajustes > Modo avión**.



Conectar el iPhone a la red Wi-Fi del AR.Drone

1. Conecte la batería del AR.Drone y asegúrese de que los LEDs de los motores están en verde.
2. Seleccione en el iPhone **Ajustes > Wi-Fi**. > Aparecerá la lista de periféricos Wi-Fi disponibles.
3. Seleccione **ardrone_parrot**.

Nota: Se puede cambiar el nombre con el que los demás periféricos Wi-Fi ven al AR.Drone. > Consulte la sección [Ajustes básicos](#) para más información.

4. Espere hasta que el nombre de la red aparezca marcado. > El logotipo Wi-Fi aparece en la pantalla del iPhone.



LEDs de motores

Comportamiento de los LEDs	Significado
Los 4 LEDs están en rojo	Puesta en funcionamiento
Todos los LEDs parpadean en rojo, uno tras otro	Los motores se están iniciando.
Los 4 LEDs parpadean en verde	El AR.Drone está despegando o aterrizando.
Los 2 LEDs delanteros están en verde Los 2 LEDs traseros están en rojo	El AR.Drone está volando. Estos colores le permitirán distinguir fácilmente la parte trasera y delantera del AR.Drone (cuando esté lejos del aparato).
Los 4 LEDs parpadean en rojo y verde	Se acaba de borrar la memoria del AR.Drone al pulsar el botón Reset.

LED del sistema

Espere 20 segundos después de conectar la batería al AR.Drone y compruebe después el color del LED del sistema.

Nota: Evite girar el AR.Drone para comprobar el color del LED del sistema. Es preferible que levante el aparato.

- Si el LED está verde, ya puede despegar.
- Si transcurridos 20 segundos, el LED está rojo o naranja, desconecte y vuelva a conectar la batería.

Ajustes

Toque el icono  para acceder al menú de ajustes del AR.Drone.

Botón	Permite...
	Confirmar los ajustes modificados
	Volver a poner el valor de la central de inercia en 0. Asegúrese de que el AR.Drone esté colocado sobre una superficie plana y pulse este botón.
	Restablecer los parámetros por defecto

Nota: Una vez establecida la conexión entre el iPhone y el AR.Drone, ambos aparatos quedan automáticamente emparejados. El AR.Drone sólo se podrá utilizar con el iPhone que acaba de conectar.

› Para utilizar el AR.Drone con otro iPhone, consulte la sección Utilizar el AR.Drone con varios iPhone.

Si tiene algún problema

Comprobar la dirección IP del AR.Drone

Compruebe la dirección IP del AR.Drone una vez establecida la conexión. Para ello, seleccione **Ajustes** > **Wi-Fi** y pulse en la flecha  que está a la derecha en la línea de la red "AR.Drone".

En el campo IP debe haber una dirección IP que comience por 192.168.1.X.

IP Address 192.168.1.2 ✓

Si la dirección IP comienza por 169, no se podrá establecer la conexión entre el iPhone y el AR.Drone.

IP Address 169.254.131.172 ✗

Esta dirección IP incorrecta puede deberse a tres problemas:

Problema	Acción correctiva
----------	-------------------

Ajustes básicos

Opción	Permite...
Accelerometer disabled <input checked="" type="checkbox"/> ON	Activar / desactivar el uso del acelerómetro del iPhone para pilotar el AR.Drone. Cuando el acelerómetro del iPhone está desactivado, un 2º joystick sustituye la tecla del acelerómetro. Utilice esta tecla para hacer avanzar / retroceder el AR.Drone y desplazarlo hacia la derecha / izquierda.
Left-handed <input type="checkbox"/> OFF	Activar / desactivar el modo zurdo. El modo zurdo permite invertir los mandos en el iPhone.
Pairing <input checked="" type="checkbox"/> ON	Eliminar el emparejamiento del iPhone con el AR.Drone. Desactive la opción Pairing para emparejar el AR.Drone con otro iPhone. <i>Nota: Desactivar la opción Pairing tiene el mismo efecto que una pulsación del botón Unpair situado debajo del AR.Drone.</i>
SSID <input type="text" value="ardrone_parrot"/>	Cambiar el nombre con el que los demás periféricos Wi-Fi ven al AR.Drone. <i>Nota: Tiene que salir de la aplicación, reiniciar el AR.Drone y restablecer la conexión con el iPhone para que el nuevo nombre quede registrado.</i>
Altitude Limited <input checked="" type="checkbox"/> ON	Limitar la altitud máxima que puede alcanzar el AR.Drone. <ul style="list-style-type: none"> • Si la opción está activada, el AR.Drone no podrá superar una altitud de 3 metros.

<p>El iPhone y el AR.Drone están demasiado alejados o hay perturbaciones en la señal Wi-Fi</p>	<ol style="list-style-type: none"> 1. Acérquese al AR.Drone. 2. Desactive el modo Wi-Fi. 3. Reinicie el AR.Drone. 4. Conecte el iPhone y el AR.Drone.
<p>El AR.Drone está siendo utilizado con otro iPhone</p>	<ol style="list-style-type: none"> 1. Elimine el emparejamiento del AR.Drone con el iPhone anterior. 2. Conecte el iPhone y el AR.Drone.
<p>El AR.Drone está conectado a otro iPhone</p>	<ol style="list-style-type: none"> 1. Obvie la red Wi-Fi del iPhone conectado. 2. Elimine el emparejamiento del AR.Drone con el iPhone anterior. 3. Conecte el nuevo iPhone al AR.Drone.

Nota: Si tiene varios AR.Drone, no se olvide de [modificar el nombre con el que los ven los demás periféricos Wi-Fi.](#)

Borrar la memoria del AR.Drone

Existen 2 formas de borrar la memoria del AR.Drone:

- Botón Unpair:

1. Arranque de nuevo el AR.Drone desconectando la batería durante al menos 5s y volviendo a conectarla después.
2. Espere hasta que el LED que está debajo del AR.Drone se ponga verde.
3. Pulse el botón **Unpair** utilizando un objeto fino. Tiene que escuchar un pequeño "clic".
> Los 4 LEDs que están en los motores parpadean alternativamente en rojo y en verde.



4. Seleccione en el iPhone la red Wi-Fi generada por el AR.Drone y pulse **Forget this network.**
> Ya puede conectar de nuevo los dos aparatos.

Vuelo libre

Lance la aplicación **AR.Free flight**  para utilizar el AR.Drone en modo de vuelo libre.

> Los LEDs de los motores parpadean en rojo, uno tras otro.

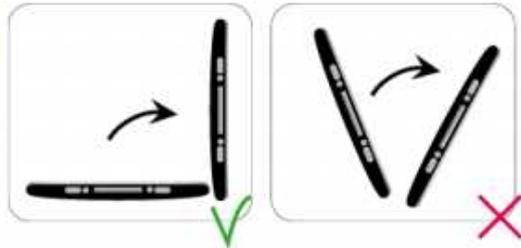
> La señal de vídeo de la cámara frontal del AR.Drone aparecerá en la pantalla de su iPhone.

Advertencia: Asegúrese de que ha conectado la batería del AR.Drone y de que los LEDs de los motores están en verde antes de conectar el iPhone a la red Wi-Fi del AR.Drone y lanzar una aplicación.

Antes de comenzar

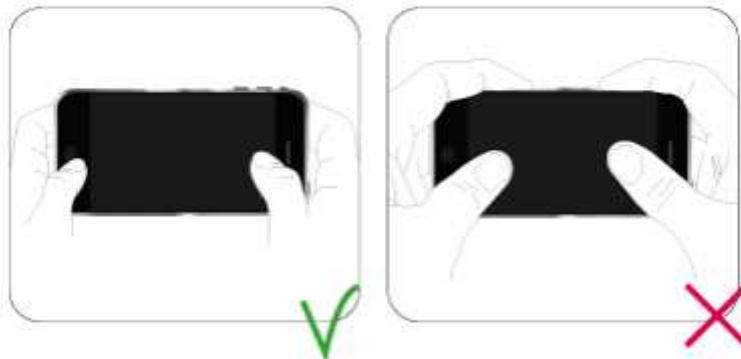
Posicionamiento del iPhone

- Por encima de un cierto umbral ($\sim 90^\circ$ con respecto a la horizontal), el acelerómetro del iPhone no tiene en cuenta los movimientos.
- Le recomendamos que mantenga el iPhone en una posición cercana a la horizontal para evitar tener que adoptar posiciones incómodas y poco eficaces cuando quiera que el AR.Drone retroceda rápidamente.
- Se toma como referencia la posición del iPhone en el momento que el que pulse el botón izquierdo.



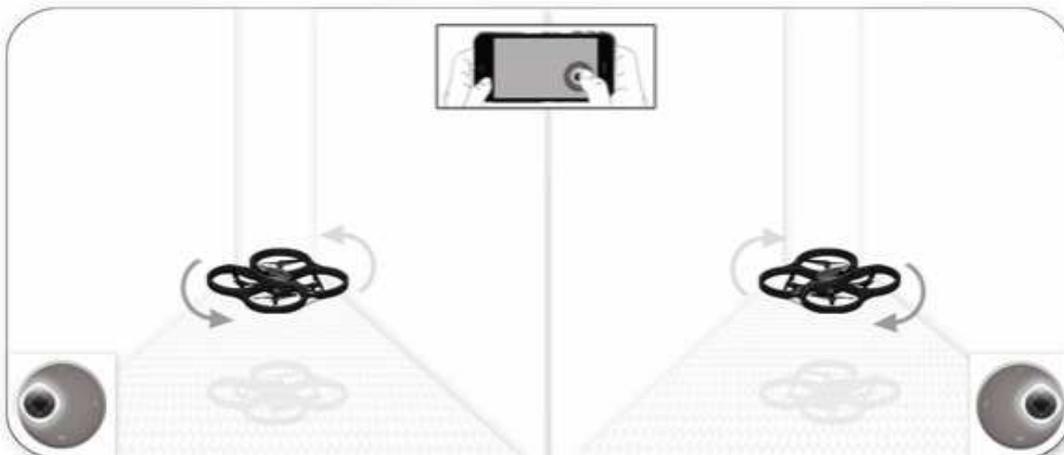
Posición de las manos en el iPhone

Como coloque las manos en el iPhone puede influir en la calidad de la transmisión Wi-Fi.

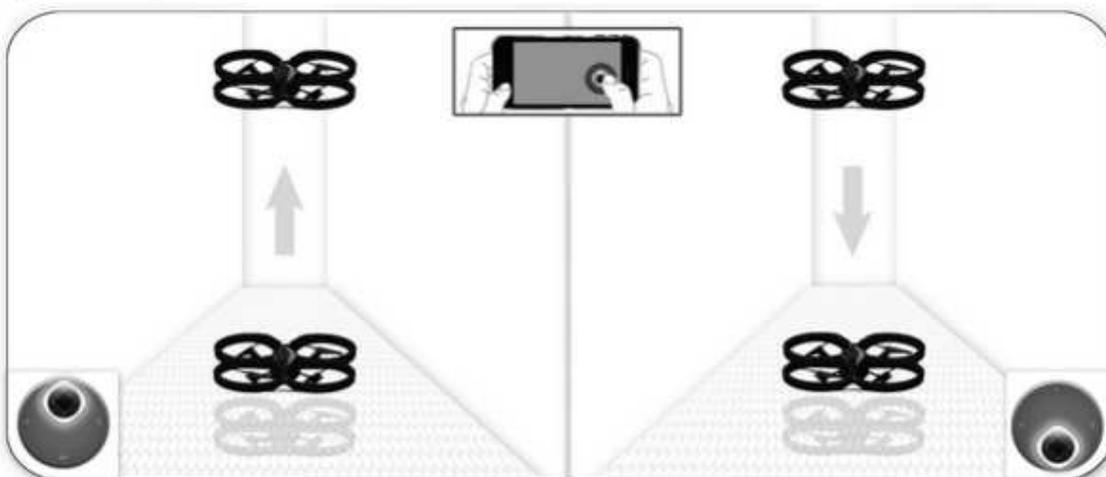


Pilotaje

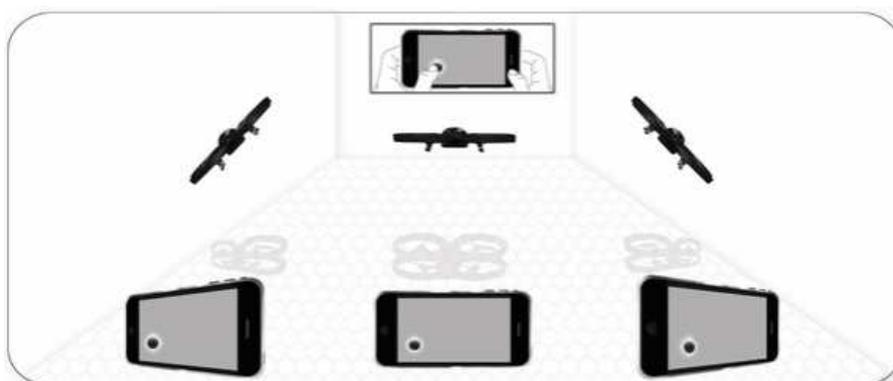
Deslice el joystick (que está en la parte inferior, a la derecha) hacia la derecha / izquierda para hacer que el AR.Drone gire a la derecha / izquierda.



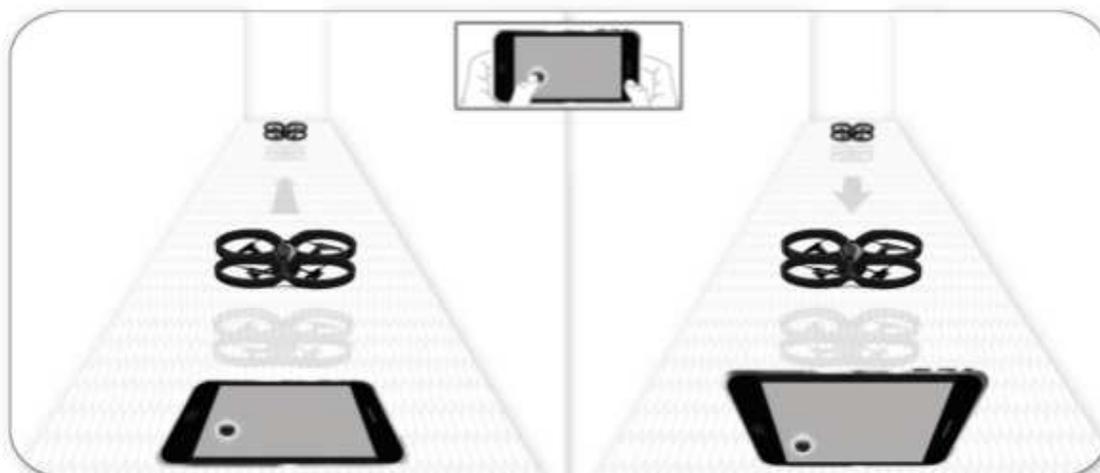
Deslice el joystick (que está en la parte inferior, a la derecha) hacia arriba / abajo para hacer que el AR.Drone suba / baje por tramos de 10cm. Mantenga el joystick hacia arriba / abajo para hacer que suba / baje sin interrupción.



Mantenga el mando de la parte inferior izquierda e incline el iPhone hacia la derecha / izquierda para mover el AR.Drone hacia la derecha / izquierda.



Mantenga el mando de la parte inferior izquierda e incline el iPhone hacia delante / atrás para mover el AR.Drone hacia delante / atrás.



Cambio de cámara

Pulse el mando  para cambiar la vista del AR.Drone.

Aterrizaje

Asegúrese de que el AR.Drone esté encima de una superficie plana, seca y sin obstáculos, y pulse el mando .

Botón Emergency

Si tiene algún problema al pilotar el AR.Drone, pulse el botón **Emergency**.

Advertencia: Pulse el botón Emergency únicamente en caso de emergencia. Se cortarán los motores y el AR.Drone caerá sin importar la altura a la que se encuentre. En la mayoría de los casos, basta con aterrizar el AR.Drone.

Botón Reset

Un mensaje en el iPhone le indicará si es necesario reiniciar el AR.Drone. Si es el caso, pulse el botón **Reset**.