

Centro Nacional de Investigación y Desarrollo Tecnológico

Subdirección Académica

Departamento de Ingeniería Electrónica

TESIS DE MAESTRÍA EN CIENCIAS

**Desarrollo de una Plataforma Electrónica para la Implementación
de Estrategias de Control para un Cuadróptero**

presentada por
Ing. Alberto Vidal Rosas

como requisito para la obtención del grado de
Maestro en Ciencias en Ingeniería Electrónica

Director de tesis
Dr. Carlos Daniel García Beltrán

SEP

SECRETARÍA DE
EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO
Centro Nacional de Investigación y Desarrollo Tecnológico

Cuernavaca, Mor., 10 de enero de 2017.

OFICIO No. DIE/005/2017

Asunto: Aceptación de documento de tesis

DR. GERARDO VICENTE GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO
PRESENTE

Por este conducto, los integrantes de Comité Tutorial jue del **C. Alberto Vidal Rosas** con número de control **M13CE012** de la Maestría en Ciencias en Ingeniería Electrónica, le informamos que hemos revisado el trabajo de tesis profesional titulado **“Desarrollo de una Plataforma Electrónica para la Implementación de Estrategias de Control para un Cuadricoptero”** y hemos encontrado que se han realizado todas las correcciones y observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.

DIRECTOR DE TESIS

CODIRECTOR DE TESIS

Dr. Carlos Daniel García Beltrán
Doctor en Ciencias en Ingeniería Electrónica
Cédula Profesional 8699605

REVISOR 1

Dr. Luis Gerardo Vela Valdés
Doctor en Ciencias en Ingeniería Electrónica
Cédula profesional 7980044

REVISOR 2

Dr. Juan Reyes Reyes
Doctor en Ciencias en la Especialidad de
Control Automático
Cédula profesional 4214833

C.p. M.T.I. María Elena Gómez Torres- Jefa del Departamento de Servicios Escolares.
Estudiante
Expediente

CMAZ/lrr

cenidet
Centro Nacional de Investigación
y Desarrollo Tecnológico

Interior Internado Palmira S/N, Col. Palmira. C.P. 62490 Cuernavaca, Mor.
Tels. (01)777 362-77-70 Ext. 4106, e-mail: direccion@cenidet.edu.mx
www.cenidet.edu.mx



"30 Aniversario del Centro Nacional de Investigación y Desarrollo Tecnológico"

Cuernavaca, Mor., 16 de enero de 2017

OFICIO No. SAC/072/2017

Asunto: Autorización de impresión de tesis

ING. ALBERTO VIDAL ROSAS
CANDIDATO AL GRADO DE MAESTRO EN CIENCIAS
EN INGENIERÍA ELECTRÓNICA
P R E S E N T E

Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado "**Desarrollo de una plataforma Electrónica para la Implementación de Estrategias de Control para un Cuadróptero**", ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

ATENTAMENTE

"CONOCIMIENTO Y TECNOLOGIA AL SERVICIO DE MEXICO"

DR. GERARDO VICENTE GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO



SEP TecNM
CENTRO NACIONAL
DE INVESTIGACIÓN
Y DESARROLLO
TECNOLÓGICO
SUBDIRECCIÓN
ACADÉMICA

C.p. M.T.I. María Elena Gómez Torres- Jefa del Departamento de Servicios Escolares.
Expediente

GVGR/mcr

A mi hija Vanessa.

A mi amada esposa Fabiola.

A mis hermanos.

A mis padres.

El amor incondicional que me han dado es el motor que mueve mi vida.

Agradecimientos

A mi esposa Fabiola, por creer en mí y apoyarme en cada paso, aun cuando estuve lejos en situaciones difíciles.

A mis hermanos, Ernesto, Carmen, Alejandro y Gloria que siempre han creído en mí y me han apoyado a mantener el rumbo, contar su apoyo y consejos me llena de felicidad, nunca podría expresar con palabras todo lo que han hecho por mí a lo largo de mi vida.

A mis padres, que me motivar a dar lo mejor de mí, y esforzarme al máximo en toda tarea, su trabajo es ejemplo a seguir.

A mi asesor el Dr. Carlos Daniel que tuvo la paciencia y perseverancia para ver terminado este trabajo. Agradezco su apoyo incondicional desde el primer día hasta el último.

Al cuerpo de revisores de este trabajo Dr. Alejandro, Dr. Juan Reyes, Dr. Gerardo Vela por exigir de mí el nivel necesario, y enriquecer no solo el contenido sino mi formación con sus comentarios

A los docentes del DIE con quienes tuve el gusto de conocer las bases el control automático Dr. Juan Reyes Reyes, Dr. Alejandro, Dr. Carlos Daniel, Dr. Alvarado, Dr. Carlos Astorga, Dr. Manuel Adam, Dr. Guerrero. Agradezco y atesoro cada minuto que estuve en sus aulas.

Al resto del cuerpo del DIE y toda la institución Cenidet, su trabajo en el aula, oficina, y fuera de la misma, mantiene a flote uno de los centros de investigación más sobresalientes del país, el prestigio que tiene se le debe a su responsabilidad y entrega. Agradecimiento especial a Lorena y Anita, que se preocupan por cada estudiante, en todo momento y en cualquier situación, muchos nos habríamos quedado en el camino de no ser por su guía y consejos.

A mis compañeros de generación, Ricardo Schacht, Amayrani Rosas, Omar Arellano, Rita Ávila, Julio Ayala, y los que se quedaron Eutiquio Bautista, Edgardo Carrera, por ser verdaderos amigos en los buenos momentos y en la batalla diaria a que llamamos vida.

Al resto de compañeros del DIE de distintas generaciones, Abraham Pérez, Marcos A. López, Gabriel Zúñiga, Moisés, Cristian Galicia, Audomaro, Diego Carbot, Juan Pablo, Alberto Arturo, Alberto Gómez, Ricardo Lozano, Gerardo Ortiz, por recibirme, adoptarme, y despedirme con una sonrisa sincera, agradezco su amistad y apoyo.

Ricardo Schacht y Gerardo Ortiz, les tocó doble por apoyarme aún después de haber abandonado por meses la institución, por preocuparse por mis situaciones personales y ofrecer su apoyo cuando posiblemente no pueda pagar en toda una vida todo lo que han hecho por mí.

Finalmente agradezco al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo económico brindado (referencia 512994) para la realización de este trabajo de tesis.

Resumen

En este trabajo se explora el desarrollo de una plataforma electrónica que permita implementar estrategias de control aplicadas en un vehículo multirroto. Se exploran lazos de control, arquitecturas, e implementaciones de quadrotores. Además se plantean los problemas de sensado y localización que se presentan al trabajar con robots móviles aéreos y como los sistemas de visión pueden ayudar en la solución de los mismos. Se presenta el desarrollo de una arquitectura de control basada en una IMU, sensor ultrasónico y controladores PID en paralelo implementados a través de una tarjeta comercial.

La tesis está organizada de la siguiente forma:

En el capítulo 1 se muestra una Introducción a los sistemas multirroto y un análisis del estado del arte de los sistemas de control que han sido utilizados en implementaciones con fines de investigación.

En el capítulo 2 se analiza el problema desde el punto de vista de control definiendo conceptos, y obteniendo el modelo matemático de un vehículo multirroto

En el capítulo 3 se explican los lazos de control que intervienen en la implementación física, y como estos se relacionan entre sí. Además, se exploran técnicas de visión utilizadas para estimar la posición y desplazamiento de vehículos.

En el capítulo 4 se describe el sistema físico que comprende un vehículo aéreo multirroto, y se explica la implementación realizada para control de hexarroto

En el capítulo 5 se presentan los resultados obtenidos de simulación de controladores, y de los experimentos realizados por el sistema de control implementado.

Abstract

This work explores the development of an electronic platform that allows the implementation of control strategies applied to a multi-rotor vehicle. Control loops, architectures and quadrotor implementations are explored. Also, the localization and sensing problems while working with mobile aerial robots are presented and how the vision systems aid on the solution of those. The development of a control architecture based on an IMU, ultrasonic sensor and PID controllers implemented in parallel through a commercial board is presented.

The thesis is organized as follows:

In Chapter 1, an Introduction to the multi-rotor systems and an analysis of the state of the art of the control systems that have been applied on implementation for research purpose is shown.

In Chapter 2. The problem is analyzed from the control point of view defining concepts and obtaining the mathematical model of a multi-rotor vehicle

In chapter 3. The control loops that intervene on the physical implementation and how they are related each other are explained. Also, vision techniques used to estimate position and displacement of vehicles are explored.

In chapter 4, the physical system that a multi-rotor aerial vehicle comprises are described, and the implementation developed to control an hexarrotor is explained.

In chapter 5 the results obtained from the simulation of controllers and the experiments performed by the implemented control are presented.

ÍNDICE

Capítulo 1. Introducción	1
1.1. Planteamiento del problema	1
1.2. Justificación.....	1
1.3. Objetivos	2
1.3.1. Objetivo general	2
1.3.2. Objetivos específicos.....	2
1.4. Estado del arte	2
Capítulo 2. Control de vehículos multirrotores.....	5
2.1. Robótica aérea.....	5
2.2. Vehículos aéreos no tripulados	6
2.3. Modelado de un hexarrotor	7
2.3.1. Marcos de referencia	9
2.3.2. Modelo matemático Newton-Euler	12
Capítulo 3. Lazos de control	17
3.1.1. Controlador PD en cascada	19
3.1.2. Controlador backstepping	20
3.2. Estimación de orientación	24
3.2.1. Algoritmos DCM (Matriz de Cosenos Directores)	24
3.3. Estimación de posición.....	28
3.3.1. Flujo óptico	28
3.3.2. Odometría visual	32
Capítulo 4. Implementación de una arquitectura para control de hexarrotor.....	37
4.1. Descripción de las partes de un multirroto.....	37
4.1.1. Motores BL-DC y hélices	37
4.1.2. Tarjeta de control	38

4.1.3. Sensores convencionales y de percepción de entorno	39
4.1.4. Controlador electrónico de velocidad (ESC).....	40
4.2. Arquitectura de control.....	41
4.3. Instrumentación del hexarrotor	42
4.4. Control estabilizador	44
4.4.1. Implementación del control en una tarjeta Arduino Mega 2560	44
4.4.2. Implementación utilizando mini IMU9.....	44
4.4.3. Generación de referencia utilizando Matlab	46
4.4.4. Mejoras en la implementación	47
Capítulo 5. Resultados obtenidos.....	49
5.1. Resultados del control de orientación	49
5.2. Simulación del control de vehículos multirrotores.....	51
5.2.1. Simulación del seguimiento de una trayectoria.....	54
5.2.2. Comparativa del control PID y un controlador backstepping	55
5.3. Implementación del algoritmo de odometría visual	56
5.4. Flujo óptico	58
Capítulo 6. Conclusiones	59

ÍNDICE DE TABLAS

Tabla 2.1. Definición de coordenadas para los sistemas de referencia	11
Tabla 2.2. Tarjetas de control de proyectos multirrotores.....	38

ÍNDICE DE FIGURAS

Fig. 2.1. El vehículo autónomo Global Hawk.....	6
Fig. 2.2. Fuerzas y pares generados en el hexarrotor	8
Fig. 2.3. Marcos de referencia: a) Inercial y b) móvil.....	9
Fig. 2.4. Control distribuido de un vehículo multirrotores (i=Número de rotores)	11
Fig. 3.1. Control distribuido de un vehículo multirrotores (i=Número de rotores)	17
Fig. 3.2. Control 3NKF Nested Kalman Filter basado en flujo óptico.....	18
Fig. 3.3. Estructura del algoritmo DCM.....	24
Fig. 3.4. Sistemas coordenados	25
Fig. 3.5. Interpretación geométrica de ecuación de flujo óptico	30
Fig. 3.6. Vehículo Mars rover	32
Fig. 3.3. Modelo de proyección central y coordenadas.....	34
Fig. 4.1. Motor BL-DC, a) BLCD tipo outrunner y b) Corriente y voltaje.....	37
Fig. 4.2. Acelerómetro MEMS: A) Movimiento creado por aceleración o gravedad, B) Masa de prueba C) Placas de sensado, D) Resortes de amortiguamiento	40
Fig. 4.3. Control de posición utilizando sensor de efecto hall	41
Fig.4.4. Arquitectura de control: a) PCM-3363, b) Arduino Mega 2560, c) Sensor de orientación, d) Sensor RGB-D	42

Fig. 4.5. Placa de motores y sensores. a) Salidas PWM, b) Puerto UART1 c) Puerto I ² C.	43
Fig. 3.6 Sistema Hexarrotor encendido.....	43
Fig. 4.7. Algoritmo de control y estimación de ángulos de Euler en Arduino para mini imu9.....	45
Fig. 4.8. Interfaz de control de posición y estimación de desplazamiento.....	46
Fig. 4.9. Lectura del sensor UM6 en simulink.....	48
Fig. 4.10. Lectura del sensor PX4FLOW en simulink.....	48
38B Fig. 5.1. Estimación de ángulos de Euler en Arduino para mini imu9.....	49
39B Fig. 5.2. Referencias PWM para controlar velocidad de motores.....	50
Fig. 5.3. Medición de los ángulos pitch y roll.....	50
Fig. 5.4. Ángulo pitch y su señal de control.....	51
Fig. 5.5. Ángulo roll y su señal de control.....	51
Fig. 5.6. Respuesta ante una perturbación externa.....	51
Fig. 5.7. Control de altura, pitch y roll ante entradas escalón arbitrarias.....	53
Fig. 5.8. Desplazamientos y velocidades lineales, Ángulos de Euler y sus derivadas.....	54
Fig. 5.9. Seguimiento de una trayectoria circular.....	55
Fig. 5.10. Seguimiento de una trayectoria en espiral.....	56
Fig. 5.11. Prueba desplazamiento lineal a pie.....	57
Fig. 5.12. Prueba de un desplazamiento en arco.....	58
Fig. 5.13. Flujo óptico entre dos imágenes para un movimiento angular.....	58

Capítulo 1.

Introducción

1.1. Planteamiento del problema

El problema de localización y navegación autónoma, en especial para vehículos aéreos, requiere información del ambiente que no puede ser obtenida con solo el uso de sensores convencionales, los sistemas de visión sobresalen por la riqueza de información que pueden proporcionar, pero a pesar de los diversos enfoques que se pueden abordar, para realizar una implementación se requiere velocidad de procesamiento de una tarjeta abordo o un sistema externo para realizar la estimación de posición, además del uso de otros sensores para fusión de datos. Existe una constante investigación dedicada a la obtención de algoritmos de control y visión que permitan localización y navegación autónoma con prestaciones como velocidad y una relación autonomía/peso aceptable.

1.2. Justificación

El control de vehículos multirrotores es un área de investigación en creciente desarrollo debido a la continua evolución tecnológica que permite dotar a estos vehículos con mayores prestaciones para la mejora de su desempeño, autonomía e incluso tolerancia a fallas.

La principal aplicación actualmente es para fines militares (vigilancia, supervisión y misiones de reconocimiento). Los UAVs (Unmanned Aerial Vehicles) deben poseer las habilidades para sensor su entorno para evitar obstáculos, reconstruir la trayectoria recorrida y comunicación en tiempo real con estaciones en tierra.

Los sistemas actuales de localización de robots móviles aéreos se han implementado principalmente con el uso de sistemas de posicionamiento global para vuelo en exteriores, los cuales poseen una precisión de hasta ± 1 metro dependiendo de la cantidad de satélites que pueda detectar el sistema; y por otro lado los sistemas externos de cámaras de captura de movimiento han permitido la experimentación en interiores con una alta velocidad de actualización de posición pero en espacios de tamaño reducido.

El problema de los sistemas de localización y mapeo simultáneo (Simultaneous Localization and Mapping, SLAM), representa la mayor dificultad para la implementación en aplicaciones reales debido al elevado costo computacional, sin embargo hay casos de éxito de implementaciones utilizando técnicas de visión artificial que han permitido obtener estimación de posición, desplazamiento y orientación incluso como retroalimentación en sistemas de control, aunque siempre apoyados con un sistema remoto de procesamiento.

La odometría visual nace como una solución a los errores producidos al realizar odometría utilizando el giro de las ruedas de vehículos terrestres, de esta forma no afecta a la estimación de desplazamiento si el vehículo se desliza, atasca o pierde agarre produciendo disparidad entre el giro de las ruedas y el desplazamiento generado, se ha extendido su uso a vehículos aéreos ya que estos no tienen ninguna interacción física que pueda tomarse como retroalimentación.

1.3. Objetivos

1.3.1. Objetivo general

Diseñar una plataforma electrónica que permita implementar y probar estrategias de control para un multirrotor, desarrollando la arquitectura y algoritmos necesarios para la adquisición, comunicación y procesamiento de las señales necesarias, para iniciar el proceso de creación de un vehículo multirrotor autónomo que posea un sistema de localización relativa utilizando odometría visual.

1.3.2. Objetivos específicos

1. Modelado del hexarrotor
2. Diseño de un controlador
3. Instrumentación electrónica en una arquitectura abierta
4. Implementación de algoritmos de odometría visual
5. Prueba de algoritmos de flujo óptico y odometría visual como retroalimentación de posición.

1.4. Estado del arte

La arquitectura de control a bordo de un UAV debe integrar una variedad de información de sensores y controladores de velocidad, además se busca agregar capacidades de percepción del ambiente para seguimiento de referencias, objetos, evasión de obstáculos y planeación de trayectorias. Aunque estas

tareas tienen solución para vehículos terrestres, las implementaciones actuales en vehículos aéreos aún presentan puntos débiles, principalmente por la reducida capacidad de carga y el consumo de energía.

La localización inercial inicia su desarrollo en la década de 1940 en aplicaciones para cohetes guiados, (King, 1998) y se vuelve muy popular para localización de vehículos aéreos. Originalmente los sistemas de localización inerciales fueron diseñados para ejecutarse independientemente de sensores externos, de esta forma el interés principal de la investigación fue incrementar la precisión del sensor. La desventaja de estos sistemas era su gran tamaño, el consumo de potencia y un alto costo, lo que decreta su uso en aplicaciones aéreas.

El desarrollo del filtro de Kalman en la década de 1960 significó un método para combinar las mediciones inerciales con otros sensores de localización para reducir el error generado por sensores de menor costo. Algunos ejemplos de filtro de Kalman se dan en (Giovanni, 1979) donde se combina un giróscopo de precisión, un sensor magnético, y sensores de velocidad del viento, y en (Var-Shalom & Fortmann, 1998)

El uso de GPS ha sido constante en la robótica aérea, ya que el costo no es elevado y el error puede no tener impacto en desplazamientos grandes donde no se requiere de tanta precisión, por otra parte sus prestaciones son limitadas en interiores ya que no podrá detectar un número adecuado de satélites lo cual incrementa su error o no le permite obtener su localización.

En (Pégard, 2013) se propone una estructura de control basada en controladores de tipo PD, los cuales aprovechan que el sistema se puede separar en subsistemas independientes, esta solución es la más fácil de implementar en dispositivos con poca capacidad de procesamiento. De igual forma en (Baránek & Šolc, 2012) a través de linealización en el punto de operación estable, se propone aplicar una ley de control PD para cada subsistema lineal, y al obtener la ecuación de estado en lazo cerrado por comparación se lleva a la forma general de un sistema de segundo grado, de esta forma se ajustan las ganancias para obtener una respuesta transitoria deseable.

En (Arellano Muro, Luque Vega, Castillo Toledo, & Loukianov, 2013) se presenta un controlador backstepping para el seguimiento de trayectoria en el cual utiliza el modelo no lineal y un observador de modos deslizantes para garantizar robustez frente a perturbaciones externas.

Por otro lado en (Liu, Derawi, Kim, & Zhong, 2013) se propone controlador óptimo de orientación, el cual se compone de un Regulador Lineal Cuadrático (LQR) y un compensador robusto que limita la influencia de incertidumbre.

Con una metodología similar en (Artale, Barbaraci, Milazzo, Orlando, & Ricciardello, 2013) se propone un Regulador Lineal Cuadrático, el cual por la naturaleza de su obtención tiende a producir oscilación, por lo que se requiere agregar un controlador PI que elimine el error en estado estacionario y es sintonizado para obtener una respuesta rápida con sobreimpulso.

Por otro lado, para alimentar los controladores de vuelo se abordan sistemas de visión, utilizando técnicas de mapeo, odometría y flujo óptico, por ejemplo (Farid, Isabelle, & Kenzo, 2009) proponen un algoritmo basado en estimación de flujo óptico, desplazamiento y mapeo utilizando una cámara de baja resolución y sensores de bajo costo aplicado al despegue, *hovering*, y aterrizaje respecto de marcadores visuales. Este tipo de algoritmos se definen como de navegación pasiva que consiste solamente en calcular sus parámetros de movimiento en un marco de referencia de navegación.

Un caso distinto ocurre en (Adrien, Jean-Christophe, & Dario, 2013), donde el flujo óptico se utiliza sólo para compensar el error acumulativo producido por los sensores inerciales, a través de la dirección de una Restricción de la Dirección Traslacional del Flujo Óptico *TOFDC* por sus siglas en inglés.

La navegación activa, es la encargada de asegurar la autonomía del vehículo con respecto a cambios en su ambiente, por su simplicidad el flujo óptico ha sido recurrentemente utilizado en la solución de problemas de navegación activa, como por ejemplo centrar el vehículo al desplazarse por pasillos, lo cual ha sido presentado utilizando una sola cámara en (Laurent, Stephane, Yves, & Jean-Arcady, 2005), dos cámaras frontales por (Antonis & Fredrik, 1999), un sistema de 3 cámaras a bordo en (Julien, Franc, Stephane, & Nicolaes, 2006), y con una cámara omnidireccional en (Joseph, Gregory, Badri, & J. Sean, 2009). Para resolver este problema se iguala el flujo óptico en los 2 lados del vehículo mientras se desplaza al frente.

Para la detección de obstáculos, se han presentado técnicas en el llamado tiempo de contacto, que se define como el tiempo en que un objeto cruza un plano infinito descrito por un plano en la imagen, también han sido exploradas técnicas basadas en la naturaleza, como en (Jean-Christophe & Dario, 2005), que utiliza directamente la divergencia del flujo óptico medido con un arreglo de sensores de flujo.

Capítulo 2.

Control de vehículos multirrotores

2.1. Robótica aérea

La robótica móvil aérea incluye el diseño, modelado y fabricación de robots que son capaces de desplazarse en el aire que permitan realizar tareas en las que no se podría utilizar un robot terrestre, con fines académicos e industriales.

Algunas de las aplicaciones principalmente estudiadas son de observación, mapeo y transporte de cargas, aunque también es posible el monitoreo y mantenimiento de estructuras. Los robots aéreos ofrecen ventajas en tareas donde la velocidad del desplazamiento es importante en mayor medida que el peso de la carga a transportar, ya que es difícil dotar de capacidad de carga a aeronaves que son ya en sí más pesadas que el aire.

Los robots aéreos se pueden clasificar de acuerdo a la configuración para lograr sustentarse en el aire en tres categorías:

- I. **Aeronaves de ala fija:** Son la forma convencional de aeronave y la más eficiente, conocidos comúnmente como aviones, su principio de sustentación consiste en pasar aire a una velocidad constante a través de sus alas, para de esa forma compensar el peso del avión con las fuerzas generadas por la diferencia de presión aerodinámica que se genera por la forma y posición de los actuadores de las alas. Requieren de un medio de propulsión que les permita controlar la velocidad de desplazamiento el cual puede ser generado por 1 o más rotores.
- II. **Aeronaves batientes:** Utilizan rotores u otro tipo de artefactos que generen propulsión en el mismo eje que la gravedad pero con dirección contraria para así compensar su peso, una ventaja sobre las aeronaves de ala fija es que permiten el despegue y aterrizaje vertical, pero son menos eficientes ya que no se mejora la eficiencia usando principios aerodinámicos. Pueden tener 2 o más rotores que sean actuados o no en cuyo caso utilizan pares diferenciales para manipular la dirección de la fuerza generada.

- III. **Aeronaves dirigibles:** Su principio de sustentación consiste en el principio de arquímedes que indica que *“La fuerza boyante sobre un objeto inmerso en un fluido es igual al peso del fluido desplazado por ese objeto”*, este tipo de aeronave utiliza gases de menor densidad que el aire almacenados en depósitos que generen una fuerza boyante que compense su peso, se utilizan para vuelo de larga duración, pero sus capacidades de maniobrabilidad son ampliamente superadas por los otros dos tipos de aeronave.

2.2. Vehículos aéreos no tripulados

Un vehículo aéreo no tripulado o también conocido como UAV por sus siglas en inglés, es aquel que se auto sustenta en el aire y es controlado remotamente o realiza operaciones de forma autónoma.

A lo largo de las últimas décadas se han realizado una gran cantidad de estudios enfocados en incrementar la duración de vuelo y la capacidad de carga, lo cual ha propiciado la generación de una variedad de vehículos que cubren ciertas necesidades, existen UAVs de Gran Altura y larga duración de vuelo (HALE) como por ejemplo los Global Hawks del corporativo Northrop Grumman Ryan, esta aeronave vuela entre 1 y 19 km de altura hasta por 35 horas con capacidad de carga de hasta 1360 kg y es utilizada principalmente por la fuerza aérea y la marina de los Estados Unidos.

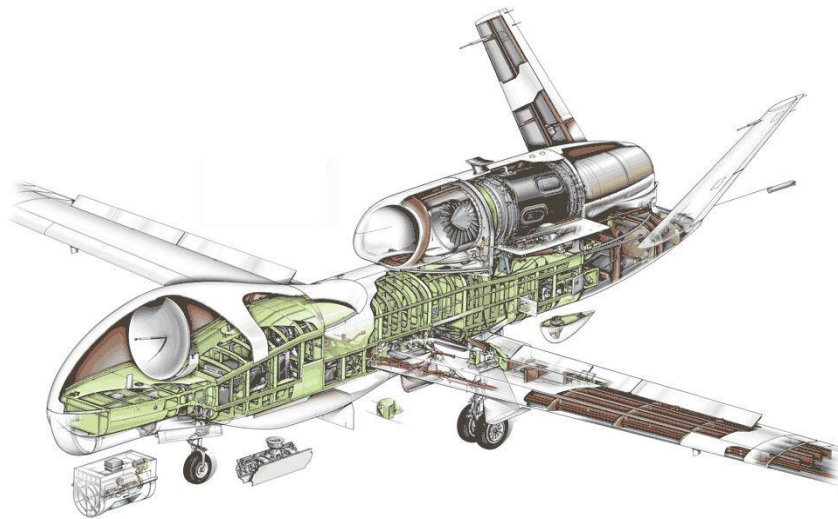


Fig. 2.1. El vehículo autónomo Global Hawk

El concepto de un vehículo multirrotores se refiere a cualquier vehículo aéreo que posea más de dos rotores, dentro de los cuales, los quadrotores han cobrado popularidad debido a la simplicidad de su construcción e incluso su control.

El quadrotor es una alternativa al helicóptero, de menor costo y complejidad. Este genera empuje diferencial a través de sus cuatro rotores, es capaz de mantener su posición y moverse sin involucrar partes móviles adicionales, es considerado un sistema subactuado debido a que sólo tiene cuatro actuadores para controlar seis grados de libertad, de los cuales actúa directamente sobre la rotación alrededor de cada eje y la altura.

La navegación autónoma se puede dividir en:

- Localización: Conocer las coordenadas del robot respecto a un sistema de referencia.
- Planeación de tareas: Decidir qué acciones y en qué orden se han de realizar para completar una misión.
- Planeación de movimiento: Resolver el problema de navegación para desplazarse desde una posición inicial a una final.

En general los métodos para localizar robots móviles en el mundo real se pueden dividir en:

- Posicionamiento relativo: Calculan la posición del robot desde un punto de referencia inicial con una tasa de actualización alta.
- Posicionamiento absoluto: Detectar y reconocer diferentes características en el ambiente del robot para que el robot pueda alcanzar un destino y realizar tareas específicas.

Para dotar al vehículo de autonomía se requiere el uso de los sensores disponibles sobre él y los algoritmos que le permitan procesar tales señales. Un robot autónomo debe ser capaz de orientarse y elegir la ruta adecuada que le permita desplazarse mientras completa la tarea encomendada, sin colisiones u otros percances. (Bermúdez, 2002)

Los robots aéreos difieren de los terrestres en el número de grados de libertad y su configuración, además son actuados a través de fuerzas, por lo que su modelado es dinámico y no cinemático. Las variables que se utilizan incluyen los pares necesarios para generar desplazamiento y la fuerza de sustentación del vehículo.

2.3. Modelado de un hexarrotor

En un hexarrotor, se controla la velocidad angular de los seis rotores de forma que se produce el impulso principal (la suma de la fuerza generada verticalmente por los seis motores) y tres momentos angulares: yaw, pitch y roll.

Para evitar que el quadrotor gire alrededor del eje Z debido a la dirección del giro de los motores, los motores M1, M3 y M6 giran en sentido horario mientras M2, M4 y M6 giran en sentido anti-horario, como se muestran en la Fig. 2.2, donde también se muestran las fuerzas que aporta cada motor (f_1, f_2, \dots, f_6) y el impulso principal (U_z).

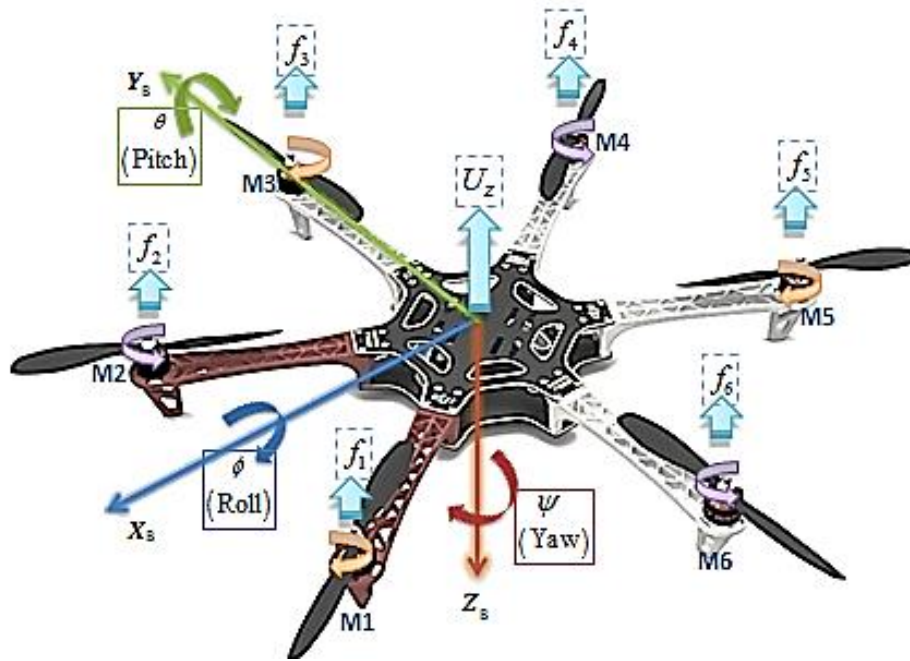


Fig. 2.2. Fuerzas y pares generados en el hexarrotor

Según su aplicación existen dos configuraciones posibles de acuerdo a la orientación que se da al hexarrotor:

- Configuración geométrica clásica o +: Un par de motores se encuentran sobre el eje X.
- Configuración geométrica tipo V o X: No hay motores sobre el eje X, hay un par de motores sobre el eje Y, como se mostró en la fig. 2.

La diferencia de velocidades de motores es lo que modifica la orientación del vehículo, por ejemplo para una configuración en V incrementando la velocidad de los motores M1 y M2 y disminuyendo M4 y M5 se produce un pitch positivo; de forma similar incrementando la velocidad de los motores M2, M3 y M4 mientras se disminuyen M1, M5 y M6 produce un roll positivo.

2.3.1. Marcos de referencia

Para representar la posición y orientación, es necesario definir los sistemas de coordenadas y su origen. Existen dos definiciones importantes relacionadas a estas representaciones, la primera es un sistema de referencia, el cual sólo comprende la definición conceptual de teorías, hipótesis y constantes que permiten situar una triplete de ejes coordenados en el espacio definiendo su origen y orientación; como consecuencia un marco de referencia es la materialización del sistema de referencia a través de observaciones, es decir, se trata de un conjunto de puntos (lugares localizados en la superficie terrestre) con coordenadas y velocidades conocidas y representables a través del sistema.

- Sistema de referencia inercial: Tiene como origen un punto arbitrario localizado en la superficie de la tierra, y los ejes X, Y y Z, se dirigen al Norte, Este y hacia abajo respectivamente (NED). (Fig. 2.3.a).
- Sistema de referencia móvil (del cuerpo): Tiene como origen el centro de gravedad del hexarrotor. (Fig. 2.3.b).

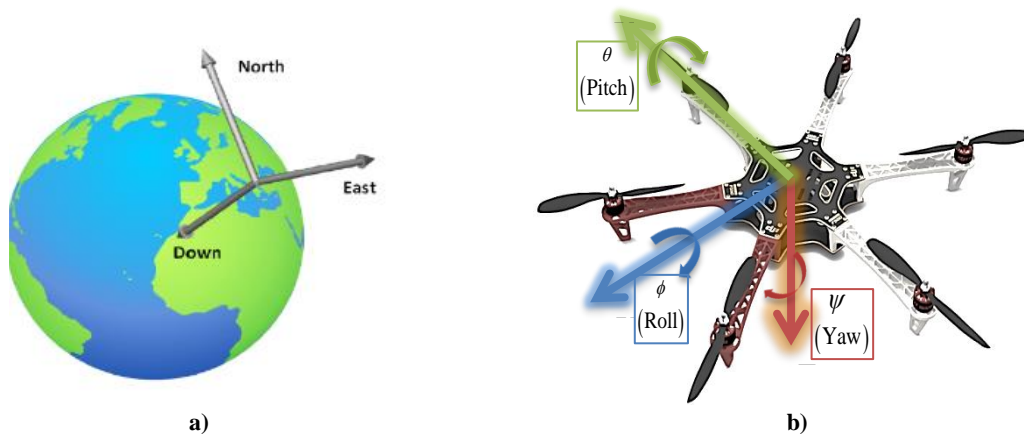


Fig. 2.3. Marcos de referencia: a) Inercial y b) móvil

Las coordenadas que se utilizan en los marcos de referencia se definen en la

Tabla 2.1: (Alaimo, Artale, Milazzo, & Ricciardello, 2013)

Tabla 2.1. Definición de coordenadas para los sistemas de referencia

Marco de referencia inercial (ξ)	Marco de referencia móvil (η)
${}^B x$: Desplazamiento sobre el eje x desde el origen del marco inercial al origen del marco móvil	θ : (Pitch) Es el ángulo de giro alrededor del eje y.
${}^B y$: Desplazamiento sobre el eje y desde el origen del marco inercial al origen del marco móvil	ϕ : (Roll) Es el ángulo de giro alrededor del eje x.
${}^B z$: Desplazamiento sobre el eje z desde el origen del marco inercial al origen del marco móvil	ψ : (Yaw) Es el ángulo de giro alrededor del eje z.

Los ángulos de Euler se utilizan para denotar la orientación a través de 3 rotaciones sucesivas alrededor de los ejes de un cuerpo rígido, el orden en que estas rotaciones producen resultados distintos por lo que debe ser tomado en consideración, además de la definición de convenciones de signo. En esta tesis la secuencia de rotación utilizada es pitch-roll-yaw (eje y, eje x, eje z) y los ángulos positivos son: pitch positivo cuando se levante el frente del vehículo, roll positivo, cuando se rote hacia la izquierda y yaw positivo hacia la derecha del vehículo.

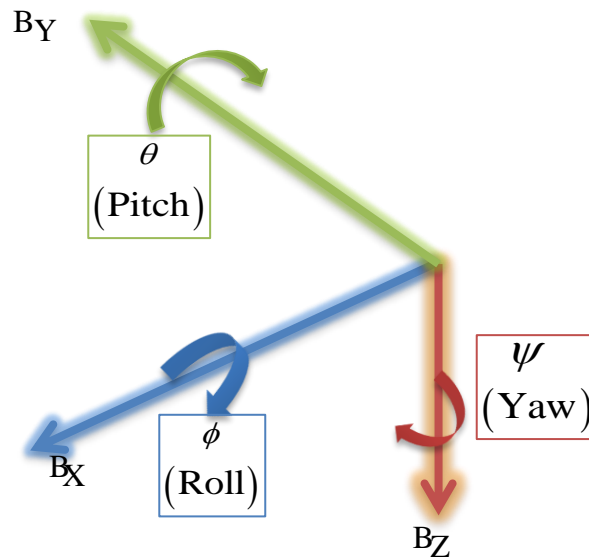


Fig. 2.4. Control distribuido de un vehículo multirrotor (i =Número de rotores)

2.3.2. Modelo matemático Newton-Euler

Un modelado matemático Newton-Euler consiste en formular las dos ecuaciones de Euler para movimiento, estas se utilizan cuando un cuerpo rígido tiene movimiento traslacional y rotacional y fueron creadas para extender las leyes de Newton a cuerpos rígidos.

La primera ecuación de Euler establece que: “La fuerza resultante aplicada a un cuerpo rígido R es igual al producto de la masa del cuerpo rígido y la aceleración del centro de masa del cuerpo rígido en un marco de referencia inercial N: $F = m^N a$ ”

La segunda ecuación de Euler nos dice que: “El par resultante aplicado a un cuerpo rígido R relativo a un punto O fijo en un marco de referencia inercial N es igual a la derivada del momento angular del

cuerpo rígido relativo al punto O en el marco de referencia N: $M_o = \frac{d}{dt} \left({}^N H_o \right)$ ”

De forma que es posible definir la fuerza lineal y el par generado como funciones de los parámetros de un cuerpo rígido y el tiempo, ahora es necesario expresar tales ecuaciones en los términos de los marcos de referencia que se tienen, recordando que el marco de referencia del cuerpo es rotatorio respecto al inercial. En la ecuación (1.1) se expresa la ecuación completa para la dinámica para un cuerpo rígido en un marco de referencia rotatorio.

$$ma = F - 2m\omega \times v - m\dot{\omega} \times r - m\omega \times (\omega \times r) \quad (1.1)$$

Donde v y a son la velocidad y aceleración traslacionales relativas al marco rotatorio, ω es la fuerza angular, r es el vector de posición y los términos que aparecen a la derecha de la ecuación son llamados fuerzas ficticias, no tienen un origen físico sino relativo a la rotación pero provocan que el cuerpo rígido se mueva como si estuviera bajo el efecto de estas fuerzas, siendo $-2m\omega \times v$ la fuerza de coriolis y $-m\omega \times (\omega \times r)$ la fuerza centrífuga.

La fuerza que actúa traslacionalmente sobre el hexarrotor está dada por: (Gonzales Sánchez, Amézquita Brooks, & Liceaga Castro, 2013)

$$F = \frac{d}{dt} (m^B v) + {}^B w \times (m^B v) \quad (1.2)$$

Donde ${}^B v = (u \ v \ w)^T$ representa la velocidad lineal y ${}^B w = ({}^B p \ {}^B q \ {}^B r)^T$ representa la velocidad angular en el marco de referencia del cuerpo. Si se desea la velocidad lineal en el marco de referencia inercial $v_e = \dot{\xi} = ({}^B \dot{x} \ {}^B \dot{y} \ {}^B \dot{z})^T$ se debe utilizar la matriz R ortogonal, que se obtiene de la rotación en secuencia (pitch-roll-yaw) mostrada en (1.3).

$$R = \begin{pmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ c_\psi s_\theta s_\phi - s_\psi c_\phi & s_\psi s_\theta s_\phi + c_\psi c_\phi & c_\theta s_\phi \\ c_\psi s_\theta c_\phi + s_\psi s_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi & c_\theta c_\phi \end{pmatrix} \quad (1.3)$$

Donde c_x , s_x y t_x son $\sin(x)$, $\cos(x)$, $\tan(x)$ respectivamente, entonces: $V_e = R^T {}^B V$

Agregando las fuerzas aerodinámicas en la ecuación (1.2) para obtener la dinámica traslacional, se tiene que las fuerzas que intervienen son el arrastre, el empuje generado por las hélices y la gravedad. Debido a que el vehículo se desplaza a velocidad baja y sus áreas laterales son pequeñas se desprecia la fuerza de arrastre, quedando la siguiente expresión:

$$\ddot{\xi} = \dot{V}_e = \begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix} = \frac{1}{m} \begin{pmatrix} -U_z (\cos \phi \cos \psi \sin \theta + \sin \phi \sin \psi) \\ -U_z (\cos \phi \sin \psi \sin \theta - \sin \phi \cos \psi) \\ -U_z \cos \phi \cos \theta + mg \end{pmatrix} \quad (1.4)$$

Cada rotor i tiene una velocidad angular w_i con dirección al eje Z, que genera una fuerza U_i con un solo componente $f_i = k w_i^2$, donde k es la constante de empuje. (Luukkonen, 2011)

$$U_i = [0 \ 0 \ f_i]^T = [0 \ 0 \ k w_i^2]^T \quad (1.5)$$

El empuje total para el caso del hexarrotor está dado por: $U_B = (0 \ 0 \ U_z)^T$ dónde:

$$U_z = \sum_{i=1}^6 f_i = k \sum_{i=1}^6 w_i^2 \quad (1.6)$$

Re-escribiendo la ecuación de la fuerza total transformando al marco de referencia del cuerpo:

$$R(m\dot{v}_B) + R({}^B v \times (m {}^B v)) = (0 \ 0 \ mg)^T - R^T (0 \ 0 \ U_z)^T \quad (1.7)$$

Donde $R({}^B v \times (m {}^B v)) = 0$ debido a que no se genera fuerza centrífuga.

Realizando las operaciones indicadas se obtiene una expresión completa de la dinámica traslacional:

$$\ddot{\xi} = \dot{V}_e = \begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix} = \frac{1}{m} \begin{pmatrix} -U_z (\cos \phi \cos \psi \sin \theta + \sin \phi \sin \psi) \\ -U_z (\cos \phi \sin \psi \sin \theta - \sin \phi \cos \psi) \\ -U_z \cos \phi \cos \theta + mg \end{pmatrix} \quad (1.8)$$

De forma similar, planteando la fuerza total que actúa rotacionalmente sobre el hexarrotor está dada por:

$$M_B = J \frac{d}{dt} (v_B) + w_B \times (m w_B) \quad (1.9)$$

Donde $v_B = (u \ v \ w)^T$ representa la velocidad lineal del sistema móvil y ${}^B w = ({}^B p \ {}^B q \ {}^B r)^T$ representa la velocidad angular en el marco de referencia del cuerpo, J es la matriz de inercias.

Para obtener las derivadas en el tiempo de los ángulos de Euler, se utiliza la relación:

$$\dot{\Omega} = R_a^{-1} {}^B w \quad (1.10)$$

Donde:

$$R_a^{-1} = \begin{pmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{pmatrix} \quad (1.11)$$

Agregando las fuerzas aerodinámicas, se tiene que el empuje generado por las hélices se puede expresar en momentos de fuerza externos en el marco de referencia inercial por:

$${}^B M = \begin{bmatrix} T_p & T_q & T_r \end{bmatrix} \quad (1.12)$$

Donde los subíndices indican el ángulo sobre el cuál actúan recordando que ${}^B w = \begin{pmatrix} {}^B p & {}^B q & {}^B r \end{pmatrix}^T$ y

$\dot{\Omega} = R_a^{-1} w_b$, se tiene entonces que son el torque en los ángulos pitch, roll y yaw respectivamente.

Re-escribiendo la ecuación de la fuerza total en términos de los ángulos de Euler:

$$J R_a^{-1} {}^B \dot{w} + R_a^{-1} ({}^B w \times J {}^B w) = {}^B M \quad (1.13)$$

Realizando las operaciones indicadas y despejando los ángulos de Euler:

$$\begin{pmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{pmatrix} = \begin{pmatrix} \frac{T_p + T_q s_\phi t_\theta + T_r C_\phi t_\theta}{I_a} & & \frac{T_r C_\phi t_\theta}{I_z} \\ \frac{T_q c_\theta}{I_a} - \frac{T_r s_\phi}{I_z} & & \\ \frac{I_a T_r c_\theta + I_z T_q s_\phi}{I_a I_z c_\theta} & & \end{pmatrix} - \begin{pmatrix} -\frac{(\dot{\psi} + \dot{\phi} s_\theta)}{c_\theta} \\ \dot{\phi} \dot{\psi} c_\theta \\ -\frac{\dot{\theta}(\dot{\phi} + \dot{\psi} s_\theta)}{c_\theta} \end{pmatrix} - \begin{pmatrix} \frac{(I_z - I_a) \dot{\psi} c_\phi c_\theta \dot{\theta}}{I_a} (\dot{\theta} c_\phi + \dot{\psi} s_\phi c_\theta - \dot{\phi} s_\phi s_\theta t_\theta) \\ -\frac{(I_z - I_a) c_\phi}{I_a} (\dot{\psi} c_\phi c_\theta - s_\phi \dot{\theta}) (\dot{\phi} - \dot{\psi} s_\theta) \\ -\frac{(I_z - I_a) s_\phi}{I_a c_\theta} (\dot{\psi} c_\phi c_\theta - s_\phi \dot{\theta}) (\dot{\phi} - \dot{\psi} s_\theta) \end{pmatrix} \quad (1.14)$$

Las ecuaciones 7 y 13 completan el modelo dinámico no lineal del hexarrotor y en general de cualquier vehículo multirroto, las diferencias yacen en la obtención de los momentos de fuerza externos

${}^B M = \begin{bmatrix} T_p & T_q & T_r \end{bmatrix}$ a partir de la configuración geométrica.

En este caso, se tiene la siguiente relación entre los momentos de fuerza y las velocidades de los rotores obtenida usando teorema de Pitágoras y las identidades trigonométricas de ángulos notables:

$$\begin{pmatrix} T_p \\ T_q \\ T_r \end{pmatrix} = \begin{pmatrix} kl \left(-\frac{w_1^2}{2} + \frac{w_2^2}{2} + w_3^2 + \frac{w_4^2}{2} - \frac{w_5^2}{2} - w_6^2 \right) \\ \frac{3}{4} (-w_1^2 - w_2^2 + w_4^2 + w_5^2) \\ b (-w_1^2 + w_2^2 - w_3^2 + w_4^2 - w_5^2 + w_6^2) \end{pmatrix} \quad (1.15)$$

Donde k es la constante de empuje, l es la distancia entre cada rotor y el centro de gravedad del hexarrotor.

Los algoritmos de estimación de posición generalmente producen derivas debido a errores acumulativos y los sistemas de visión requieren mayores prestaciones para el procesamiento de imágenes, así que por lo regular este procesamiento se realiza en computadoras remotas y las señales de control se envían de forma inalámbrica al lazo de control intermedio. Una vez que se ha determinado la referencia angular deseada se obtienen las fuerzas externas que deben generar los motores para lograrlas, así hay una pequeña etapa de control que obtiene la referencia de velocidad para cada rotor que produce a través de la dinámica del multirrotor desplazamiento e inclinación. Cabe destacar que el control de altura se encuentra en el lazo de bajo nivel ya que el empuje debe mantenerse a pesar de que se realicen movimientos rotacionales.

Capítulo 3.

Lazos de control

En el control de vehículos multirrotores se encuentra una tendencia a usar estrategias de control jerárquico para separar el control de posición del control de actitud por grandes motivos, como son el tiempo de procesamiento y el mecanismo de estimación o medición de la posición. Esta configuración se puede observar en el siguiente diagrama a bloques: (Pégard, 2013):

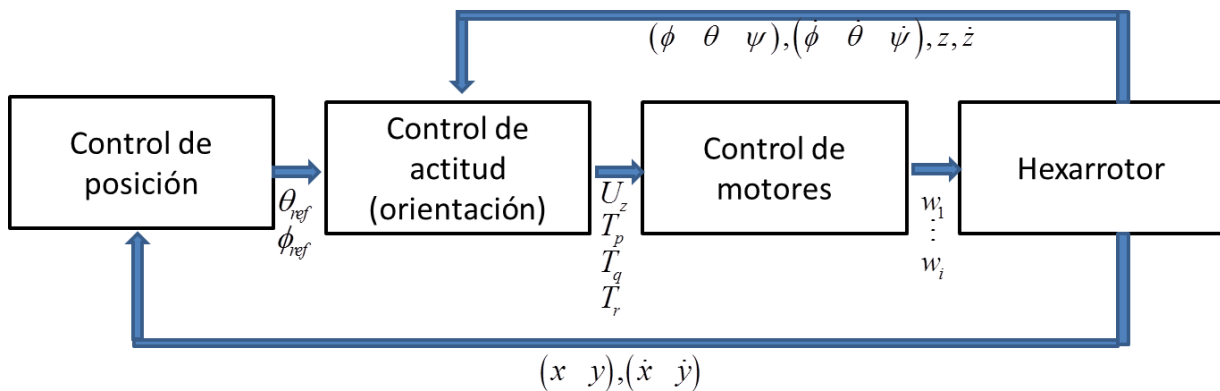


Fig. 3.1. Control distribuido de un vehículo multirrotores (i=Número de rotores)

Se observa que el lazo de control de posición es el nivel más alto en la estructura de control, y a su vez es el que más complejidad presenta ya que requiere conocer la posición del vehículo y ser capaz de producir una señal de momento angular deseado para llevarlo a otra posición deseada cada periodo de muestreo.

El interés particular de los desarrollos de controladores abarca los micro y pequeños UAVs, que vuelan en espacios reducidos desconocidos, donde la información es escasa y debe ser determinada a través de estrategias de alto nivel que proporcionen estimaciones de estados no medibles, a menudo se limitan las condiciones de vuelo en altura respecto del suelo, velocidad de desplazamiento, y cercanía a puntos de equilibrio.

Se puede observar en la **¡Error! No se encuentra el origen de la referencia.** la forma en que se agrupan los diversos algoritmos de control y sus relaciones.

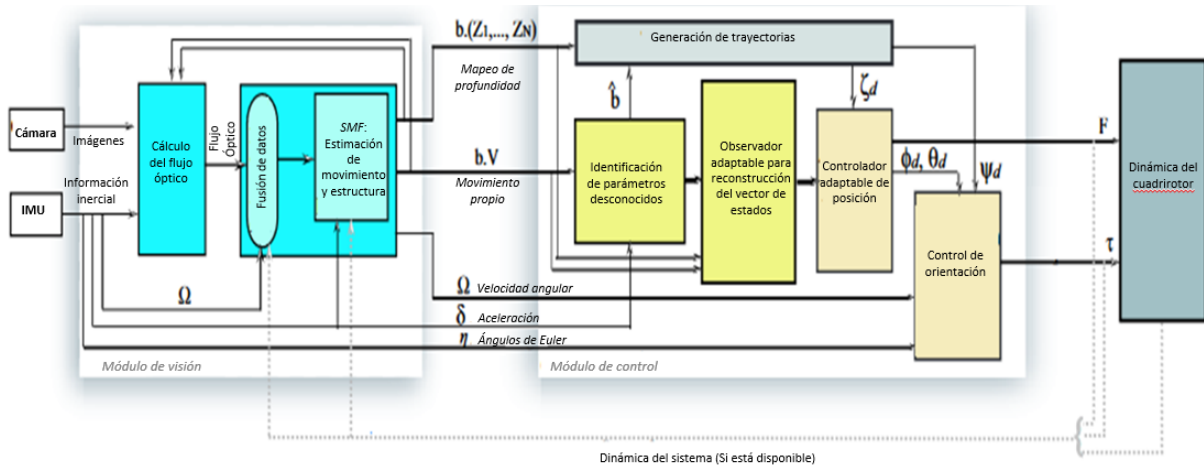


Fig. 3.2. Control 3NKF Nested Kalman Filter basado en flujo óptico

Esta propuesta contiene 3 filtros de Kalman anidados, cálculo del flujo óptico como filtrado, fusión de sensores para la interpretación flujo óptico, y un tercer nivel que hace una función de observación, obteniendo estados estimados del movimiento del vehículo y el mapeo de su ambiente. La finalidad principal de esta configuración es afrontar los problemas derivados de la orientación del cuadricóptero respecto del plano XY, que puede ser interpretado como un desplazamiento si no son consideradas las fuerzas en el vehículo.

Su implementación comprende hardware comercial de un costo en conjunto de 3500 dolares, siendo los elementos principales una estructura X-3D-BL de *Ascending Techonologies*, de 53cm de longitud total, un peso de 400 gramos con batería y capacidad de carga de 300gr, la cual consiste en una computadora Gumstix de 400 MHz con procesador Marvell PXA255, el sensor MNAV100CA de gumstix que incluye una IMU, receptor, y sensor de presión, entregando información al procesador de 3 giroscopos, 3 magnetómetros a una velocidad de 50Hz, se lee posición GPS a 4Hz,

La comunicación con los sensores se envía por RS-232, además se incluye tarjeta wifi wifilix con un ancho de banda de 50Mbps/s, se incluye una cámara análoga con salida NTFS que transmite su video en una frecuencia de 1.3GHz, enviando imágenes de 320x240 pixeles a 25Hz, la cámara va montada en la parte inferior con dirección hacia abajo. El procesamiento visual se realiza en una computadora remota a 10Hz, el flujo óptico calculado regresa a la computadora de a bordo a través de wifi. El software se desarrolla como un conjunto de tareas que son independientes para dar flexibilidad.

3.1.1. Controlador PD en cascada

Se utiliza la arquitectura el controlador presentada en (Corke, 2011) donde un lazo PD externo obtiene el error de posición entre la deseada p^* y la posición actual p , expresadas en el marco de referencia v y el error en velocidad, asumiendo que la referencia en velocidad es 0 en todo momento.

En el caso del ángulo pitch para generar un desplazamiento de la posición actual p_x a la deseada

$$p_x^* = p_{xd}$$

Para obtener la ley de control partimos del torque requerido, el cual en un controlador PD aplicado directamente al ángulo pitch y a su primera derivada

$$\tau_y = K_p (\theta_p^* - \theta_p) + K_d (\dot{\theta}_p^* - \dot{\theta}_p)$$

A partir del hecho de que, en el marco de referencia, para mover el vehiculo se relacionan las fuerzas generadas directamente con los angulos de Euler por:

$$f = R_y(\theta_p) \begin{pmatrix} 0 \\ 0 \\ T \end{pmatrix} = \begin{pmatrix} T \sin \theta_p \\ 0 \\ T \cos \theta_p \end{pmatrix}$$

Se puede utilizar la componente en el eje x

$$f_x = T \sin \theta_p \approx T \theta_p$$

Y la velocidad controlada por una ganancia proporcional

$$f_x^* = m K_1 (v_x^* - v_x)$$

Combinamos las ecuaciones para obtener los ángulos deseados

$$\theta_p^* = \frac{m}{T} K_f (v_x^* - v_x)$$

El control para el ángulo pitch entonces quedaría descrito por la siguiente ley:

$$\theta_p^* = k_1 ({}^v p_x^* - {}^v p_x - k_2 {}^v v_x)$$

La salida de este lazo externo son los ángulos deseados puesto que el error en cada eje es una distancia y tomando 1 segundo como un tiempo arbitrario para cubrirla $p^* - p$ es la velocidad deseada.

De esta forma se obtienen los ángulos deseados a través de la relación entre la fuerza generada en cada eje y la velocidad.

3.1.2. Controlador backstepping

Con el vector de entradas $U = [U1; U2; U3; U4]^T$ y el vector de estados

$$X = [\phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}, z, \dot{z}, x, \dot{x}, y, \dot{y}]^T = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}]^T$$

El control rotacional se diseña a partir del error de seguimiento dado por

$$z_1 = x_{1d} - x_1$$

Donde x_1 y x_{1d} representan el valor del ángulo ϕ y su valor deseado respectivamente.

Se elige una función de Lyapunov $V(z_1)$ definida positiva con su derivada definida negativa para que el termino z_1 converja a cero cuando el tiempo tienda a infinito

$$V(z_1)$$

$$\dot{V}(z_1) = z_1 \dot{z}_1$$

Se puede igualar

$$\dot{z}_1 = \dot{x}_{1d} - \dot{x}_1$$

$$= \dot{x}_{1d} - x_2$$

Por lo tanto reescribiendo la derivada de la función de Lyapunov se tiene

$$V(z_1) = z_1(\dot{x}_{1d} - x_2)$$

La estabilización de estabilización de z_1 se puede obtener introduciendo una entrada virtual x_2 de la forma:

$$x_2 = \dot{x}_{1d} + \alpha_1 z_1 + \lambda_1 \Lambda_1$$

Donde α_1 es una constante positiva que determina la velocidad de convergencia del error, λ_1 es una constante integral positiva y $\Lambda_1 = \int_0^t z_1(\tau) d\tau$. La acción integral elimina el error en estado estable.

Sustituyendo

$$V(z_1) = -\alpha_1 z_1^2 - \lambda_1 \Lambda_1$$

Si $\alpha_1 > 0$ y $\lambda_1 > 0$ entonces $V(z_1)$ es semidefinida negativa.

Si se elige x_{2d} igual a x_2 entonces, se tiene el error de seguimiento

$$z_2 = x_{2d} - x_2$$

Para obtener x_{1d} y x_{2d} , es necesario que z_1 y z_2 converjan a cero, para lo que se propone la siguiente función de Lyapunov

$$V(z_1, z_2) = \lambda_1 \frac{1}{2} (\Lambda_1^2 + z_1^2 + z_2^2)$$

Cuya derivada es

$$\dot{z}_2 = \ddot{x}_{1d} + \alpha_1 \dot{z}_1 + \lambda_1 z_1 - \dot{x}_2$$

Sustituyendo \dot{z}_1 obtenido previamente, se tiene

$$z_2 = \dot{z}_1 + \alpha_1 z_1 + \lambda_1 \Lambda_1$$

De donde se puede despejar \dot{z}_1

$$\dot{z}_1 = z_2 - \alpha_1 z_1 - \lambda_1 \Lambda_1$$

Ahora, sustituyendo \dot{z}_1 y \dot{z}_2 en la derivada de la función de Lyapunov, se obtiene

$$\begin{aligned} V(z_1, z_2) &= z_1 \lambda_1 \Lambda_1 + z_1 z_1 + z_2 z_2 \\ &= z_1 \lambda_1 \Lambda_1 + z_1 (z_2 - \alpha_1 z_1 - \lambda_1 \Lambda_1) + z_2 (x_{1d} + \alpha_1 (z_2 - \alpha_1 z_1 - \lambda_1 \Lambda_1) + \lambda_1 z_1 - x_2) \end{aligned}$$

Recordando que \dot{x}_2 es una dinámica del multirroto dada por: $x_2 = x_4 x_6 a_1 + x_4 a_2 \Omega_r + b_1 U_2$

$$V(z_1, z_2) = z_1 \lambda_1 \Lambda_1 + z_1 (z_2 - \alpha_1 z_1 - \lambda_1 \Lambda_1) + z_2 (x_{1d} + \alpha_1 (z_2 - \alpha_1 z_1 - \lambda_1 \Lambda_1) + \lambda_1 z_1 - x_4 x_6 a_1 + x_4 a_2 \Omega_r + b_1 U_2)$$

Que simplificando nos da

$$V(z_1, z_2) = z_1 z_2 - z_1^2 \alpha_1 + z_2 x_{1d} + z_2^2 \alpha_1 - \alpha_1^2 z_1 z_2 - a_1 z_2 \lambda_1 \Lambda_1 + \lambda_1 z_1 z_2 - x_4 x_6 a_1 - x_4 a_2 \Omega_r - b_1 U_2$$

Donde U_2 representa la entrada de control.

Por lo tanto, diseñando una U_2 para que $V(z_1, z_2)$ sea semidefinida negativa y los términos de error de seguimiento converjan a cero:

$$U_2 = \frac{1}{b_1} \left[z_1 (1 - \alpha_1^2 + \lambda_1) + z_2 (\alpha_1 + \alpha_2) - \alpha_1 \lambda_1 \Lambda_1 + x_{1d} - x_4 x_6 a_1 - x_4 a_2 \Omega_r \right]$$

Donde el término $a_2 z_2$ con $a_2 > 0$ se agrega para estabilizar z_1

Finalmente, la derivada de la función de Lyapunov es definida negativa, garantizando la convergencia a cero de los términos de error z_1 y z_2

$$\dot{V}(z_1, z_2) = -\alpha_1 z_1^2 - a_2 z_2^2 < 0$$

Repitiendo el procedimiento backstepping a los dos movimientos angulares restantes se obtienen las entradas de control:

$$U_3 = \frac{1}{b_2} [z_3 (1 - a_3^2)]$$

$$U_3 = \frac{1}{b_2} [z_3 (1 - \alpha_3^2 + \lambda_2) + z_4 (\alpha_3 + \alpha_4) - \alpha_3 \lambda_2 \Lambda_2 + x_{3d} - x_2 x_6 a_3 - x_2 a_4 \Omega_r]$$

$$U_4 = \frac{1}{b_3} [z_5 (1 - \alpha_5^2 + \lambda_3) + z_6 (\alpha_5 + \alpha_6) - \alpha_5 \lambda_3 \Lambda_3 + x_{5d} - x_4 x_2 a_5]$$

Con

$$z_3 = x_{3d} - x_3$$

$$z_4 = x_{3d} + \alpha_3 z_3 + \lambda_2 \Lambda_2 - x_4$$

$$z_5 = x_{5d} - x_5$$

$$z_6 = \dot{x}_{5d} + \alpha_5 z_5 + \lambda_3 \Lambda_3 - x_6$$

Con lo que queda desarrollado el control rotacional de multirroto.

3.2. Estimación de orientación

Se tienen un conjunto de sensores que nos dan información en los dos marcos de referencia, el acelerómetro y el magnetómetro miden cambios físicos respecto al sistema inercial, de forma que sólo el sensor ultrasónico y el giróscopo miden utilizando el sistema de referencia del hexarrotor, por lo tanto es necesario un sistema que permita transformar las magnitudes vectoriales de un sistema a otro y operar sobre ellas para obtener la orientación del vehículo.

Las soluciones que se han utilizado en algunos proyectos son filtro de Kalman extendido, filtro complementario lineal, filtro complementario no lineal y algoritmos de matriz de cosenos directores DCM (*Direction Cosine Matrix*). Además como se ha mencionado anteriormente existen algunas unidades de medición inercial (IMU) que incluyen una etapa de procesamiento capaz de realizar la estimación de orientación devolviendo ya sea ángulos de Euler o cuaterniones.

3.2.1. Algoritmos DCM (Matriz de Cosenos Directores)

La Matriz de Cosenos Directores (DCM), es una matriz de 3x3 que se conforma por las proyecciones de los vectores unitarios del sistema de referencia inercial en el sistema de referencia del cuerpo rígido o inversamente, Esta se utiliza para transformar un vector de un marco referencial a otro, y un algoritmo para su cálculo es parte de la implementación realizada.

La estructura del algoritmo incluye controles PI generalmente para eliminar derivas en los ángulos calculados, dejando una estructura similar a la representada en el diagrama de bloques de la **¡Error! No se encuentra el origen de la referencia..**

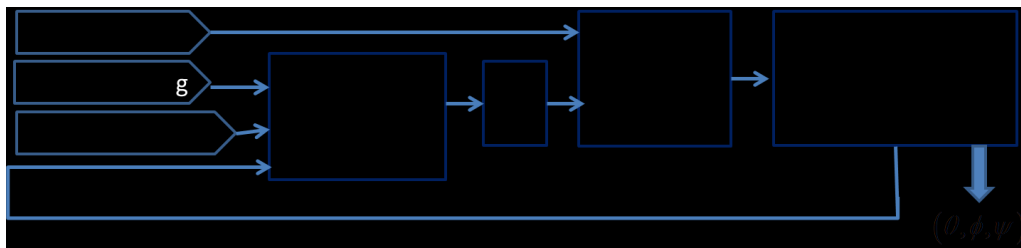


Fig. 3.3. Estructura del algoritmo DCM

Para explicar este algoritmo se introduce una notación distinta, y dos sistemas coordenados, el primero generado por los vectores ortogonales (I, J, K) denominado G y el segundo H, por (i, j, k) denotado

por B, como se muestra en la Fig. 3.4.. Se puede considerar el sistema coordenado H, como el sujeto a un cuerpo que gira sobre los ejes del sistema G y se busca obtener una matriz de cosenos directores que permita simplificar tales rotaciones y su representación.

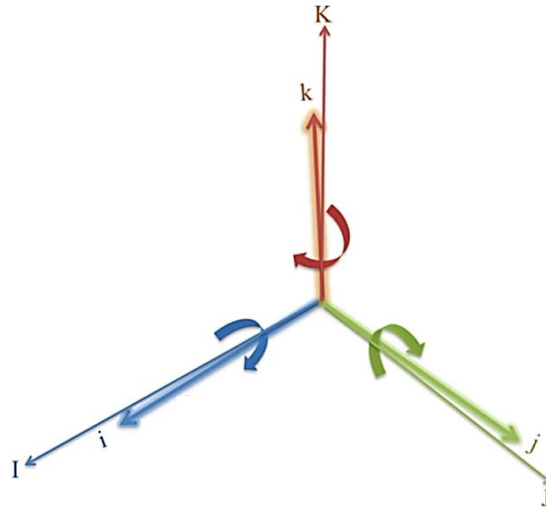


Fig. 3.4. Sistemas coordenados

Haciendo coincidir los dos sistemas coordenados, y considerando los vectores $A = (A_x, A_y, A_z)$ como la lectura del acelerómetro y $M = (M_x, M_y, M_z)$ la lectura del magnetómetro, se realiza el análisis para encontrar la DCM.

En estado estable el acelerómetro medirá la fuerza de gravedad con dirección hacia el centro de la tierra por lo que:

$$K^B = -A$$

El magnetómetro de forma similar medirá la intensidad del campo magnético al que está sujeto por efecto de los polos magnéticos de la tierra, con dirección al norte.

$$I^B = M$$

Se tiene que se obtienen dos vectores que por convención (NED) se asumen perpendiculares, y permiten la obtención de un tercero perpendicular a ellos a través de un producto cruz.

$$J^B = K^B \times I^B$$

Con estos vectores se puede formar una matriz de cosenos directores como la mostrada en la ec. (2.1) que permite describir rotaciones de una forma simplificada.

$$DCM^{BT} = DCM^G = [I^B, J^B, K^B]$$

$$DCM^G = [i^G, j^G, k^G] = \begin{bmatrix} I \cdot i & I \cdot j & I \cdot k \\ J \cdot i & J \cdot j & J \cdot k \\ K \cdot i & K \cdot j & K \cdot k \end{bmatrix} = \begin{bmatrix} \cos(I, i) & \cos(I, j) & \cos(I, k) \\ \cos(J, i) & \cos(J, j) & \cos(J, k) \\ \cos(K, i) & \cos(K, j) & \cos(K, k) \end{bmatrix} = \begin{bmatrix} I^{BT} \\ J^{BT} \\ K^{BT} \end{bmatrix} \quad (2.1)$$

Pero la estimación obtenida es susceptible de campos magnéticos diferentes al de la tierra y a aceleraciones que eliminen el efecto de la gravedad, por lo tanto es necesario agregar una compensación utilizando el giróscopo, ya que este último mide la velocidad angular sin recibir perturbaciones externas.

Otra ventaja que ofrece el giróscopo es la capacidad de estimar la orientación ante incrementos en los ángulos de giro siempre y cuando sean de una variación pequeña en un tiempo conocido.

Ahora vamos a demostrar que es posible obtener el valor de un vector K_1 que es rotado durante dt segundos conociendo el estado anterior y midiendo la velocidad angular a través de un acelerómetro:

Definiendo v como velocidad lineal del vector y w_g la velocidad angular medida por el giróscopo, la cual para un periodo muy corto de tiempo se aproxima a $d\theta_g$ que es el desplazamiento angular o giro realizado se tiene:

$$\begin{aligned} K_{1G}^B &= K_0^B + dtv \\ &= K_0^B + dt(w_g \times K_0^B) \\ &= K_0^B + (d\theta_g \times K_0^B) \end{aligned}$$

Es decir que se demostró la hipótesis siempre y cuando la velocidad angular se aproxime al desplazamiento angular, de forma que el tiempo debe tender a cero.

Es entonces posible utilizar de la misma forma el acelerómetro, haciendo el procedimiento de forma inversa para obtener del vector medido el ángulo de giro:

$$\begin{aligned} d\theta_a &= dtw_a \\ w_a &= \frac{K_0^B \times v_a}{|K_0^B|^2} \end{aligned}$$

Donde $v_a = (K_0^B - K_{1A}^B) / dt$, es la velocidad lineal del vector, sustituyendo v_a y como $|K_0^B|^2 = 1$, se tiene:

$$d\theta_a = dtw_a = K_0^B \times (K_{1A}^B - K_0^B)$$

Por último es conveniente añadir una estimación similar pero trabajando con el vector I_B que apunta al norte y la medición dada por el magnetómetro: $d\theta_m = dtw_m = I_0^B \times (I_{1M}^B - I_0^B)$

De esta forma se tienen 3 estimaciones distintas del giro y este puede ser representado como el promedio ponderado de las 3.

$$d\theta = \frac{(s_a d\theta_a + s_g d\theta_g + s_m d\theta_m)}{s_a + s_g + s_m}$$

De esta forma se ha disminuido la susceptibilidad al ruido, y se pueden expresar los vectores necesarios para obtener la matriz de cosenos directores después de dt segundos como:

$$\begin{aligned} K_1^B &= K_0^B + (d\theta \times K_0^B) \\ I_1^B &= I_0^B + (d\theta \times I_0^B) \\ J_1^B &= J_0^B + (d\theta \times J_0^B) \end{aligned} \quad (2.2)$$

Pero los vectores generados no son ortogonales por su formulación, para ajustarlos se realiza un proceso conocido como renormalización, que consiste en compensar el error producido por cualquier ruido inducido y mantener lo máximo posible la ortogonalidad de los vectores unitarios calculados, por ejemplo trabajando con el plano KI su producto cruz genera un vector Jn ortogonal a los dos anteriores, si se deja fijo K y haciendo producto cruz con Jn se obtiene un nuevo In ortogonal a K .

3.3. Estimación de posición

Se utilizan para estimación relativa del desplazamiento de vehículos multirrotores principalmente sistemas de visión, o incluso sistemas de captura de movimiento externo de mayor frecuencia de muestreo. Existen principalmente dos sistemas que generan un costo computacional menor que los que incluyen mapeo, estos son la odometría visual y el flujo óptico, y han demostrado ser capaces de trabajar en línea o en estaciones de control remotas.

3.3.1. Flujo óptico

El flujo óptico es el patrón de movimiento aparente de objetos, superficies y bordes en una escena causada por el movimiento relativo con el punto de observación. Su aplicación en vehículos multirrotores es para calcular el desplazamiento relativo, integrando las velocidades calculadas y compensando ángulos de orientación y altura, incluso ya se ha desarrollado un sensor integrado para tal fin, una desventaja con respecto a la odometría visual es que no utiliza parámetros de la cámara y requiere información adicional. Una prueba realizada a los conjuntos de imágenes reales se muestra en la sección de resultados.

Considérese un punto en el marco de referencia de la cámara, $X_c = (x_c, y_c, z_c)$ capturado en una imagen $I_t(X_c, t)$ de dos dimensiones, asumiendo que la intensidad de las imágenes es constante, es posible escribir:

$$f(x_c, y_c, z_c) = f(x_c + dx_c, y_c + dy_c, t + dt) \quad (2.3)$$

Donde dx_c y dy_c son el movimiento del punto durante dt

La formulación de esta técnica consiste en encontrar una función que relaciona en el tiempo dos imágenes de una misma escena considerando desplazamientos pequeños de los puntos en la misma a través de la consideración de que el brillo permanece constante (*Brightness constancy assumption*) y el tiempo de adquisición entre las mismas es corto, la cual se muestra en la ec. (2.3).

Después se obtiene la serie de Taylor que aproxime la función del término a la derecha de la ec.

$$f(x_c, y_c, z_c) = f(x_c, y_c, z_c) + \frac{\delta f}{\delta x_c} dx_c + \frac{\delta f}{\delta y_c} dy_c + \frac{\delta f}{\delta t} dt$$

Ahora se hace un cambio de notación donde $\frac{\delta f}{\delta i} = f_i$ y cancelando el término al moverlo al otro lado de la ecuación se tiene:

$$f_x dx + f_y dy + f_t dt = 0$$

Dividiendo toda la ecuación por dt se obtiene la ecuación que representa el flujo óptico.

$$f_x u + f_y v + f_t = 0$$

Es posible llegar a una interpretación geométrica de esta ecuación despejando $v = -\frac{f_x}{f_y} u - \frac{f_t}{f_y}$ que permite representar una recta en un plano (u, v) , como se muestra en la fig. (A3.1)

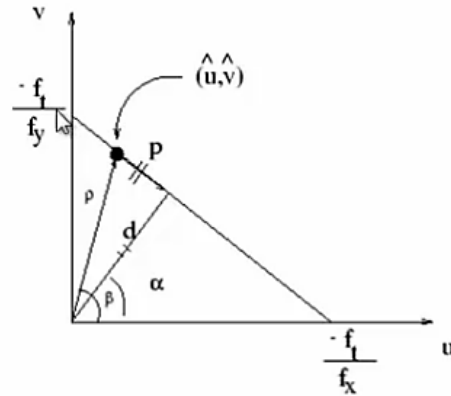


Fig. 3.5. Interpretación geométrica de ecuación de flujo óptico

Esta línea pasa por dos puntos que pueden ser conocidos, pero considerando que se tiene una ecuación y dos variables, se puede proceder imaginando un punto (\hat{u}, \hat{v}) y usando trigonometría, para obtener dos vectores, el flujo paralelo p y el flujo normal $d = \frac{f_t}{\sqrt{f_x^2 + f_y^2}}$. Es notable ahora que para todos los casos de puntos el flujo normal será el mismo.

Para llegar a un método de solución de la ecuación dada se puede hacer referencia al método de optimización de Horn&Schunck, el cual relaciona la constancia del brillo y un criterio de suavidad, la cual se muestra en la ecuación A3.2.

$$\iint \left\{ (f_x u + f_y v + f_t)^2 + \lambda (u_x^2 + u_y^2 + v_x^2 + v_y^2) \right\} dx dy \quad (A3.2)$$

La expresión propuesta tiene una solución por cálculo variacional que lleva a:

$$(f_x u + f_y v + f_t) f_x + \lambda (\Delta^2 u) = 0$$

$$(f_x u + f_y v + f_t) f_y + \lambda (\Delta^2 v) = 0$$

La forma de obtener las derivadas en las imágenes se utilizan máscaras, las cuales se presentan en la tabla A3.1:

Tabla 3.1. Máscaras derivativas y Laplaciano

f_x	f_y	f_t	Laplaciano $f_{xx} + f_{yy}$
$\begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}$ Primer imagen	$\begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix}$ Primer imagen	$\begin{bmatrix} -1 & -1 \\ -1 & -1 \end{bmatrix}$ Primer imagen	$\begin{bmatrix} 0 & -\frac{1}{4} & 0 \\ -\frac{1}{4} & 1 & -\frac{1}{4} \\ 0 & -\frac{1}{4} & 0 \end{bmatrix}$
$\begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}$ Segunda imagen	$\begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}$ Segunda imagen	$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ Segunda imagen	

Observando que $\lambda(\Delta^2 v) = 0$ se aproximó a $f_{xx} + f_{yy} = f - f_{av}$, su forma discreta es:

$$(f_x u + f_y v + f_t) f_x + \lambda(u - u_{av}) = 0$$

$$(f_x u + f_y v + f_t) f_y + \lambda(v - v_{av}) = 0$$

Por lo tanto con dos ecuaciones y dos variables, las soluciones a las ecuaciones, que representan la velocidad relativa entre dos imágenes son:

$$\begin{aligned} u &= u_{av} - f_x \frac{P}{D} \\ v &= v_{av} - f_y \frac{P}{D} \end{aligned} \tag{A3.3}$$

Donde

$$P = f_x u_{av} + f_y v_{av} + f_t$$

$$D = \lambda + f_x^2 + f_y^2$$

3.3.2. Odometría visual

Es una técnica visual de medición de desplazamiento relativo que se basa en crear transformaciones homográficas entre el plano en el mundo real capturado por una cámara utilizando sus parámetros internos y las dimensiones medidas en píxeles, de forma que puede a través de matrices de transformación obtener información del mundo real.

El primer desarrollo práctico de la odometría visual se atribuye a Hans Moravec en 1980, quien en su tesis doctoral utiliza una cámara móvil para obtener el cálculo del movimiento de un vehículo terrestre a través de los cambios en las imágenes obtenidas (Moravec, 1980), para esto inventó un filtro predecesor del detector Harris, el cual es actualmente el mayormente usado para detectar puntos de referencia característicos como bordes o esquinas.

Los siguientes veinte años la Administración Nacional de la Aeronáutica y del Espacio *NASA* y el Laboratorio de Propulsión de Jets *JPL*. Lideran la investigación hasta lograr en 2004 que la odometría visual sea utilizada en los vehículos de Exploración Mars rovers Spirit y Opportunity en Marte.

Fig. 3.6. Vehículo Mars rover

Se han desarrollado dos implementaciones del algoritmo principalmente, la odometría visual binocular y la odometría visual monocular, donde la primera aprovecha las mediciones adicionales que se pueden obtener conocidas las distancia y configuraciones de las dos cámaras sobre el vehículo para reducir el

error, aunque los resultados de la odometría monocular han sido bastante cercanos el procesamiento es mayor, y el error acumulativo también es mayor.

3.3.2.1. Formulación del problema

Un agente dotado de una cámara fija se mueve en un ambiente tomando imágenes a un tiempo de muestreo discreto.

$$I_{0:n} = [I_0, \dots, I_n]$$

Dos posiciones de la cámara en instantes de muestreo adyacentes están relacionadas por una transformación de cuerpo rígido:

$$T_k = \begin{bmatrix} R_{k,k-1} & t_{k,k-1} \\ 0 & 1 \end{bmatrix}$$

Si se conserva un conjunto de las transformaciones de movimiento, se tendrá el conjunto de posiciones con respecto al marco de referencia inicial. La posición actual C_n puede ser calculada concatenando las transformaciones t_k conociendo la posición inicial C_0 .

$$C_n = C_{n-1}T_n$$

El problema principal de VO es calcular las transformaciones relativas T_k de las imágenes I_k e I_{k-1} y luego concatenar las transformaciones para recuperar la trayectoria completa $C_{0:n}$ de la cámara.

3.3.2.2. Modelo de la cámara y calibración

Para la representación en píxeles de una imagen, se puede utilizar el modelo de proyección central, que consiste en asumir que existe un punto principal que une el origen de la cámara y el punto central del plano de la imagen a través del eje óptico y un punto en el mundo real, esta línea que es ortogonal al plano de la imagen como se muestra en la. Es preferible para la representación de líneas, representar en coordenadas homogéneas y establecer la relación de la ecuación

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (\text{A2.1})$$

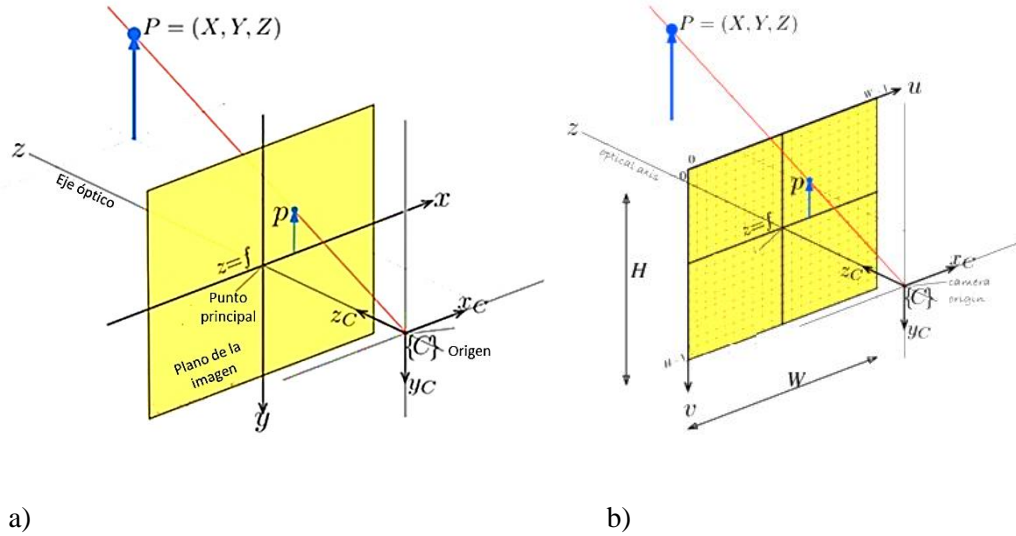


Fig. 3.7. Modelo de proyección central y coordenadas

Para obtener una transformación entre las dimensiones en la imagen a pixeles, se tiene que recorrer el origen de la cámara a la esquina superior izquierda, como se muestra en la fig. A2.1. Además se debe encontrar la relación entre las dimensiones de los puntos de la imagen y los pixeles, la cual se muestra en la ec. A2.2.

$$\begin{pmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{pmatrix} = \begin{pmatrix} \frac{1}{\rho_u} & 0 & 0 \\ 0 & \frac{1}{\rho_v} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{pmatrix} \quad (\text{A2.2})$$

Donde un punto en la imagen se representa en pixeles por: $p = \begin{pmatrix} \tilde{u} \\ \tilde{v} \end{pmatrix} = \begin{pmatrix} \tilde{u}/\tilde{w} \\ \tilde{v}/\tilde{w} \end{pmatrix}$

De forma que se puede crear una matriz que depende solamente de parámetros internos de la cámara y se conoce como de parámetros intrínsecos, pero hace falta añadir la orientación de la cámara a través de una matriz de rotación R y vector de traslación t . El producto de estas tres matrices se conoce como matriz de la cámara que se muestra en la ec. A2.3, y permite pasar de un punto en el espacio a un punto en pixeles, y es la base de la odometría visual.

$$\begin{pmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ \rho_u & & \\ 0 & 1 & 0 \\ & \rho_v & \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f_l & 0 & 0 & 0 \\ 0 & f_l & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \left[\begin{pmatrix} R & t \\ 0_{1 \times 3} & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \right] \quad (\text{A2.3})$$

Por la naturaleza de invariancia ante escalamiento, y para reducir los términos a estimar, se lleva a una forma reducida.

$$\begin{pmatrix} \tilde{u} \\ \tilde{v} \\ 1 \end{pmatrix} = \begin{pmatrix} f_l & 0 & cu \\ 0 & f_l & cv \\ 0 & 0 & 1 \end{pmatrix} \left[(R(r) \ t) \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} - \begin{pmatrix} s \\ 0 \\ 0 \end{pmatrix} \right] \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (\text{A2.3})$$

Donde:

$(u \ v \ 1)^T$ Coordenadas homogéneas en la imagen

f_l Longitud focal

$R(r) = R_x(r_x)R_y(r_y)R_z(r_z)$ Matriz de rotación

$$t = (t_x \quad t_y \quad t_z)^T$$

Vector de traslación

$$X = (x \quad y \quad z)^T$$

Coordenadas tridimensionales

$$s = 0$$

Desplazamiento

También es posible demostrar que ya que los puntos en una imagen pertenecen a un mismo plano, se puede obtener una relación homográfica planar, por la cual resulta más fácil calcular una sola matriz homográfica a través de geometría epipolar, y mediante operaciones matemáticas obtener para un par de imágenes el vector de traslación. Es por tanto necesario conocer los parámetros intrínsecos de la cámara para que la estimación sea válida.

Capítulo 4.

Implementación de una arquitectura para control de hexarrotor

4.1. Descripción de las partes de un multirrotor

4.1.1. Motores BL-DC y hélices

Un motor Brushless de corriente directa (BL-DC) es una máquina síncrona de imanes permanentes con campos magnéticos distribuidos uniformemente en el entrehierro de tal modo que cuando el motor gira a una velocidad constante la fuerza contra-electromotriz tiene una forma trapezoidal y presenta un desempeño similar a un motor de DC. Debido a que los imanes permanentes no requieren fuentes externas de electricidad es posible colocar la armadura del motor en el estator a lo que se denomina *outrunner*, es decir el rotor se encuentra externamente y está constituido de imanes permanentes. Las ventajas que presenta un motor de este tipo son menores pérdidas (no utiliza excitación), mejor enfriamiento y funcionamiento a voltajes mayores que un motor de DC.

En la Fig. 4.1 se muestra la forma de onda de excitación de seis pasos necesaria para lograr el giro completo del motor, y la fuerza contra-electromotriz trapezoidal.

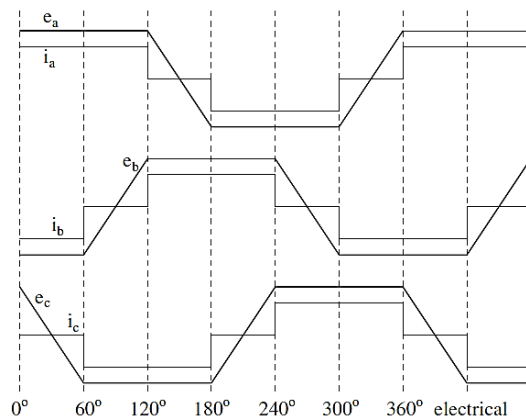


Fig. 4.1. Motor BL-DC, a) BLCD tipo *outrunner* y b) Corriente y voltaje

La fuerza de empuje vertical que produce un motor con una hélice está dada por la siguiente relación cuadrática:

$$f_i = k_f w_i^2 \quad (2.4)$$

Donde f_i es el empuje generado, k_f es una constante adimensional y w_i es la velocidad angular del motor, de forma que adicionalmente se puede incluir o no la dinámica del motor. Para determinar la ganancia k_f se realiza una regresión lineal con el empuje en gramos contra los cuadrados de las velocidades medidas. Algunos modelos incluyen fenómenos propios de las hélices como son el *flapping* cuya presencia no influye considerablemente en la dinámica de un multirrotor.

4.1.2. Tarjeta de control

Es un sistema de procesamiento capaz de leer sensores y generar las salidas necesarias para estabilizar y mover el multirrotor, además puede poseer funciones de telemetría y comunicación para recibir señales de referencia externas. Se han utilizado distintos procesadores para tales fines, los cuales comprenden desde procesadores digitales de señales hasta computadoras con sistemas operativos de tiempo real, en la

Tabla 4.1 se muestran algunos procesadores utilizados en algunos proyectos multirrotos comparando su velocidad de procesamiento.

Tabla 4.1. Tarjetas de control de proyectos multirrotos

Proyecto	Procesador	Frecuencia procesador (MHz)	Memoria Flash (Kbytes)	Puertos de comunicación y digitales
Arducopter	ATmega2560	16 (8 bits)	256	5xSPI, 1xI2C, 86 pines I/O (16 ADC de 10 bits)
Openpilot	STM32F103CB (ARM Cortex-M3)	72 (32 bits)	128	1xCAN, 3xUSART, 2xSPI, 2xI2C, 1xUSB, 16xADC 12 bits, 16 PWM.
Paparazzi	STM32F105RCT6 (ARM Cortex-M3)	60 (32 bits)	256	1xCAN, 3xUSART, 2xSPI, 2xI2C, 1xUSB, 16xADC 12 bits, 16 PWM.
Pixhawk	LPC2148	60 (32 bits)	512	14xADC 10 bits, 1xDAC 10 bits, SPI, 1xUSB, 2xUART, 2xI2C, 6xPWM
Mikrokopter	ATmega644	20 (8 bits)	128	5xSPI, 1xI2C, 86 pines I/O (16xADC 10 bits)
KKmulticopter	ATmega168	20 (8 bits)	128	5xSPI, 1xI2C, 86 pines I/O (16xADC 10 bits)

Multiwii	Arduino Board	8-20 (8 bits)	256	5xSPI, 1xI2C, 86 pines I/O (16xADC 10 bits)
Aeroquad	Arduino Board	8-20 (8 bits)	256	5xSPI, 1xI2C, 86 pines I/O (16xADC 10 bits)
X-Flyer	TMS320F2812 DSP	150 (32 bits)	256	1xSPI, 2xUART, 1xCAN, 56 GPIO (16xADC 12 bits)

4.1.3. Sensores convencionales y de percepción de entorno

4.1.3.1. Sensores convencionales:

- **Giróscopo:** Mide la velocidad angular de un sistema en el marco de referencia inercial
- **Acelerómetro:** Mide la aceleración lineal de un sistema en el marco de referencia inercial
Su principio de funcionamiento es medir la diferencia en la aceleración lineal del sensor respecto del campo gravitacional local, en la Fig. 4.2 se muestra una imagen de microscopio electrónico de un acelerómetro MEMS. Donde una masa se suspende por medio de resortes de amortiguamiento y se mide la capacitancia que se genera entre secciones de la masa y placas fijas.
- **Magnetómetro:** Mide el campo magnético de la tierra y su dirección. El principio de funcionamiento es capacitivo el cual detecta el movimiento de una masa cargada generado por fuerzas magnéticas externas.
- **Unidad de medición inercial (IMU):** Mide y/o estima velocidad, orientación y/o fuerzas gravitacionales, utilizando una combinación de giróscopo, acelerómetro y magnetómetro.

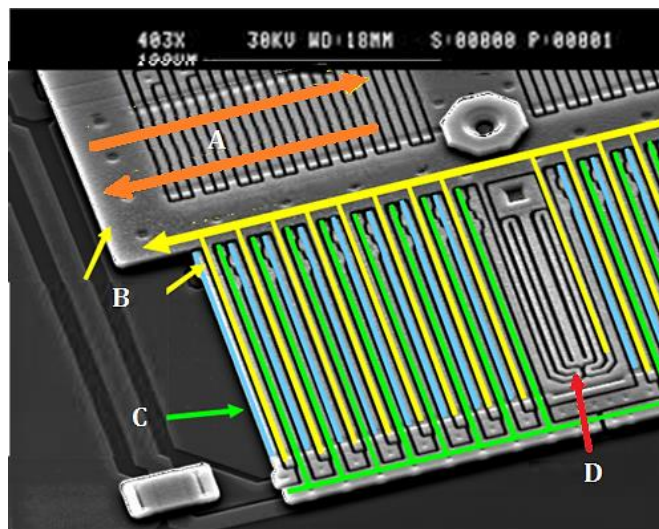


Fig. 4.2. Acelerómetro MEMS: A) Movimiento creado por aceleración o gravedad, B) Masa de prueba C) Placas de sensado, D) Resortes de amortiguamiento

4.1.3.2. Sensores para localización y percepción del ambiente

- Sistemas de posicionamiento global (GPS): Provee la localización absoluta en cualquier lugar de la tierra cuando se encuentra en la línea de visión de cuatro o más satélites.
- Sensor Láser de Profundidad (LRF): Utiliza un rayo láser para determinar la distancia a un objeto, es ampliamente utilizado en aplicaciones de reconocimiento de objetos en 3D.
- Sensor ultrasónico: Genera y emite ondas de alta frecuencia y mide el intervalo entre el envío y su eco para determinar la distancia a un objeto.
- Sensor infrarrojo: Transmite pulsos infrarrojos y mide el tiempo que tardan en regresar para estimar distancia, es de un alcance menor al sensor ultrasónico.
- Sensores de imagen: Los también llamados sensores visuales son pasivos, sin contacto, y de bajo costo, proveen una gran cantidad de información pero introducen un mayor costo computacional.

4.1.4. Controlador electrónico de velocidad (ESC)

Es un inversor electrónico que controla el giro del motor brushless al recibir una señal PWM, la cual tiene un periodo de 20 mS y de un ancho de pulso entre 1 y 2 mS, esta señal, es llamada comúnmente señal servo porque se utiliza para controlar tales dispositivos, a través de la esta señal se fija un bus de voltaje que varía teniendo como máximo el voltaje entregado por la batería.

En la práctica la señal servo solamente regula un nivel de voltaje para la etapa de potencia es decir controla indirectamente la velocidad, por lo que se han empleado métodos de medición de velocidad y/o posición a través de sensores de efecto hall entre otros para crear un control más preciso (Gonzales, Salazar, Romero, Lozano, & Torres). Este tipo de implementación tiene un costo mayor al incluir sensores, además de las modificaciones físicas necesarias para la colocación del mismo, se puede observar en la Fig. 4.3 (a) un ejemplo de este tipo de control.

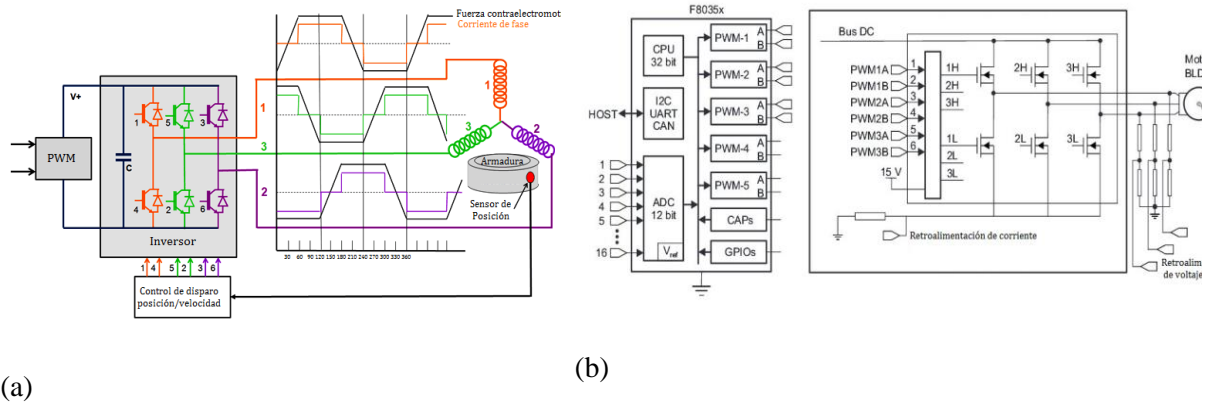


Fig. 4.3. Control de posición utilizando sensor de efecto hall

Una desventaja de la retroalimentación por sensor de efecto hall es que a velocidades bajas no se obtiene una señal clara y esto puede dificultar el control, los ESC deben incluir un filtro pasa bajas que ayude a incrementar la amplitud de la señal de forma que se pueda obtener una señal de posición válida. Por otro lado se puede evitar la instalación de sensores adicionales y utilizar un ESC que tenga una técnica *sensorless* para controlar la conmutación interna de transistores, midiendo sólo el voltaje en las tres fases, y se utiliza como retroalimentación para el control de posición, como se puede observar en la Fig. 4.3. (b).

4.2. Arquitectura de control

Se desarrolló una estructura de control jerárquico añadiendo la etapa de nivel superior, la computadora encargada del procesamiento de imágenes se encuentra a bordo del hexarrotor, esta obtiene del sensor RGB-D una imagen cada 33mS y en aproximadamente 41mS (24Hz) realiza el procesamiento necesario para obtener una estimación de desplazamiento relativo a través de un algoritmo de odometría visual (el tiempo se puede disminuir si no se muestra la imagen que se está procesando hasta los 31mS). A través de una función temporizada esta computadora envía a Arduino referencias angulares para generar desplazamiento.

Una configuración similar se muestra en (Huang, y otros, 2011) donde se utiliza un PID de posición utilizando estimaciones entregadas por un filtro de Kalman que combina odometría visual y un algoritmo de estimación de posición a partir de una IMU. En dicho trabajo el procesamiento se realiza a bordo con una velocidad de procesamiento 25ms (40Hz), además el uso del sensor Kinect en una tarjeta de 1.86GHz core2duo con 4Gb de RAM

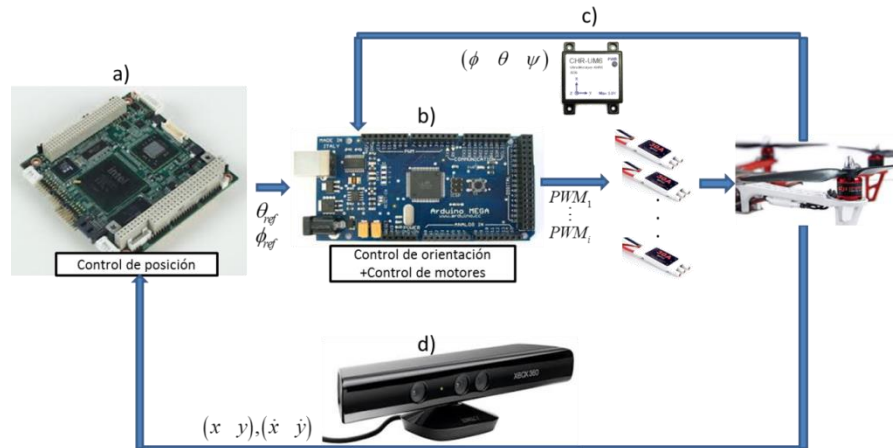


Fig.4.4. Arquitectura de control: a) PCM-3363, b) Arduino Mega 2560, c) Sensor de orientación, d) Sensor RGB-D

4.3. Instrumentación del hexarrotor

El control estabilizados se aloja en una tarjeta Arduino Mega 2560, la cual cuenta con cuatro puertos UART y un puerto para comunicación síncrona (i²c), posee 54 pines digitales de entrada/salida de los cuales catorce manejan PWM, posee un procesador que trabaja a 16Mhz y puede almacenar 256k en memoria flash

La computadora de abordo es una tarjeta Avantech CPM-3363 la cual cuenta con un procesador Intel dual core a 1.8GHz, 1GB de memoria SDRAM, utiliza como disco duro una memoria Flash. Esta tarjeta tiene instalada una versión de Windows 7 sobre el cuál se instaló Matlab 2013a y los controladores necesarios para la tarjeta Arduino y el sensor kinect.

Para facilitar conexiones se diseñó una placa que se introduce en la parte superior del Arduino y proporciona las conexiones para motores utilizando las salidas PWM y una distribución de pines compatible y se incluyen los puertos UART1 e I2C para conectar los sensores necesarios además de pines e interruptores para encendido/apagado y reset.

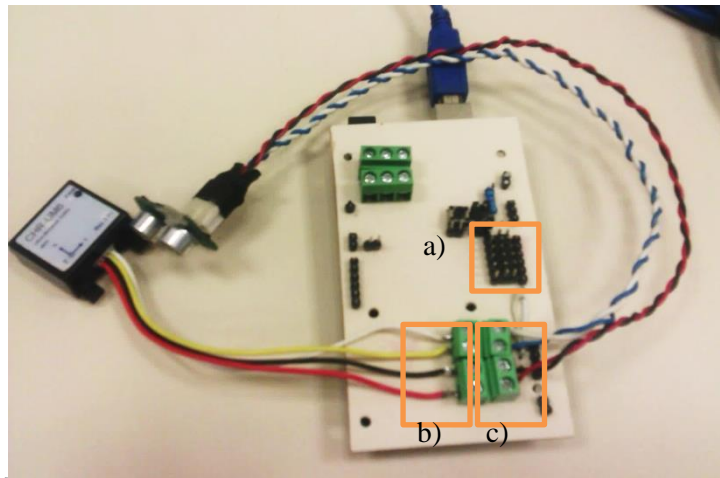


Fig. 4.5. Placa de motores y sensores. a) Salidas PWM, b) Puerto UART1 c) Puerto I²C.

La apariencia final se muestra a continuación, donde también se puede apreciar el sistema operativo funcionando. La tarjeta Advantech y el sensor Kinect son alimentados con la batería a través de dos fuentes de voltaje independientes una de 5 V a 3 A para la PC y otra con dos salidas de 12 V y de 5 V a 2 A. Se agregó un disco duro de estado sólido para almacenar el sistema operativo debido a que la memoria flash con la que cuenta la PC abordo sólo tiene una capacidad de 8Gigabytes.



Fig. 4.6 Sistema Hexarrotor encendido.

4.4. Control estabilizador

4.4.1. Implementación del control en una tarjeta Arduino Mega 2560

La tarjeta Arduino Mega 2560 cuenta con cuatro puertos UART y un puerto para comunicación síncrona (i²c), posee 54 pines digitales de entrada/salida de los cuales catorce manejan PWM, posee un procesador que trabaja a 16Mhz y puede almacenar 256k en memoria flash.

La computadora de abordo es una tarjeta Avantech CPM-3363 la cual cuenta con un procesador Intel dual core a 1.8GHz, 1GB de memoria SDRAM, utiliza como disco duro una memoria Flash. Esta tarjeta soporta Windows 7, Windows XP, Windows CE, WES7 y WES, por lo que se agregó un disco duro de estado sólido para ejecutar Windows 7 y de esta forma utilizar software como MATLAB.

4.4.2. Implementación utilizando mini IMU9

Se utilizó la unidad inercial mini IMU9 de Pololu que combina un acelerómetro de 3 ejes LSM303DLHC y un giroscopio y magnetómetro ST L3GD20 ambos de 3 ejes para formar una unidad de medición inercial (IMU).

Se obtiene la estimación de los ángulos de Euler combinando un giróscopo, acelerómetro y magnetómetro a una velocidad de 50Hz en la tarjeta Arduino utilizando un algoritmo basado en la matriz de cosenos directores, que se explica en la sección de estimación de orientación. En la 4.7 se presenta un diagrama de flujo del control implementado en la tarjeta Arduino:

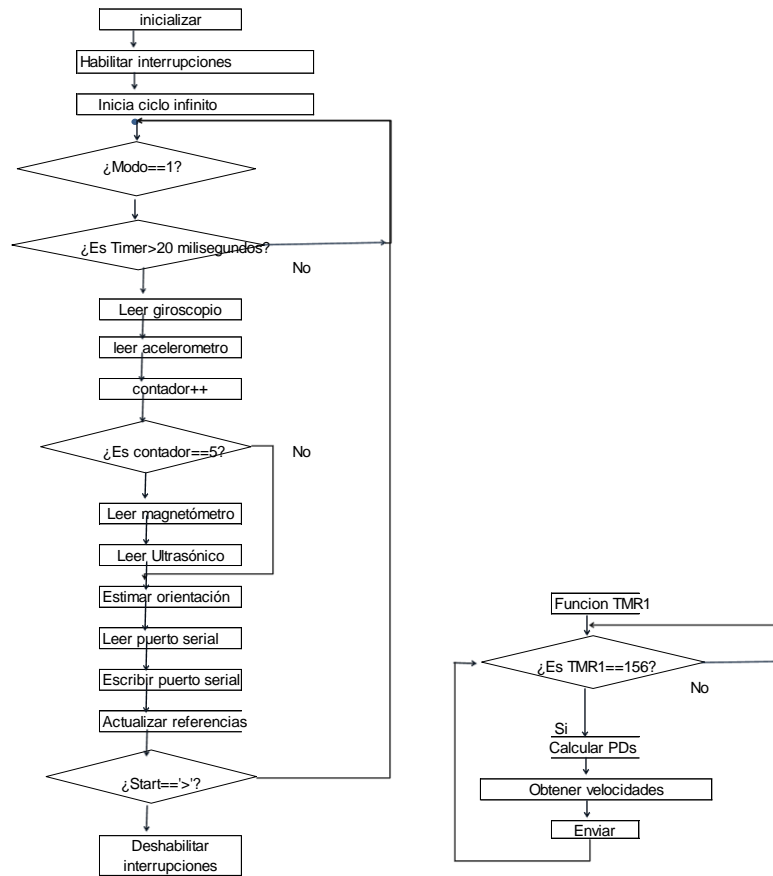


Fig. 4.7. Algoritmo de control y estimación de ángulos de Euler en Arduino para mini imu9

4.4.3. Generación de referencia utilizando Matlab

Se desarrolló un script en Matlab que permite la obtención de imágenes de un dispositivo de imagen, ya sea la cámara de una computadora portátil o el sensor kinect, y obtener en cada periodo de muestreo la estimación de desplazamiento respecto de la imagen anterior y enviar 3 referencias angulares a la tarjeta Arduino. Aunque no fue posible demostrar el seguimiento de trayectorias el script desarrollado permitió la adquisición de los datos mostrados en los resultados de esta tesis.

En la Fig. 4.8 se muestra un diagrama de flujo que explica el funcionamiento del código de Matlab realizado.

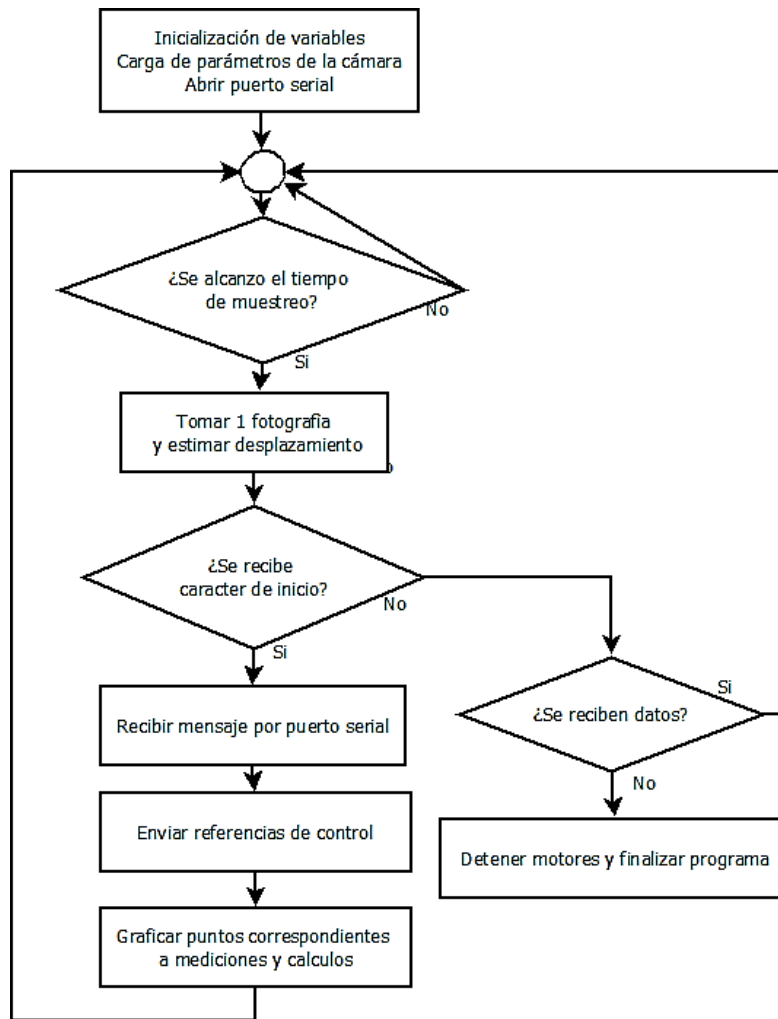


Fig. 4.8. Interfaz de control de posición y estimación de desplazamiento

Para el uso de este programa, es deseable crear un vector de referencias que comprenda las coordenadas en el plano x, y de una trayectoria predefinida, en el desarrollo práctico de esta tesis solo se realiza control para mantener la posición y orientación, por lo que las coordenadas son cero y las referencias angulares en consecuencia también son cero.

4.4.4. Mejoras en la implementación

El sensor UM6 es un sensor de orientación que combina lecturas de giróscopos, acelerómetros y magnetómetros para obtener una medición de orientación a una velocidad de 500Hz, tiene opción para conectarse a un GPS u otros dispositivos a través de comunicación UART o SPI. Tiene dos modos de comunicación, en el modo por defecto transmite los valores de algunos registros habilitados a una velocidad de 115200 baudios, el segundo modo funciona a solicitud por el puerto UART, es decir, se envía una cadena de bytes al sensor, la cual contiene entre otras cosas la dirección de registro que se desea consultar, así se obtienen dos o cuatro bytes de datos en formato 16 bits complemento a dos. Su principal ventaja es que posee un microprocesador interno que realiza estimación de los ángulos de Euler, evitando así costo computacional para el sistema.

El sensor PX4FLOW de 3DRobotics es un sensor de flujo óptico integrado que compensa la medición en píxeles con la medición de altura dada por un sensor ultrasónico y con un giróscopo de 3 ejes compensa a través de un algoritmo de fusión la orientación de un multirrotor, devolviendo la medición de velocidad en los ejes X, Y, y la altura (eje Z).

Se creó una interfaz para cada uno de los sensores en simulink, de esta forma se tienen mediciones más rápidas y que eliminan costo computacional en la tarjeta Arduino, a continuación se muestra en la Fig. 4.9 la interfaz para el sensor UM6 y en la Fig. 4.10 la del sensor PX4FLOW incluyendo la integración discreta necesaria para obtener posición, cabe mencionar que aún es necesario utilizar el Arduino para las salidas PWM, por lo que debe mantenerse cierta consideración en la velocidad de implementación, ya que la computadora necesita comunicarse con 3 dispositivos a la vez y en ocasiones puede perder datos. Se menciona como mejora este tipo de implementación porque es posible realizar algoritmos de más alto nivel en un microprocesador más rápido y utilizando todas las funciones disponibles en simulink.

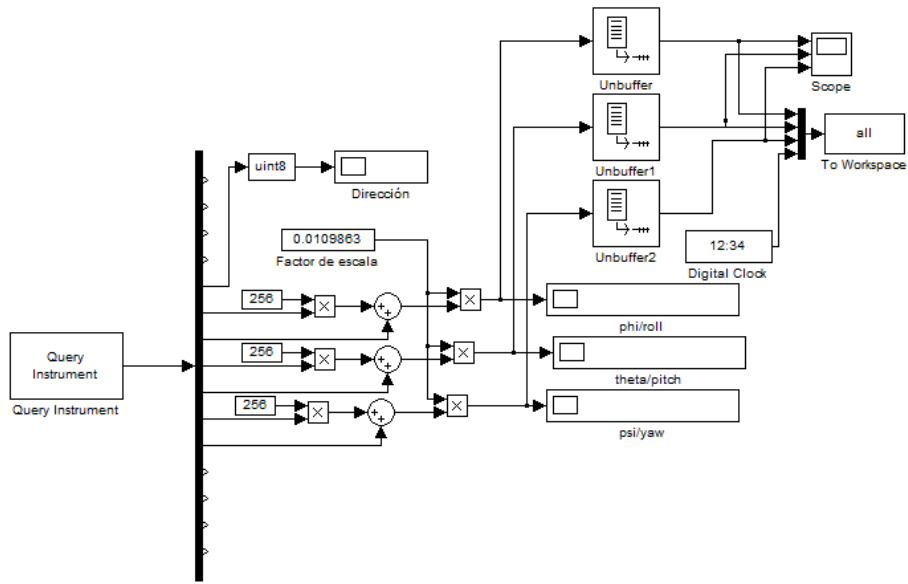


Fig. 4.9. Lectura del sensor UM6 en simulink

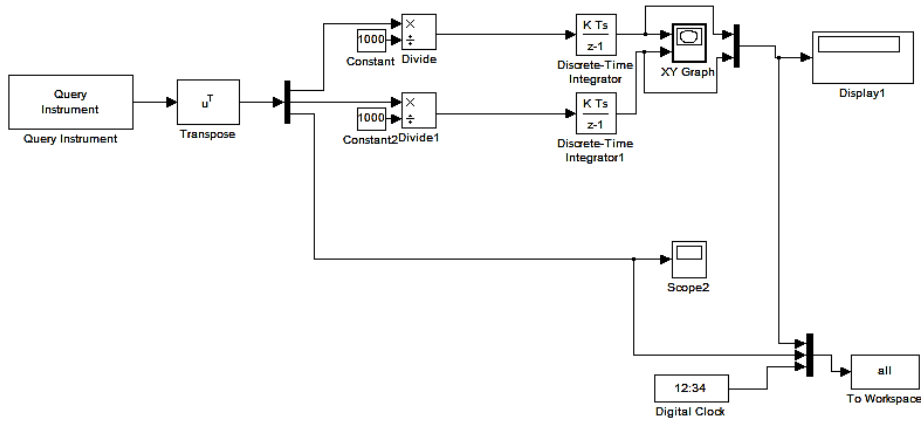


Fig. 4.10. Lectura del sensor PX4FLOW en simulink

Capítulo 5.

Resultados obtenidos

5.1. Resultados del control de orientación

Se realizó una prueba de estabilización, que se ejecuta usando el programa de comunicación con Arduino a 50Hz con referencias de 0° para los ángulos pitch, roll y yaw, en un intervalo de tiempo de 30 segundos, la medición de orientación se muestra en la _ 5.1 , y las señales de control generadas que se muestran en la figura _ 5.2.

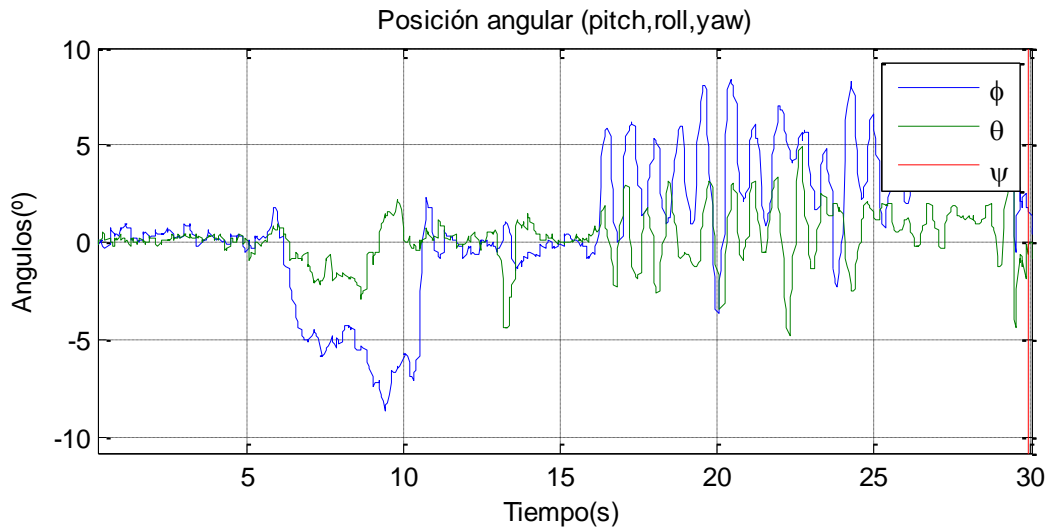


Fig. 5.1. Estimación de ángulos de Euler en Arduino para mini imu9

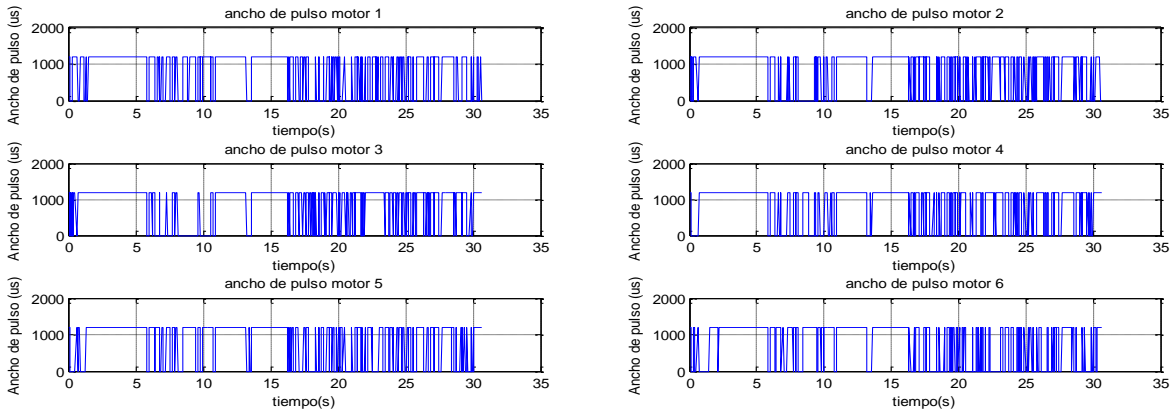


Fig. 5.2. Referencias PWM para controlar velocidad de motores

Se observa que el sistema presenta cierta oscilación menor a 5° , realizando un análisis de la configuración física y de acuerdo a (Pounds, Bersak, & Dollar, 2012), se debe colocar el peso adicional por encima del centro de gravedad. A continuación se presentan los resultados correspondientes a la misma prueba pero retirando el sensor kinect para reducir el peso y de esta forma demostrar el aumento de la estabilidad que se obtiene al utilizar una menor velocidad de giro en los motores. Se muestra en la Fig. 5.3 la medición de orientación, en la Fig. 5.4 la señal de control y medición para el ángulo pitch, y en la Fig. 5.5 la señal de control y medición para el ángulo roll.

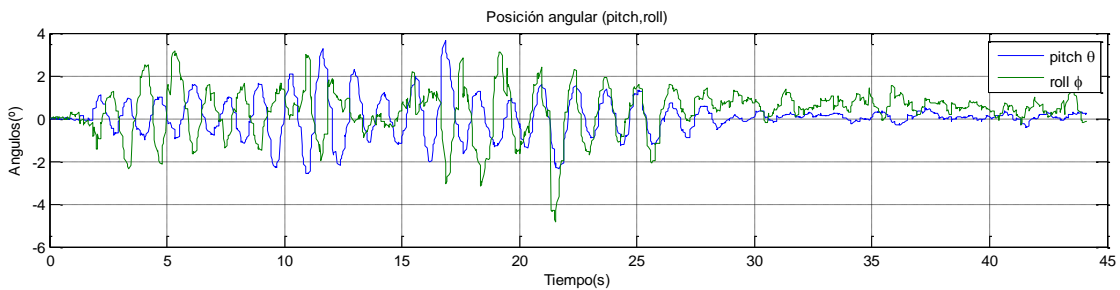


Fig. 5.3. Medición de los ángulos pitch y roll

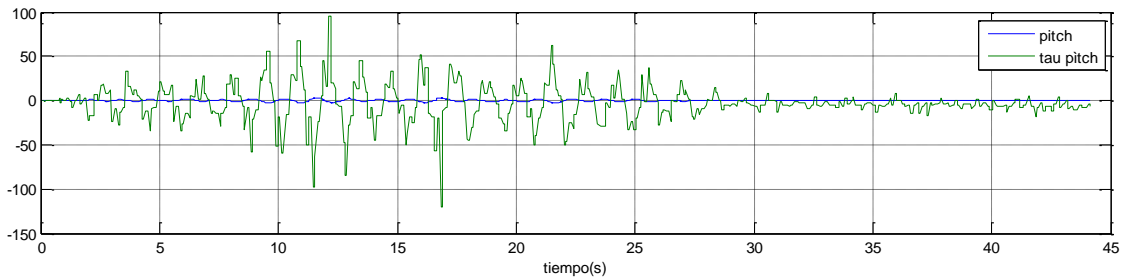


Fig. 5.4. Ángulo pitch y su señal de control

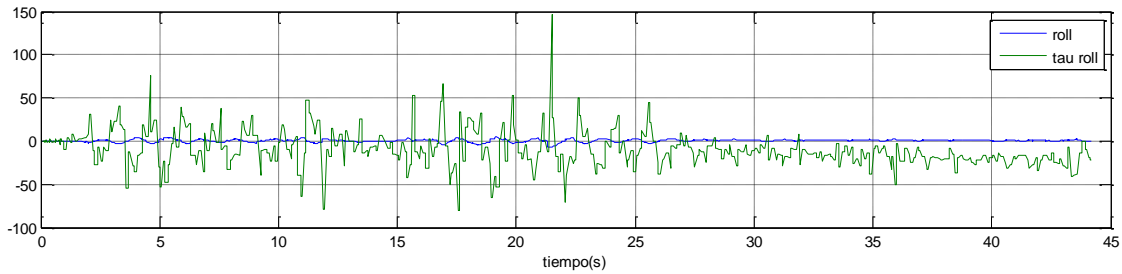


Fig. 5.5. Ángulo roll y su señal de control

De esta forma se comprueba lo descrito por (Pounds, Bersak, & Dollar, 2012) en cuanto a cómo la distribución de carga alrededor del centro de gravedad afecta la estabilidad. Para enfrentar esta situación una práctica común en el diseño de vehículos multirrotores es colocar hélices de mayor longitud para que de esta forma la velocidad del giro de los motores sea menor y produzca una fuerza de sustentación más uniforme, en este caso el vehículo es capaz de llevar una carga de 1kg extra, pero ha sido demostrado que no es garantizada estabilidad, reduciendo la carga se obtienen buenos resultados pero la problemática yace ahora en el control,

Se muestra en la Fig. 5.6 la respuesta ante una perturbación en el segundo 32, donde, a pesar de volver a estabilizar, la respuesta es amortiguada y lenta (aproximadamente 5 segundos).

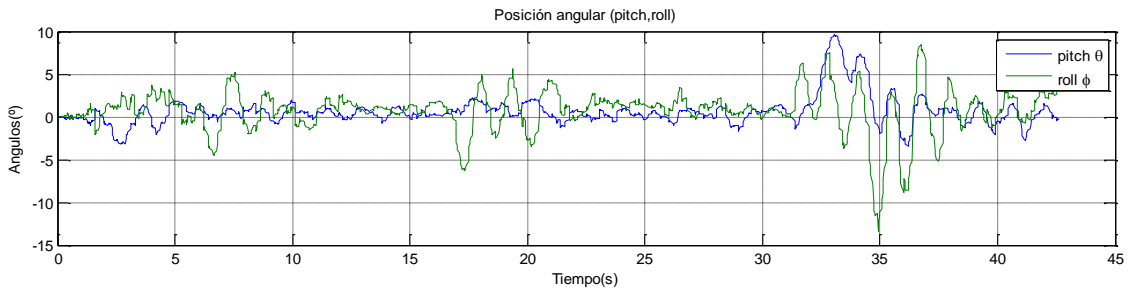


Fig. 5.6. Respuesta ante una perturbación externa.

Se demuestra que el sistema es controlable pero no se puede disminuir el tiempo de establecimiento considerablemente debido a la velocidad de ejecución del control.

5.2. Simulación del control de vehículos multirrotores

Partiendo de las ecuaciones (1.8) y (1.14) se reproduce el modelo con el siguiente vector de estado:

$$\dot{x}_h = (\dot{\xi}, \dot{\zeta}, \dot{\Omega}, \dot{\Omega}) = (\dot{x}, \dot{y}, \dot{z}, x, y, z, \dot{\phi}, \dot{\theta}, \dot{\psi}, \phi, \theta, \psi) \quad (2.5)$$

Donde $\xi, \dot{\xi}$ son la posición y velocidad lineales en el marco de referencia inercial, $\Omega, \dot{\Omega}$ son la posición y velocidad angulares en el marco de referencia del cuerpo, cuyas coordenadas han sido definidas previamente. Se utiliza una función S en la que se entrega como salida la integral del vector de estado, es decir la posición y velocidad en el marco de referencia inercial y su orientación y velocidad en términos de los ángulos de Euler.

Los parámetros se obtienen a partir de mediciones y se validan utilizando mecánica de cuerpo rígido rotacional:

Tabla 5.1. Parámetros de simulación

Parámetro	Valor	Unidades
I_a	0.0049	kgm^2
I_z	0.0088	kgm^2
k	2.98e-6	adimensional
l	0.275	m
m	1.065	kg

La simulación tiene una duración de 20 segundos con el método de integración es Runge-Kutta (ODE4) utilizando un paso de integración fijo de 0.001. La señal de referencia es arbitraria para producir movimientos traslacionales sobre los ejes x y y no simultáneos. Se parte de una altura inicial 0 y esta se logra estabilizar en la altura deseada.

La referencia de altura propuesta es una señal escalón de amplitud 5, a partir de t=0, cuyo seguimiento significa que el hexarrotor despegar y alcanza la altura de 5m. La referencia de orientación

del ángulo pitch de 0.5 rad (28.64°) con duración de 0.5s a partir de los 6 segundos de vuelo, para detener el hexarrotor debido a la ausencia de fricción es necesario aplicar la misma señal con signo contrario (-0.5 rad) lo cual se realiza a los 8 segundos.

La referencia de orientación del ángulo roll deseado inicia a partir de los 15 segundos y es de la misma amplitud y duración que la indicada en el ángulo pitch. En la siguiente figura se pueden observar las referencias de altura, pitch y roll, así como la altura y orientación del hexarrotor ante ellas.

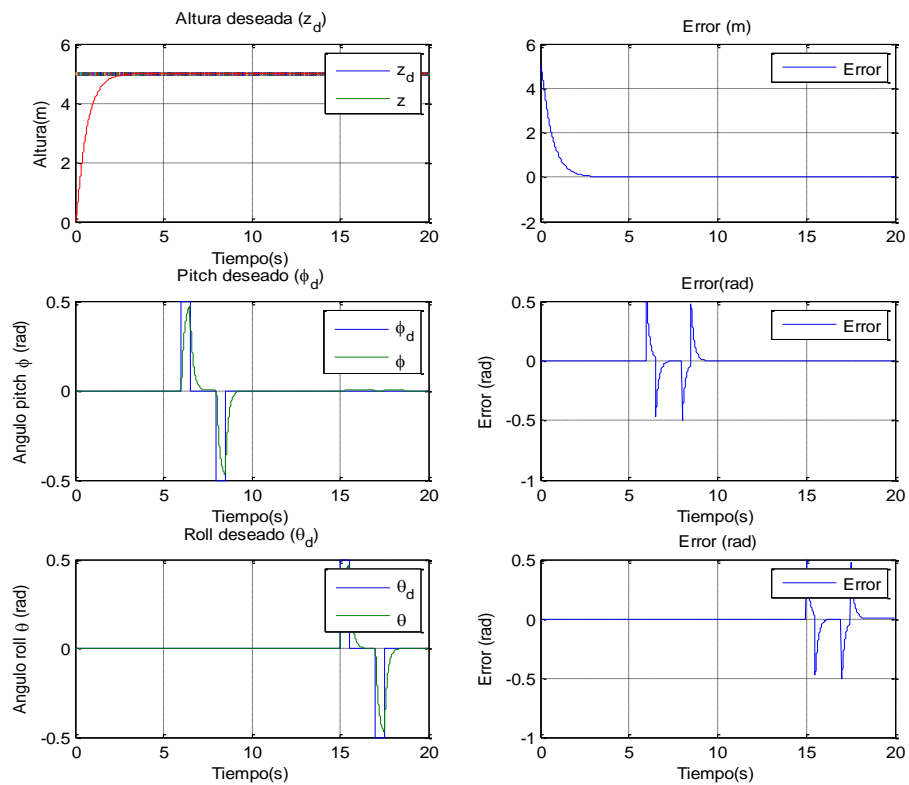


Fig. 5.7. Control de altura, pitch y roll ante entradas escalón arbitrarias.

Así, en las salidas del modelo, se pueden ver los desplazamientos lineales consecuencia de los momentos generados por los rotores.

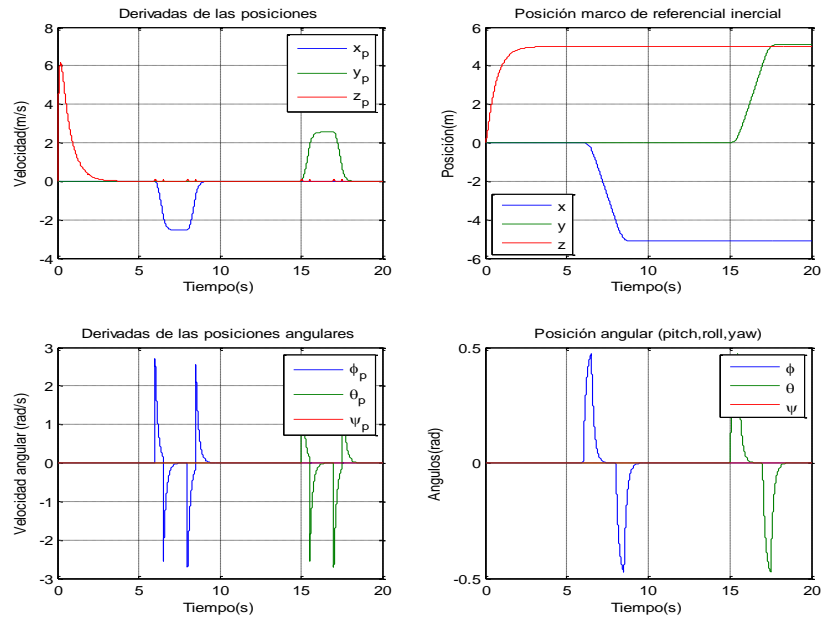


Fig. 5.8. Desplazamientos y velocidades lineales, Ángulos de Euler y sus derivadas

5.2.1. Simulación del seguimiento de una trayectoria

Para obtener las referencias angulares se utilizan las relaciones de las ecuaciones (2.6) y (2.7) dadas por (Mahoney, Kumar, & Corke, 2012) obtenidas geoméricamente,

$$\phi^* = \frac{1}{g} (x_d \sin \psi^* - y_d \cos \psi^*) \quad (2.6)$$

$$\theta^* = \frac{1}{g} (x_d \cos \psi^* + y_d \sin \psi^*) \quad (2.7)$$

la trayectoria deseada es una circunferencia de radio 0.65 m a una altura de 4 m, descrita por las ecuaciones parametrizadas siguientes

En simulación es posible no crear una trayectoria de altura ya que se simula en condiciones ideales, se muestra en la Fig. 5.9 la respuesta del sistema ante la trayectoria propuesta.

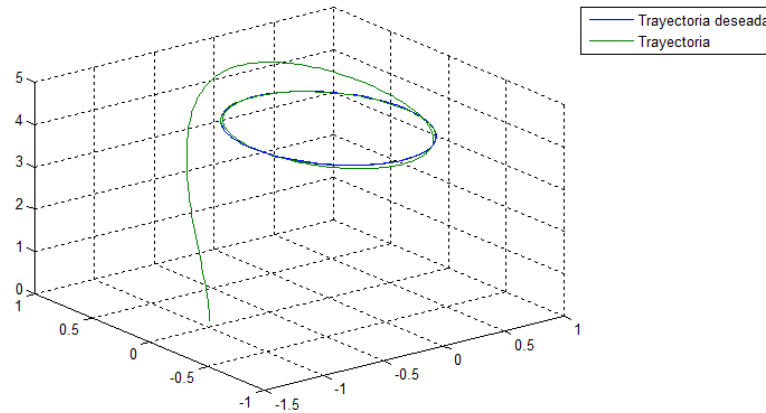


Fig. 5.9. Seguimiento de una trayectoria circular

5.2.2. Comparativa del control PID y un controlador backstepping

Se toman el controlador PD presentado previamente y se compara el desempeño con el controlador backstepping presentado en (Ortiz Torres, García Beltrán, Reyes Reyes, Vidal Rosas, & Astorga Zaragoza, 2015) ante una trayectoria en espiral descrita por las ecuaciones

$$x_d = 10 \cos(0.2t) + 10$$

$$y_d = 10 \sin(0.2t) + 10$$

$$z_d = 0.1t$$

$$\psi_d = 0.1 \sin(0.5t)$$

En la siguiente figura se puede observar de forma gráfica el error entre los dos controladores, mostrando un mejor desempeño el controlador backstepping a lo largo de toda la trayectoria.

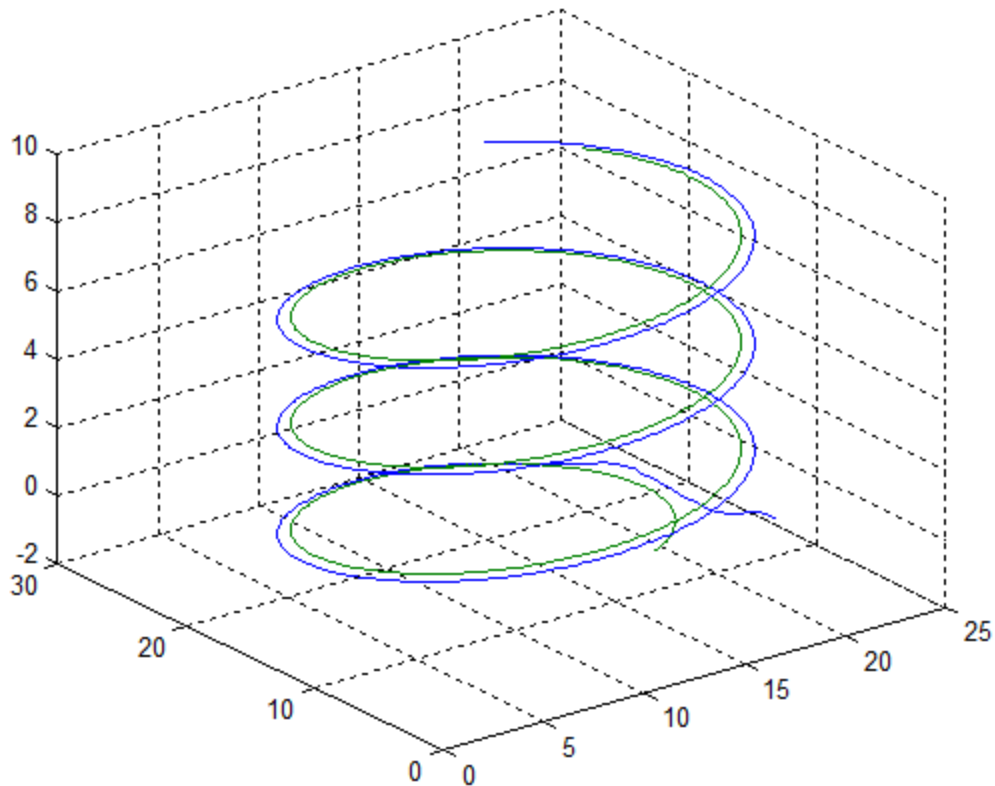


Fig. 5.10. Seguimiento de una trayectoria en espiral

5.3. Implementación del algoritmo de odometría visual

Se implementó un algoritmo de odometría visual basado en adquisición monocular de imágenes secuenciales cada 33 milisegundos. Este consiste en la ejecución iterativa de un conjunto de funciones sobre un par de imágenes consecutivas. Para esto se requiere calibrar la cámara a utilizar, conocer la altura y el pitch del vehículo que se asume cambiarán suavemente para garantizar la estabilidad del algoritmo. Debido a que el sensor kinect posee un campo de visión de 62.7° horizontales y 43° verticales se sugiere no enviar referencias angulares superiores a 20° .

La librería utilizada es LIBVISO2 desarrollada por Andreas Geiger en C++, para utilizar sus funciones y/o realizar cambios es necesario tener instalado un compilador de C++ compatible con Matlab, en este caso se utilizó Microsoft Visual Studio 2008, luego de tener guardados los archivos en un directorio accesible por Matlab, se ejecuta la función `make.m` (una vez seleccionado el compilador usando el comando `mex -setup`). De esta forma se crean los archivos `*.mexw32` que permiten el uso de las

funciones desde Matlab. En el anexo 2 se encuentra una descripción de las funciones principales del algoritmo de odometría visual

En la sección de odometría se realizó una prueba lineal y una curva, en la Fig. 5.11 se muestra un desplazamiento a pie, intentando seguir una línea recta por 5 pasos, se puede ver la última imagen capturada y la localización estimada aproximadamente realizada a cada paso. Se proporciona en el anexo 4, el conjunto de imágenes para demostrar la congruencia de la estimación con el desplazamiento realizado, se tiene un error en Y de 5.5 cm ya que se midió el punto final, cabe destacar que X fue muy preciso aunque existe una rotación sobre el plano XY en la última imagen.

De igual forma se hizo una prueba de desplazamiento en un segmento de circunferencia con un radio de 1.1m, y la estimación de desplazamiento obtenida se muestra en la Fig. 5.12 con un error máximo de 8cm, y un error promedio de 5cm.

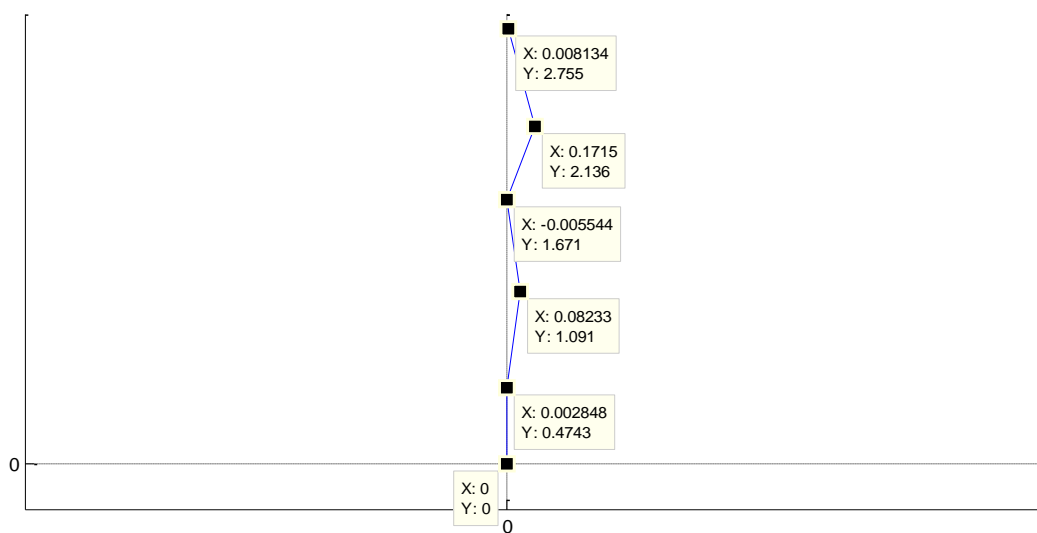


Fig. 5.11. Prueba desplazamiento lineal a pie

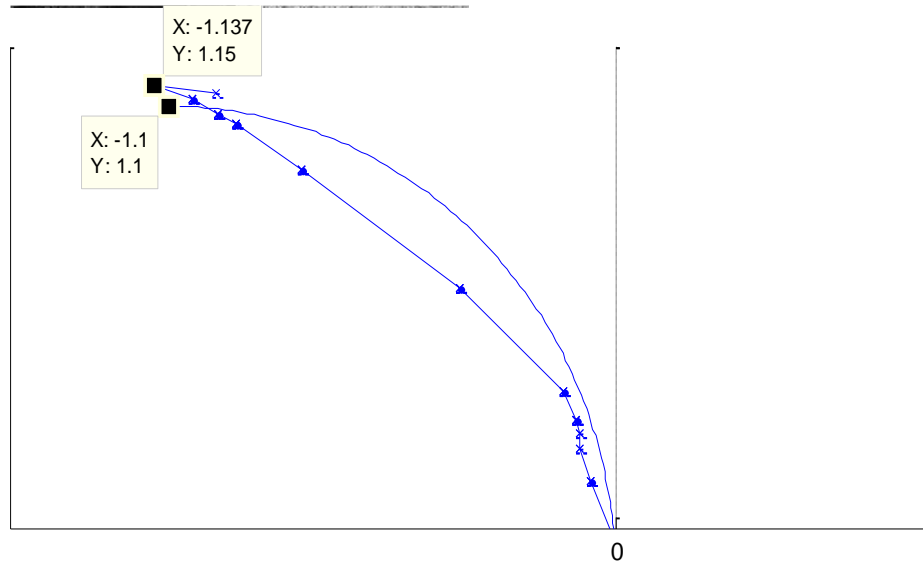


Fig. 5.12. Prueba de un desplazamiento en arco

5.4. Flujo óptico

Como fue mencionado en la sección de estimación de desplazamiento, es necesario incluir información adicional como altura y orientación para producir una salida de velocidad en metros y no en píxeles, de cualquier forma se realizó una prueba de un algoritmo de flujo óptico para mostrar su principio de funcionamiento, la cual se ejecuta en el primer par de imágenes del conjunto obtenido para la prueba en arco, se puede apreciar en la Fig. 5.13 la dirección del giro indicada por el movimiento aparente de algunos puntos característicos, pero no se obtiene ninguna información ni medición en el mundo real.



Fig. 5.13. Flujo óptico entre dos imágenes para un movimiento angular

Capítulo 6.

Conclusiones

Se realizó el modelado de un hexarrotor y se simuló controles básicos utilizando simulink, se muestran resultados de control de orientación y desplazamiento. Se implementó un control de posición que sigue referencias angulares cero, es decir conserva su posición inicial, a una frecuencia de 50Hz utilizando una plataforma abierta y un programa propio. Se realizó además una estimación de desplazamiento utilizando un algoritmo de odometría visual en línea basado en funciones pre-compiladas de una librería de procesamiento de imágenes de código abierto. A partir de las pruebas realizadas se instrumentó el prototipo para realizar desarrollos y pruebas con leyes de control distintas embebidas en el microcontrolador Atmega o implementadas incluso en simulink, todo esto utilizando sensores convencionales. Como consecuencia de este trabajo y el esfuerzo colectivo del grupo de vehículos aéreos en CENIDET de reciente creación se enviará un artículo a la XVI Convención de Ingeniería Eléctrica CIE2015.

Referencias

- A. A., & F. B. (1999). Combining Central and Peripheral Vision for Reactive Robot Navigation. *Computer Vision and Pattern Recognition Conference*, (págs. 2646-2651). Colorado, USA.
- A. B., J.-C. Z., & D. F. (2013). Optic-Flow Based Control of a 46g Quadrotor. *International Conference on Intelligent Robots and Systems*.
- Alaimo, A., Artale, V., Milazzo, C. L., & Ricciardello, A. (2013). PID Controller Applied to Hexacopter Flight. *Journal of Intelligent and Robotic Systems*, 261-270.
- Altug, Z. C. (2012). Image Based and Hybrid Visual Servo Control of an Unmanned Aerial Vehicle. *Journal of Intelligent & Robotic Systems Volume*, 324-344.
- Arellano Muro, C. A., Luque Vega, L. F., Castillo Toledo, B., & Loukianov, A. G. (2013). Backstepping Control with Sliding Mode Estimation for a Hexacopter. *10th International Conference on Electrical Engineering, Computing Science and Automatic Control(CCE)* (págs. 31-36). Mexico: IEEE.
- Artale, V., Barbaraci, G., Milazzo, C., Orlando, C., & Ricciardello, A. (2013). Dynamic Analysis of a Hexacopter Controlled via LQR-PI. *AIP Conference Proceedings*, 1212-1215.
- Baránek, R., & Šolc, F. (2012). Modelling and Control of a Hexa-copter. *International Carpathian Control Conference (ICCC)* (págs. 19-23). IEEE.
- Bermúdez, G. (2002). Robots móviles. Teoría, aplicaciones y experiencias. *Tecnura*, 6-17.
- Chaumette, R. O. (2011). Dynamic Visual Servoing with Image Moments for a Quadrotor Using a Virtual Spring Approach. *IEEE International Conference on Robotics and Automation*, 5670-5676.
- Corke, P. (2011). *Robotics, Vision and Control. Fundamental Algorithms in MATLAB*. Berlin: Springer Tracts in Advanced Robotics.

-
- D. Scaramuzza, M. A. (Conditionally Accepted.). Vision-Controlled Micro Flying Robots: from System Design to Autonomous Navigation. *IEEE Robotics and Automation Magazine*.
- F. K., I. F., & K. N. (2009). Optic Flow-Based Vision System for Autonomous 3D Localization and Control of Small Aerial Vehicles. *Robotics and Autonomous Systems, Elsevier*, 591-602.
- Fraundorfer, F., & Scaramuzza, D. (2011). Visual Odometry. Part I: The First 30 years and Fundamentals. *IEEE Robotics and automation Magazine*, 80-92.
- Fraundorfer, F., & Scaramuzza, D. (2012). Visual Odometry. Part II: Matching Robustness, Optimization and Applications. *IEEE ROBOTICS AND AUTOMATION MAGAZINE*, 78-90.
- García Carrillo, L., Dzul, A., & Lozano, R. (2012). Hovering Quad-Rotor Control: A Comparison of Nonlinear Controllers using Visual Feedback.. *IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS*, 3159-3170.
- Giovanni, C. (1979). Performance of a ring laser strapdown attitude and heading reference for aircraft. *Journal of Guidance, Control, and Dynamics*, 320–327.
- Gonzales Sánchez, M., Amézquita Brooks, E., & Liceaga Castro, E. (2013). Simplificación de Modelos de Cuadritores. *Congreso Nacional de Control Automatico* (págs. 81-86). Ensenada, Baja California: AMCA.
- Gonzales, I., Salazar, S., Romero, H., Lozano, R., & Torres, O. (s.f.). Attitude Control of a Quadrotor using Speed sensing in Brushless DC Motors.
- Horn, B. K., & Schunck, B. G. (1981). Determining optical flow. *Artificial intelligence*, 185-203.
- Hu, H., & Gu, D. (2000). Landmark-based Navigation of Industrial Mobile Robot. *International Journal of Industry Robot*, 458-467.
- Huang, A., Bachrach, A., Henry, P., Krainin, M., Maturana, D., Fox, D., & Roy, N. (2011). Visual Odometry and Mapping for Autonomous. *15th International Symposium on Robotics Research*.

-
- J. C., G. G., B. R., & J. H. (2009). Implementation of wide-field integration of optic flow for autonomous quadrotor navigation. *Autonomous Robots*, 189-198.
- J. S., F. R., S. V., & N. F. (2006). Toward Optic Flow Regulation for Wall-Following and Centring Behaviours. *International Journal of Advanced Robotic Systems*, 147-154.
- J.-C. Z., & D. F. (2005). Toward 30-Gram Autonomous Indoor Aircraft: Vision-Based Obstacle Avoid-Ance and Altitude Control. *International Conference on Robotics and Automation*. Barcelona, Spain: IEEE.
- King, T. (1998). *Impact: The History of Germany's V-Weapons in World War II*. New York: Sarpedon Publishers.
- L. M., S. D., Y. B., & J.-A. M. (2005). A contribution to Vision-Based Autonomous Helicopter Flight in Urban Environments. *Robotics and Autonomous Systems*.
- Leishman, R., Macdonald, J., McLain, T., & Beard, R. (2012). Relative Navigation and Control of a Hexacopter. *International Conference on Robotics and Automation* (págs. 4937-4942). Saint Paul, Minnesota: IEEE.
- Liu, H., Derawi, D., Kim, J., & Zhong, Y. (2013). Robust optimal attitude control of hexarotor robotic vehicles. *Nonlinear Dynamics*. Springer, 1155-1168.
- Lozano, H. R. (2006). Stabilization and location of a four rotor helicopter applying vision. *Proceedings of the 2006 American Control Conference*, 3930-3935.
- Luukkonen, T. (2011). *Modelling and control of quadcopter*. Espoo, Finland: Aalto University.
- Mahoney, R., Kumar, V., & Corke, P. (2012). Multirrotor Aerial Vehicles, Modeling, Estimation and Control of Quadrotor. *IEEE Robotics & Automation Magazine*, 20-32.
- Mahony, E. A. (2002). Control of a Quadrotor Helicopter Using Visual Feedback. *International Conference on Robotics & Automation*, 72-77.

-
- Mahony, N. G. (2008). A Practical Visual Servo Control for an Unmanned. *IEEE TRANSACTIONS ON ROBOTICS*, 331-340.
- Martínez Ramírez, F. (2012). *Control Embebido de un Vehículo Guiado Automáticamente Mediante Redes Neuronales Artificiales*. Cuernavaca, Mor.: CENIDET.
- Moravec, H. (1980). *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover*. California. Estados Unidos: Stanford University.
- Ortiz Torres, G., García Beltrán, C., Reyes Reyes, J., Vidal Rosas, A., & Astorga Zaragoza, C. (2015). Control Tolerante a Fallas Pasivo de un Octorotor tipo X8 utilizando Controladores Backstepping en Cascada. *XVI CONVENCIÓN DE INGENIERÍA ELÉCTRICA, CIE*.
- Pégard, L. R. (2013). *Quad Rotorcraft. Vision-Based Hovering and Navigation*. Advances in Industrial Control: Springer.
- Pounds, P. E., Bersak, D. R., & Dollar, A. M. (2012). Stability of small-scale UAV helicopters and quadrotors. *Autonomous Robots*, 129-142.
- Schereier, M. (2012). Modeling and Adaptive Control of a Quadrotor. *International Conference on Mechatronics and Automation* (págs. 383-390). Chengdu, China: Proceedings of 2012 IEEE.
- Suseong Kim, D. L. (2011). Image Based Visual Servoing for an Autonomous Quadrotor with Adaptive Backstepping Control. *11th International Conference on Control, Automation and Systems*, 532-537.
- Var-Shalom, Y., & Fortmann, T. (1998). *Tracking and Data Association*. Boston: Mathematics in Science and Engineering. Elsevier Science.
- Winston Yui, N. D. (2011). Visual Localisation of a Robot with an external RGBD Sensor. *Australasian Conference on Robotics and Automation (ACRA)*. Melbourne, Australia.

ANEXO 1

Conjunto de imágenes para prueba de desplazamiento lineal



ANEXO 2

Conjunto de imágenes para prueba de desplazamiento en arco

