







**CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS  
DEL INSTITUTO POLITÉCNICO NACIONAL**

UNIDAD ZACATENCO

PROGRAMA DE SISTEMAS AUTÓNOMOS DE NAVEGACIÓN AÉREA Y SUBMARINA

## **Vuelo en Formación de Múltiples UAV en Modo "Líder-Seguidor" para Evasión de Obstáculos**

Tesis que presenta

**Roberto Carlos Osorio Castro**

Para obtener el grado de

**MAESTRO EN CIENCIAS**

En

**SISTEMAS AUTÓNOMOS DE NAVEGACIÓN AÉREA Y SUBMARINA**

Directores de Tesis:

Dr. Iván González Hernández

Dr. Mauricio Lara Barrón



# Índice general

---

<b>Índice de tablas</b>	<b>VII</b>
<b>Índice de figuras</b>	<b>IX</b>
<b>1. Introducción</b>	<b>5</b>
1.1. Descripción y Evolución de los <i>UAV</i>	6
1.1.1. Clasificación de los <i>UAV</i>	6
1.2. Vuelo en formación	7
1.2.1. Configuración MIMO (Multiple-Input and Multiple-Output)	7
1.2.2. Configuración Líder-Seguidor	7
1.2.3. Configuración en Estructura Virtual	8
1.2.4. Configuración de Control Basado en Comportamiento	8
1.2.5. Configuración de Control Basado en Pasividad	8
1.3. Sistema de Visión Artificial	8
1.3.1. Detección de obstáculos	9
1.4. Objetivo general	9
1.5. Objetivos específicos	9
1.6. Planteamiento del problema	10
1.7. Motivación	10
<b>2. Modelado matemático del prototipo cuadri-rotor</b>	<b>11</b>
2.1. Descripción del cuadri-rotor	11
2.2. Modelo matemático del cuadri-rotor	11
<b>3. Plataforma experimental</b>	<b>17</b>
3.1. Interfaz de comunicación RS-485 y distribución de energía	17
3.2. Autopiloto	18
3.3. Esquema de la plataforma experimental: Líder-Seguidor	19
3.3.1. Agente "Líder"	19
3.3.2. Agente "Seguidor"	21
3.3.3. <i>ROS</i>	22
3.4. Módulo de radiofrecuencia	22
3.5. Interfaz de Control en Tierra	24

<b>4. Estrategia de control</b>	<b>29</b>
4.1. Procesamiento de señales . . . . .	29
4.1.1. Concepto de Frecuencia para señales en tiempo continuo y tiempo discreto	29
4.1.1.1. Señales sinusoidales en tiempo continuo . . . . .	29
4.1.1.2. Señales sinusoidales en tiempo discreto . . . . .	30
4.1.2. Conversión Analógico-Digital . . . . .	30
4.1.3. Muestreo . . . . .	31
4.1.4. Cuantificación . . . . .	33
4.1.5. Codificación . . . . .	33
4.2. Sistemas Discretos . . . . .	33
4.3. Controlador <i>PID</i> continuo . . . . .	34
4.4. Sistemas de Control en tiempo Discreto . . . . .	34
4.5. Controlador <i>PID</i> discreto . . . . .	34
4.6. Control de Traslación: Linealización exacta . . . . .	36
4.7. Análisis de estabilidad . . . . .	38
<b>5. Sistema de Visión Artificial</b>	<b>41</b>
5.1. Modelo de Cámara . . . . .	41
5.1.1. Representación en 2-D . . . . .	41
5.1.2. Representación en 3-D . . . . .	42
5.2. Detección de Puntos de Interés y Emparejamiento . . . . .	43
5.3. Escala y Pirámides Gaussianas . . . . .	44
5.3.1. Pirámides Gaussianas . . . . .	44
5.3.2. Espacio escala . . . . .	44
5.4. Descriptores Binarios . . . . .	45
5.5. Reconocimiento de patrones . . . . .	45
5.6. El problema de clasificación . . . . .	46
5.6.1. Modelos lineales para clasificación . . . . .	46
5.6.2. Máquina de Soporte Vectorial ( <i>SVM</i> ) . . . . .	47
5.7. Algoritmo de detección de personas . . . . .	50
5.7.1. Histograma de Gradientes Orientados ( <i>HOG</i> ) . . . . .	51
<b>6. Vuelo en formación</b>	<b>53</b>
6.1. Teoría de Grafos . . . . .	53
6.1.1. Grafo dirigido de Dependencia . . . . .	54
6.2. Navegación en exteriores . . . . .	55
6.2.1. Sistema de coordenadas geodésicas . . . . .	55
6.2.2. Sistema <i>ECEF</i> . . . . .	56
6.2.3. Sistemas de Coordenadas Fijos a la Superficie Terrestre . . . . .	57
6.2.3.1. Sistema de Coordenadas NED . . . . .	57
6.2.3.2. Sistema de Coordenadas ENU . . . . .	57
<b>7. Protocolo de Comunicación</b>	<b>59</b>
7.1. Modelo de Referencia ( <i>OSI</i> ) . . . . .	59
7.1.1. Arquitectura . . . . .	60
7.1.1.1. Estratificación o Sistemas de Capas . . . . .	60

7.1.1.1.1.	Capa de Aplicación	
	.....	61
7.1.1.1.2.	Capa de Presentación	
	.....	61
7.1.1.1.3.	Capa de Sesión	
	.....	61
7.1.1.1.4.	Capa de Transporte	
	.....	62
7.1.1.1.5.	Capa de Red	
	.....	62
7.1.1.1.6.	Capa de Enlace de Datos	
	.....	62
7.1.1.1.7.	Capa de Física	
	.....	62
7.1.2.	Especificaciones de Servicio	62
7.1.3.	Especificaciones de Protocolo	63
7.2.	Protocolos de Acceso Múltiple	64
7.2.1.	De Contención	66
7.2.2.	Protocolos Libres de Conflicto	66
7.2.3.	Protocolos de Detección de Portadora	66
7.2.3.1.	Carrier Sense Multiple Access/Collision Avoidance (CS- MA/CA)	67
7.2.3.1.1.	Estación Escondida	
	.....	67
7.2.3.1.2.	Estación Visible	
	.....	67
7.3.	Constitución de la trama de la capa de enlace de datos	69
7.3.1.	Tramas orientadas a bits	70
7.3.2.	Trama de datos	70
7.3.3.	Control de flujo de datos	71
7.3.4.	Control de Errores	72
7.3.4.1.	Solicitud de Repetición Automática (ARQ)	72
7.3.4.2.	Código de Redundancia Cíclica (CRC)	73
7.3.5.	Características del protocolo implementado	73
7.3.5.1.	Calendarización de envío de paquetes	73
7.3.5.2.	Información de los dispositivos dentro de la red	74

<b>8. Simulación y Resultados Experimentales</b>	<b>75</b>
8.1. Controlador . . . . .	75
8.2. Controlador de Orientación . . . . .	76
8.2.1. Controlador de Orientación para el agente Líder . . . . .	76
8.3. Controlador de Posición . . . . .	77
8.3.1. Controlador de Posición para el agente Líder . . . . .	77
<b>9. Conclusiones y Trabajo a Futuro</b>	<b>81</b>
<b>10. Apéndice A</b>	<b>83</b>
10.1. Transformación del sistema de coordenadas geodésicas al sistema NED . . . . .	83
<b>11. Apéndice B</b>	<b>87</b>
11.1. Representación y Cálculo del CRC . . . . .	87
<b>12. Apéndice C</b>	<b>91</b>
12.1. Modelo de propagación de espacio libre . . . . .	91
<b>Bibliografía</b>	<b>93</b>

# Índice de tablas

---

3.1. Especificaciones del <i>IC CC2530F256</i> . . . . .	23
3.2. Especificaciones de la antena dipolo <i>GW.59</i> . . . . .	24
7.1. Constitución de la trama de datos y función de cada campo . . . . .	71



# Índice de figuras

---

1.1. UAV RQ-1A utilizado por el Ejército de Estados Unidos . . . . .	6
2.1. Diagrama de cuerpo libre del cuadri-rotor . . . . .	12
2.2. Marcos de referencia inercial y del cuerpo . . . . .	12
3.1. Modelo 3D de la PCB . . . . .	18
3.2. PCB ensamblada . . . . .	18
3.3. Autopiloto Pixhawk PX4 . . . . .	18
3.4. Tarjeta Odroid XU4 . . . . .	20
3.5. Diagrama de flujo del funcionamiento e implementación de la plataforma "Líder"	20
3.6. Tarjeta Raspberry Pi 3 . . . . .	21
3.7. Diagrama de flujo del funcionamiento e implementación de la plataforma "Se- guidor" . . . . .	22
3.8. Módulo de radiofrecuencia . . . . .	23
3.9. Antena dipolo GW.59 . . . . .	23
3.10. Módulo de radiofrecuencia empleado en la GCS . . . . .	24
3.11. Ventana de inicio . . . . .	25
3.12. Ventana de Menú . . . . .	25
3.13. Ventana de Dispositivos dentro de la red . . . . .	26
3.14. Ventana de Consola . . . . .	26
3.15. Ventana de Datos del estado del UAV . . . . .	27
3.16. Gráfica en tiempo real de los datos de $pitch(\theta)$ . . . . .	28
3.17. Ejemplo de un archivo <i>logger.txt</i> . . . . .	28
4.1. Muestreo periódico de una señal en tiempo continuo . . . . .	31
4.2. Proceso y elementos de la conversión analógico-digital . . . . .	33
5.1. Proyección de imagen del espacio tridimensional al plano imagen . . . . .	42
5.2. Sistema coordenado 3D . . . . .	43
5.3. Comportamiento de las señales dentro de una imagen a medida que aumenta la escala . . . . .	45
5.4. Detección del punto de interés en el espacio escala . . . . .	47
5.5. Proceso de detección del objeto . . . . .	51
6.1. Representación de un grafo dirigido . . . . .	54

6.2.	Representación de un grafo dirigido . . . . .	54
6.3.	Sistema de Coordenadas Geodésicas . . . . .	56
6.4.	Sistema de Coordenadas <i>ECEF</i> . . . . .	56
6.5.	Sistema de Coordenadas <i>ECEF</i> . . . . .	57
6.6.	Sistema de Coordenadas <i>ENU</i> . . . . .	58
7.1.	Niveles de abstracción en <i>OSI</i> . . . . .	60
7.2.	Arquitectura <i>OSI</i> de capas . . . . .	61
7.3.	Modelo de Servicio <i>OSI</i> . . . . .	63
7.4.	Modelo de Protocolo <i>OSI</i> . . . . .	63
7.5.	Representación de elementos de <i>OSI</i> . . . . .	64
7.6.	Clasificación de Protocolos de Acceso Múltiple . . . . .	65
7.7.	(a) Problema de la estación escondida. (b) Problema de la estación visible . . . . .	68
7.8.	Ejemplo de la detección virtual . . . . .	68
7.9.	Detección virtual de canal utilizando <i>CSMA/CA</i> . . . . .	69
7.10.	Formato de bloque de datos . . . . .	70
7.11.	Trama orientada a bits . . . . .	70
7.12.	Detección de errores . . . . .	72
7.13.	Diagrama del control de flujo de datos . . . . .	74
8.1.	Esquemático a bloques del modelo Líder-Seguidor . . . . .	75
8.2.	Respuesta del controlador para orientación del Ángulo Yaw ( $\psi$ ) del agente Líder . . . . .	76
8.3.	Respuesta del controlador para orientación del Ángulo Pitch ( $\theta$ ) del agente Líder . . . . .	77
8.4.	Respuesta del controlador para orientación del Ángulo Roll ( $\phi$ ) del agente Líder . . . . .	77
8.5.	Respuesta del controlador para posición en el Eje X del agente Líder . . . . .	78
8.6.	Respuesta del controlador para posición en el Eje Y del agente Líder . . . . .	78
8.7.	Respuesta del controlador para posición en el Eje Z del agente Líder . . . . .	78
8.8.	Posición tridimensional de los agentes Líder y Seguidor dentro de la formación . . . . .	79
8.9.	Posición biimensional de los agentes Líder y Seguidor dentro de la formación . . . . .	79

# Agradecimientos

Agradezco al Consejo Nacional de Ciencia y Tecnología (*CONACYT*) por brindarme el apoyo necesario para realizar mis estudios de posgrado durante el tiempo de duración de los mismos, y cuya culminación se ve reflejada en el presente trabajo de tesis.

Agradezco a la *UMI-LAFMIA* por brindarme los conocimientos y el material necesario para realizar el presente trabajo de Tesis.

Agradezco al Dr. Iván González Hernández y al Dr. Mauricio Lara Barrón por su asesoría durante la realización del proyecto de Tesis.

Agradezco al Dr. Aldo Orozco Lugo y a la Dra. Giselle Galván Tejada por su co-asesoría para llevar a cabo el presente trabajo de Tesis y por su diligencia en la realización del mismo.

## Abreviaturas

<b>UAV</b>	Unmanned Aerial Vehicle
<b>DS</b>	Deep Space
<b>POE</b>	Planetary Orbital Environments
<b><math>\mu C</math></b>	Microcontrolador/Microcontroller
<b>LASER</b>	Light Amplification by Stimulated Emission of Radiation
<b>TTL</b>	Transistor-Transistor Logic
<b>CMOS</b>	Complementary Metal-Oxide Semiconductor
<b>GCS</b>	Ground Control Station
<b>BLDC</b>	Brushless Direct Current
<b>PCB</b>	Printed Circuit Board
<b>SBC</b>	Single Board Computer
<b>LDO</b>	Low Drop Out
<b>EDA</b>	Electronic Design Automation
<b>USB</b>	Universal Serial Bus
<b>GPU</b>	Graphics Processor Unit
<b>RAM</b>	Random-Access Memory
<b>HDMI</b>	High-Definition Multimedia Interface
<b>GPS</b>	Global Positioning System
<b>ESC</b>	Electronic Speed Controller
<b>SoC</b>	System on a Chip
<b>ROS</b>	Robot Operating System
<b>IC</b>	Integrated Circuit
<b>CSMA/CA</b>	Carrier Sense Multiple Access with Collision Avoidance
<b>MAC</b>	Medium Access Control
<b>PID</b>	Proportional Integral Derivative
<b>PV</b>	Process Variable
<b>SVM</b>	Support Vector Machine
<b>HOG</b>	Histogram of Oriented Gradients
<b>SIFT</b>	Scale-Invariant Feature Transform
<b>WGS</b>	World Geodetic System
<b>ECEF</b>	Earth-Centered Earth-Fixed
<b>NED</b>	North-East-Down
<b>ENU</b>	East-North-Up
<b>ISO</b>	International Organization for Standarization
<b>OSI</b>	Open Systems Interconnection
<b>RTS</b>	Request To Send
<b>CTS</b>	Clear To Send
<b>ACK</b>	Acknowledgement
<b>NAV</b>	Network Allocation Vector
<b>CRC</b>	Cyclic Redundancy Check
<b>ARQ</b>	Automatic Repeat Request

## Resumen

El presente trabajo de tesis se enfoca en el *Vuelo en Formación* de múltiples UAV bajo la configuración de *Líder-Seguidor* y en la detección de un objetivo por parte del líder de la misma, el cual transmitirá dicha información al resto de las plataformas que conforman el enjambre para realizar las maniobras pertinentes para evitar una posible colisión con el objeto.

Para realizar la estabilización del prototipo, tanto en orientación como en posición, se desarrolló una ley de control de tipo *PID* programada dentro de la plataforma *Pixhawk*, la cual es la encargada de modificar la dinámica del prototipo con la finalidad de llegar a un estado deseado.

Para la detección de una persona dentro de la trayectoria del prototipo, se hizo uso de una cámara digital, con la cual se adquieren imágenes del entorno para obtener Histogramas de Gradientes Orientados para alimentar una *Máquina de Soporte Vectorial*, con la finalidad de realizar la clasificación del objeto detectado.

# Abstract

The present thesis focuses on *Flight formation* of multiple *UAV* under *Leader-follower* configuration and in the detection of a target by the configuration's leader, which will transmit this information to the rest of the platforms that integrate the swarm in order to take the needed maneuvers to avoid a possible collision with the object.

To stabilize the prototype in attitude and position, a *PID* control law was developed, which was programmed on a *Pixhawk* platform, the one that has the task of modify de prototype's dynamics to archieve a desired state.

For the person detection within the prototype's trayectory, a digital camera was used, form which a series of images of the surroundings were obtained, to develop Histograms of Oriented Gradients to feed a Support Vector Machine, to sort out the detected object.

---

# Introducción

---

El vuelo en formación y el control de múltiples *UAV*, del inglés *Unmanned Aerial Vehicle*, ha recibido mucha atención a lo largo de las últimas décadas debido a su vasta gama de aplicaciones que pueden ser logradas de manera más eficiente por parte de un grupo de agentes que por un solo vehículo autónomo, tanto para labores en ambientes en el espacio profundo (*DS*), del inglés *Deep Space*, en ambientes orbitales planetarios (*POE*), del inglés *Planetary Orbital Environment*, como para misiones dentro de la atmósfera del planeta [1].

Dentro de las diversas tareas que debe ser capaz de realizar un sistema de navegación autónomo se encuentra la relacionada con la detección y evasión de obstáculos que puedan localizarse a lo largo de su trayectoria o en sus alrededores inmediatos. Para el caso de los *UAV*, dicho problema ha sido abordado de distintas maneras, siendo la forma más común, el uso de sensores reflectivos de tipo ultrasónico o por medio de *LASER*, del inglés *Light Amplification by Stimulated Emission of Radiation* para la detección de un objetivo y para la estimación de la posición del mismo, dando de esta manera información a la plataforma sobre una posible colisión y qué en consecuencia, se realice una rutina de evasión del objetivo.

Aunque estos métodos presentan resultados aceptables, los mismos se obtienen en una primer aproximación dentro de ambientes controlados en donde la reflexión de la señal emitida se garantiza, evitando de esta manera los llamados "falsos positivos" que pudiesen entregar una lectura errónea y en consecuencia, afectar a la acción de control del sistema. Para escenarios en donde las condiciones ambientales no presentan una cierta uniformidad como en el caso antes descrito, el uso de dichos sensores se realiza de manera convencional en conjunto con algún otro tipo de sensor que pueda entregar más información sobre el entorno, para de esta manera obtener una mejor respuesta por parte del sistema.

Un método alternativo para la detección de un objetivo es por medio de un sistema de visión artificial embebido dentro del *UAV*, el cual puede constar de una o más cámaras fotográficas, cuya principal función es obtener imágenes consecutivas del entorno o parte del mismo, para su posterior procesamiento y análisis. De esta manera es posible obtener información relevante sobre el medio en el que se desempeña el *UAV*.

## 1.1. Descripción y Evolución de los UAV

Un UAV puede ser descrito como "una aeronave sin un piloto humano a bordo, la cual puede ser reutilizada y es capaz de realizar vuelo controlado, transportar cargas, etc". De cierta manera, el origen de los UAV guarda similitud con el desarrollo, manufactura y uso de misiles, con la diferencia inherente relacionada al uso de cada uno de ellos: el propósito de un misil es el impactar un objetivo, destruyéndose a si mismo y al objetivo en el proceso, mientras que un UAV se encuentra diseñado para realizar múltiples misiones [2]. A lo largo de la historia, los mayores desarrollos y avances dentro del área de UAV se han dado principalmente durante conflictos bélicos, en donde, los primeros de los que se tiene información consistían en sistemas propulsados por aire caliente, utilizados para desplegar bombas a ciudades enemigas, los cuales no contaban con ningún tipo de acción de control en lazo cerrado que hiciera más eficiente su aplicación.

Posteriormente, el desarrollo de UAV sufrió un mayor avance durante la Segunda Guerra Mundial [2] y la época de la Guerra Fría, esto debido al desarrollo de nuevos componentes en el área de electrónica, principalmente con el desarrollo del transistor en 1947 en los Laboratorios Bell por parte de los Drs. William Shockley, John Bardeen y Walter H. Brattain, lo cual dió pie al diseño y construcción de sistemas electrónicos mucho más robustos y veloces [3].

El uso de los UAV se intensificó mucho más a lo largo de la década de 1990 durante los conflictos bélicos en Europa Oriental y Medio Oriente, por parte del Ejército de Estados Unidos, el cual hizo uso de los modelos *RQ-1L* y *RQ-1A* para la obtención de video en tiempo real de áreas en conflicto. El modelo *RQ-1A* se puede observar en la Figura 1.1.



Figura 1.1: UAV *RQ-1A* utilizado por el Ejército de Estados Unidos

### 1.1.1. Clasificación de los UAV

La clasificación normalmente adoptada para los UAV se centra en el tipo de configuración que posee el vehículo, teniendo principalmente:

- Ala fija
- Ala rotativa
- Ala libre

- Ala/rotor basculante
- Tailsitter

Las clasificaciones de ala fija y ala rotativa actualmente son las más utilizadas, esto debido a las ventajas que presenta cada una. Los UAV de ala fija presentan gran confiabilidad y tiempos de vuelo prolongados, sin embargo, no son capaces de realizar vuelo en modo *hover* ni vuelo a bajas velocidades. Por otra parte, las plataformas de ala rotativa son capaces de realizar despegue y descenso de manera vertical y vuelo en modo *hover*, no obstante, presentan limitaciones como vuelo a baja velocidad y poca resistencia durante vuelos en dirección horizontal [2].

## 1.2. Vuelo en formación

El control cooperativo y el control de formación de múltiples vehículos consiste en llevar a cabo cierta tarea o aplicación mediante el uso coordinado de diversos vehículos autónomos, denominados "agentes", los cuales afectan su dinámica de acuerdo al comportamiento del resto de los agentes dentro de la formación, esto con la finalidad de cumplir la aplicación a la que se encuentran destinados de la mejor manera posible. Lo anteriormente descrito es logrado a partir del acoplamiento de los estados dinámicos de los agentes por medio de una ley de control común [1].

Hoy en día, las aplicaciones en donde se hace el uso de múltiples vehículos autónomos se centran en exploración y mapeo de estructuras, operaciones militares, transporte de cargas, operaciones de búsqueda y rescate, etc [2]. De acuerdo a [1] se consideran cinco principales topologías para el vuelo en formación de múltiples UAV, las cuales corresponden con:

- MIMO (Multiple-Input and Multiple-Output)
- Líder-Seguidor
- Estructura Virtual
- Control Basado en Comportamiento
- Control Basado en Pasividad

### 1.2.1. Configuración MIMO (Multiple-Input and Multiple-Output)

En este tipo de topología, la formación es abordada como un sistema con múltiples entradas y múltiples salidas, teniendo por ende un sistema dinámico que representa el comportamiento de la planta en general a partir del cual se diseña una ley de control de formación común que lleve los diversos estados dinámicos de cada agente a un estado deseado [1] [2].

### 1.2.2. Configuración Líder-Seguidor

También llamada 'Configuración Maestro-Esclavo', dentro de este tipo de arquitectura uno de los UAV que integra la formación es designado como "Líder", mientras el resto de los agentes adoptan los roles de "Seguidores", los cuales deben de seguir la orientación y la posición del líder con una cierta compensación [2]. Debido a lo anteriormente asentado,

las posiciones deseadas de los seguidores se definen de manera relativa al estado actual del líder.

En esta configuración se emplea un arreglo jerárquico de controles para los distintos vehículos, con los cuales la trayectoria de referencia de la formación se encuentra definida por los movimientos del líder, mientras que la estabilidad de la estructura interna de la formación se encuentra relacionada y garantizada a través de la estabilidad de los distintos agentes [1] [4].

### **1.2.3. Configuración en Estructura Virtual**

Dentro de este tipo de arquitectura, cada agente dentro de la formación se considera como un elemento de una estructura mucho más grande, en donde el movimiento de la misma se da a partir de la implementación de esquemas de control para cada uno de los agentes de manera individual, siguiendo sus trayectorias de referencia. Los movimientos de la estructura virtual consideran los casos de contracción y expansión generados a partir de los movimientos angulares y traslacionales de los agentes que conforman a la misma [1] [2].

### **1.2.4. Configuración de Control Basado en Comportamiento**

En este tipo de configuración, la salida de diversos controles son combinadas con la finalidad de obtener diversos comportamientos por parte de la formación, teniendo como un comportamiento, la realización de cierta tarea, como puede ser evasión de colisiones, evasión de obstáculos, centrado de formación, etc., las cuales pueden ser realizadas de manera individual o en conjunto. En resumen, este tipo de control se basa en la idea de que al agregar acciones de control para comportamientos o tareas individuales, es posible realizar la tarea general [1] [2].

### **1.2.5. Configuración de Control Basado en Pasividad**

En este tipo de arquitectura, es posible descomponer el sistema conformado por múltiples agentes en subsistemas pasivos, cada uno encargado de distintas tareas dentro de la formación, entre las que destacan principalmente [2] [5]:

- Mantener a los diversos agentes dentro de la formación
- Realizar la traslación del sistema a partir de la consideración de la dinámica actual del mismo

## **1.3. Sistema de Visión Artificial**

La visión artificial se centra en extraer, analizar y entender información útil obtenida a través de una o más imágenes, esto con el fin de dotar a un sistema computacional de la capacidad de analizar y reconocer su entorno de una manera visual similar a cómo lo realizan los seres vivos. Se basa en el desarrollo de una base teórica y algorítmica para obtener un entendimiento visual automático [6].

### 1.3.1. Detección de obstáculos

La detección de un objetivo por medio de un sistema de visión se simplifica en gran manera por medio de un sistema de visión estereoscópica, en donde se cuenta con un juego de cámaras de video/fotográficas de las cuales se obtienen dos o más imágenes y se estima un modelo 3D de la escena al encontrar un conjunto de píxeles que coincidentes en el conjunto de imágenes, para posteriormente convertir sus posiciones en 2D a posiciones en 3D, en donde se cuenta con información de una profundidad estimada de los objetos dentro de la escena ([7], Cap. 11) [8]. La gran desventaja que presentan este tipo de sistema de visión es el gran costo computacional empleado para el procesamiento del algoritmo de estimación.

Por otro lado, existen métodos de estimación de un objetivo que emplean una sola cámara de video y hacen uso de información previamente conocida del entorno para estimar la distancia a la que la cámara se encuentra con respecto al objetivo y de esta manera realizar la detección del mismo; la principal desventaja que presentan estos métodos es el que deben de contar con información previa para realizar su procesamiento, por lo cual no presentan resultados aceptables ante un posible cambio de escena. A su vez, el uso de puntos característicos dentro de la escena y el uso de la información de los mismos es una opción viable para la detección de un posible objetivo [?].

## 1.4. Objetivo general

Vuelo coordinado de múltiples UAV del tipo cuadri-rotor para evasión de obstáculos mediante una estrategia de control basada en visión monocular para la generación de una trayectoria de evasión.

## 1.5. Objetivos específicos

- Desarrollo de una plataforma de comunicación para retroalimentación de la información de vuelo de cada *drone* dentro de la formación a una estación en tierra para monitoreo en tiempo real.
- Construcción de una plataforma del tipo cuadri-rotor basado en un autopiloto *Pixhawk* para la comprobación de las estrategias de control implementadas en el presente trabajo.
- Obtención de datos provenientes de la tarjeta *Pixhawk* a través de la tarjeta *Raspberry Pi 3* y la tarjeta *Odroid* e implementación de la comunicación entre cada una de ellas y el módulo de radiofrecuencia para el envío a la *GCS*, del inglés *Ground Control Station*, en modo Half-Duplex.
- Diseño de una estrategia de control para la estabilización del vehículo cuadri-rotor.
- Diseño de una estrategia de control para el seguimiento del agente "Líder" por parte del agente "Seguidor"

## 1.6. Planteamiento del problema

Dentro de la presente tesis se plantean tres problemas a resolver, los cuales son:

- Estabilización del prototipo cuadri-rotor en orientación, altura y posición en tiempo-real en ambientes exteriores.
- Vuelo en formación de múltiples *UAV* del tipo cuadri-rotor.
- Detección y evasión de obstáculos

El tópico de estabilización del cuadrirotor pretende ser abordado mediante un controlador del tipo *PID* discreto, el cual por sus características lineales es capaz de compensar pequeñas perturbaciones alrededor del punto de equilibrio (referencia) que para este caso es cero (modo estacionario).

Para el problema concerniente al vuelo en formación, se plantea el uso de la configuración "Líder-Seguidor" dentro del cual se cuenta con un controlador encargado de mantener la formación de cada agente dentro de la misma considerando el estado actual del líder y la trayectoria del mismo, para mantener una compensación de posición del seguidor con respecto al líder.

Para la cuestión de la detección y evasión de un obstáculo, se plantea el uso de un algoritmo de visión artificial basado en un sistema de visión monocular.

## 1.7. Motivación

La detección y evasión de obstáculos en tiempo real para cuadri-rotos representa un problema fundamental para la navegación autónoma de los mismos. El uso de sistemas de visión estereoscópica o en su defecto, monocular, constituyen una solución viable para la detección de objetos dentro de la línea de vista del drone, en comparación con los resultados obtenidos con sensores analógicos/digitales (sensores ultrasónicos, LASER, etc), esto debido a la amplia información que puede ser obtenida por medio de una cámara digital.

---

# Modelado matemático del prototipo cuadri-rotor

---

## 2.1. Descripción del cuadri-rotor

El UAV utilizado como base para el siguiente trabajo consiste en un helicóptero miniatura en configuración de cuadri-rotor, es decir, teniendo sus cuatro rotores coplanarios, utilizando la configuración en "X", en donde cada par de rotores se mueven en sentido horario y el siguiente par en sentido antihorario, anulando de esta manera los efectos giroscópicos y los torques aerodinámicos generados [?].

El movimiento del vehículo cuadri-rotor se obtiene variando las velocidades de los cuatro rotores que conforman al mismo, obteniendo el movimiento hacia adelante, disminuyendo la velocidad de los rotores delanteros y aumentando la velocidad de los rotores traseros y de manera contraria para obtener el movimiento de retroceso; de esta manera se obtiene una variación en el ángulo de cabeceo (*pitch*,  $\theta$ ). A su vez, para obtener el movimiento lateral, es necesario realizar el mismo procedimiento de variación de velocidad con los rotores laterales, dependiendo de la dirección deseada, obteniendo así una variación en el ángulo de alabeo (*roll*,  $\phi$ ). Para obtener la variación en el ángulo de guiñada (*yaw*,  $\psi$ ), es necesario acelerar ambos rotores que se encuentren girando en sentido horario y desacelerar los dos rotores que se encuentren girando en sentido anti-horario y viceversa [?] [11].

## 2.2. Modelo matemático del cuadri-rotor

El modelado matemático del vehículo aéreo se realizará mediante el formalismo de Euler-Lagrange, en configuración "+", en donde se considerará al vehículo como un cuerpo rígido en el espacio, el cual se encuentra sometido a una fuerza de empuje y tres momentos o pares. En la Figura 3.1 se puede observar el diagrama de cuerpo libre del cuadri-rotor.

## 2. MODELADO MATEMÁTICO DEL PROTOTIPO CUADRI-ROTOR

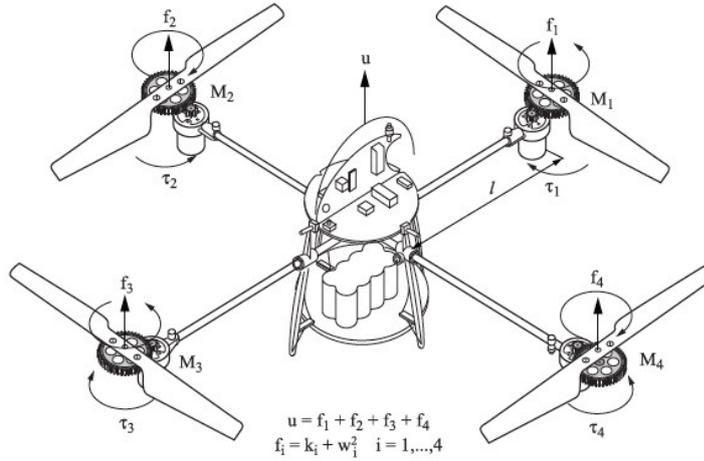


Figura 2.1: Diagrama de cuerpo libre del cuadri-rotor

Debido a que el cuadri-rotor es considerado como un cuerpo rígido, cuenta con un sistema de coordenadas fijo al mismo, llamado "Marco del Cuerpo", el cual coincide en su origen con el centro de masa del vehículo. Dicho sistema se puede definir como  $B = (\vec{x}_L, \vec{y}_L, \vec{z}_L)$  en donde  $\vec{x}_L$  corresponde con la dirección normal de ataque del cuadri-rotor,  $\vec{y}_L$  es ortogonal a  $\vec{x}_L$  y  $\vec{z}_L$  se encuentra ortogonal a ambos y de manera ascendente, teniendo de esta manera un sistema dextrógiro.

A su vez, se considera un sistema de coordenadas inercial I el cual se encuentra fijo a la tierra. Una representación de ambos marcos de referencia se puede observar en la Figura 3.2.

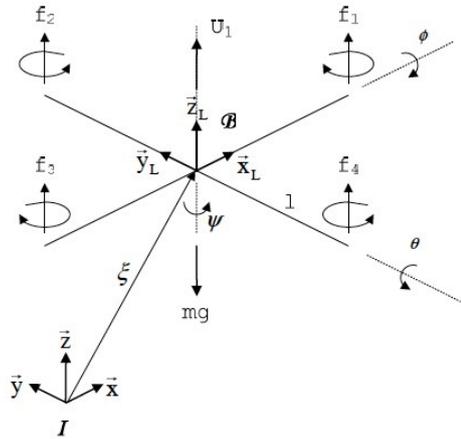


Figura 2.2: Marcos de referencia inercial y del cuerpo

Las coordenadas generalizadas del sistema se encuentran descritas por

$$q = (x, y, z, \psi, \theta, \phi) \in R^6 \quad (2.1)$$

en donde  $(x, y, z)$  representan la posición del centro de masa del cuadri-rotor con respecto al marco inercial I y  $(\psi, \theta, \phi)$  representan la orientación del vehículo aéreo, el cual es descrito

## 2.2. MODELO MATEMÁTICO DEL CUADRI-ROTOR

---

mediante los ángulos de yaw, pitch y roll, respectivamente. De acuerdo a lo anterior, es posible aislar coordenadas traslacionales y rotacionales, dentro del cuadri-rotor, dadas por

$$\xi = (x, y, z) \in \mathbb{R}^3 \quad , \quad \eta = (\psi, \theta, \phi) \in \mathcal{R}^3 \quad (2.2)$$

Dado lo anterior, es posible obtener la energía cinética traslacional y rotacional, descritas por

$$E_{c-Trans} \triangleq \frac{m}{2} \dot{\xi}^T \dot{\xi} \quad (2.3)$$

$$E_{c-Rot} \triangleq \frac{1}{2} \dot{\eta}^T \mathbb{J} \dot{\eta} \quad (2.4)$$

en donde

- $m$  Masa del cuadri-rotor
- $\mathbb{J} \rightarrow$  Matriz de inercia del cuadri-rotor

La energía potencial considerada viene dada por

$$E_p = mgz \quad (2.5)$$

La formulación de Euler-Lagrange se encuentra basada en el concepto de energía mecánica, en donde se consideran tanto la energía cinética como la energía potencial del mecanismo, la cual se encuentra establecida como

$$\Gamma = \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}} - \frac{\partial \mathcal{L}}{\partial q} \quad (2.6)$$

en donde

- $\Gamma \rightarrow$  Fuerzas y/o pares generalizados y la fuerza traslacional debido a las entradas de control
- $q \rightarrow$  Coordenadas generalizadas
- $\dot{q} \rightarrow$  Primera derivada de las coordenadas generalizadas
- $\mathcal{L} \rightarrow$  Lagrangiano

Representando la ecuación de Euler-Lagrange de forma matricial [?], se tiene

$$\begin{bmatrix} \frac{d}{dt} \left[ \frac{\partial L_{tras}}{\partial \dot{\xi}} \right] - \frac{\partial L_{tras}}{\partial \xi} & 0 \\ 0 & \frac{d}{dt} \left[ \frac{\partial L_{rot}}{\partial \dot{\eta}} \right] - \frac{\partial L_{rot}}{\partial \eta} \end{bmatrix} = \begin{bmatrix} F_{\xi} \\ \tau \end{bmatrix} \quad (2.7)$$

El Lagrangiano viene dado por

$$\mathcal{L} = E_{c-Total} - U \quad (2.8)$$

en donde la energía cinética total se define como

$$E_{c-Total} = E_{c-Trans} + E_{c-Rot} \quad (2.9)$$

Debido a que la entrada de control corresponde con la suma de la fuerza empuje generada por cada motor, se tiene que

$$u = f_1 + f_2 + f_3 + f_4 \quad (2.10)$$

en donde la fuerza de empuje generada por cada motor se define como

$$f_i = k_i w_i^2 \quad i = 1, \dots, 4 \quad (2.11)$$

donde

- $k_i \rightarrow$  Constante mayor a cero que depende de la densidad del aire, el radio y forma de la hélice, entre otros factores
- $w_i \rightarrow$  Velocidad angular el  $i$ -ésimo motor [9]

Una vez que se ignoran las fuerzas de pequeña magnitud presentes en el cuerpo, se tiene

$$\hat{F} = \begin{bmatrix} 0 \\ 0 \\ u \end{bmatrix} \quad (2.12)$$

la cual es la fuerza neta aplicada al cuadri-rotor debido únicamente a la entrada de control  $u$ .

Considerando la multiplicación de una matriz de transformación que represente la orientación del vehículo por el vector de fuerzas netas obtenido, se llega a la la fuerza traslacional antes mencionada.

Los torques generalizados para las variables de orientación  $(\psi, \theta, \phi)$ , vienen dados por

$$\tau \triangleq \begin{bmatrix} \tau_\psi \\ \tau_\theta \\ \tau_\phi \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^4 \tau_{M_i} \\ (f_2 - f_4)l \\ (f_3 - f_1)l \end{bmatrix} \quad (2.13)$$

donde

- $l \rightarrow$  Distancia de los motores al centro de gravedad
- $\tau_{M_i} \rightarrow$  Unión producida por el motor  $M_i$  [9]

Analizando el Lagrangiano, se puede observar que no contiene términos cruzados dentro de la energía cinética que combinen  $\dot{\xi}$  y  $\dot{\eta}$ , la ecuación de Euler-Lagrange se puede separar en dos dinámicas, una para las coordenadas traslacionales y otra para las coordenadas rotacionales.

Para el movimiento de traslación, las ecuaciones de Euler-Lagrange son

$$\frac{\partial L_{tras}}{\partial \dot{\xi}} = m\dot{\xi} \quad (2.14)$$

$$\frac{d}{dt} \left[ \frac{\partial L_{tras}}{\partial \dot{\xi}} \right] = m\ddot{\xi} \quad (2.15)$$

$$\frac{\partial L_{tras}}{\partial \xi} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \quad (2.16)$$

Con lo que se tiene

$$F_{\xi} = \begin{bmatrix} m\ddot{x} \\ m\ddot{y} \\ m\ddot{z} - mg \end{bmatrix} \quad (2.17)$$

Para cuestiones prácticas, es posible reescribir la expresión anterior como

$$F_{\xi} = m\ddot{\xi} + \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \quad (2.18)$$

Para la parte rotacional, la formulación de Euler-Lagrange corresponde con

$$\frac{d}{dt} \left[ \frac{\partial L_{rot}}{\partial \dot{\eta}} \right] - \frac{\partial L_{rot}}{\partial \eta} = \tau \quad (2.19)$$

Desarrollando

$$\frac{d}{dt} \left[ \dot{\eta}^T \mathbb{J} \frac{\partial \dot{\eta}}{\partial \dot{\eta}} \right] - \frac{1}{2} \frac{\partial}{\partial \eta} (\dot{\eta}^T \mathbb{J} \dot{\eta}) = \tau \quad (2.20)$$

Con lo que se tiene

$$\mathbb{J} \ddot{\eta} + \dot{\mathbb{J}} \dot{\eta} - \frac{1}{2} \frac{\partial}{\partial \eta} (\dot{\eta}^T \mathbb{J} \dot{\eta}) = \tau \quad (2.21)$$

en donde el segundo y tercer término corresponden con el vector de Coriolis, el cual contiene los términos giroscópicos y centrífugos asociados con el vector  $\eta$  y la matriz de inercia  $\mathbb{J}$ . Después de factorizar el término  $\dot{\eta}$ , se puede reescribir la ecuación anterior como

$$\mathbb{J} \ddot{\eta} + C(\eta, \dot{\eta}) \dot{\eta} = \tau \quad (2.22)$$

Reescribiendo las ecuaciones ?? y ?? en función del vector de estado, se tiene

$$m\ddot{\xi} = u \begin{bmatrix} -\sin \theta \\ \cos \theta \sin \psi \\ \cos \theta \cos \psi \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \quad (2.23)$$

$$\mathbb{J} \ddot{\eta} = -C(\eta, \dot{\eta}) \dot{\eta} + \tau \quad (2.24)$$

Se propone la siguiente entrada de control, en donde se asume que la matriz de Coriolis es conocida, para lo cual se tiene:

$$\tau = C(\eta, \dot{\eta}) \dot{\eta} + \mathbb{J} \tilde{\tau} \quad (2.25)$$

Sustituyendo el control anterior propuesto en (1.26), se obtiene:

$$\tilde{\tau} = \begin{bmatrix} \tilde{\tau}_\psi \\ \tilde{\tau}_\theta \\ \tilde{\tau}_\phi \end{bmatrix} \quad (2.26)$$

por lo cual

$$\ddot{\eta} = \tilde{\tau} \quad (2.27)$$

Despreciando el término de las fuerzas de Coriolis, de la matriz de inercia y considerando la presencia de ángulos muy pequeños, se llega al modelo simplificado del cuadri-rotor, el cual viene dado por

$$\begin{aligned} m\ddot{x} &= -u \sin \theta \\ m\ddot{y} &= u \cos \theta \sin \psi \\ m\ddot{z} &= u \cos \theta \cos \psi - mg \\ \ddot{\psi} &= \tilde{\tau}_\psi \\ \ddot{\theta} &= \tilde{\tau}_\theta \\ \ddot{\phi} &= \tilde{\tau}_\phi \end{aligned} \quad (2.28)$$

en donde

- $x \rightarrow$  Coordenada  $x$  en el plano horizontal
- $y \rightarrow$  Coordenada  $y$  en el plano horizontal
- $z \rightarrow$  Posición vertical
- $\psi \rightarrow$  Ángulo de guiñada alrededor del eje  $Z$
- $\theta \rightarrow$  Ángulo de cabeceo alrededor del eje  $Y$
- $\phi \rightarrow$  Ángulo de alabeo alrededor del eje  $X$

Las entradas de control  $u$ ,  $\tilde{\tau}_\psi$ ,  $\tilde{\tau}_\theta$  y  $\tilde{\tau}_\phi$  corresponden al empuje total generado por los motores y los nuevos momentos angulares.

---

## Plataforma experimental

---

Las plataformas empleadas para el presente trabajo de tesis consisten de un conjunto de vehículos de ala rotativa de tipo cuadri-rotor, los cuales cuentan con la instrumentación (aviónica) necesaria para realizar la tarea designada, y posteriormente serán descritos a detalle. A su vez, a lo largo de este capítulo se abordará el tema concerniente a la GCS desarrollada en el framework *Qt*, la cual es la encargada de recibir información relacionada con el estado de cada plataforma, y de enviar instrucciones específicas a los *UAV* detectados dentro de la red.

### 3.1. Interfaz de comunicación RS-485 y distribución de energía

El prototipo cuenta con un módulo de potencia encargado de proveer de energía a los motores *BLDC*, del inglés *Brushless Direct Current*, utilizados para la acción de propulsión del cuadri-rotor y a su vez, llevar a cabo la regulación del voltaje proveniente de la batería Li-Po a un nivel de tensión menor, y de esta manera suministrar la energía adecuada al autopiloto.

Debido a que la alimentación de los actuadores antes mencionados y la etapa de control se encuentran sobre un mismo módulo de suministro de energía, esta última se encuentra susceptible a ser afectada por el ruido de alta frecuencia generado por los motores *BLDC*, motivo por el cual la *PCB*, del inglés *Printed Circuit Board*, diseñada cuenta con un filtro pasa bajas *LC* a la entrada y cuya frecuencia de corte se calculó a 900 Hz, esto con el fin de suprimir la mayor parte de dicha señal ruidosa, para posteriormente utilizar la nueva señal filtrada para proveer de energía al autopiloto, a la *SBC*, del inglés *Single Board Computer* y a un regulador de voltaje de tipo *LDO*, del inglés *Low Drop Out*, para energizar el transductor de niveles de voltaje para la comunicación *RS-485*.

El esquemático y el diseño de la *PCB* antes descrita se encuentra realizado en el software de *EDA*, del inglés *Electronic Design Automation*, *Altium Designer*, cuyo modelo 3D y el ensamble final de la misma se puede observar en las Figuras 3.1 y 3.2, respectivamente.

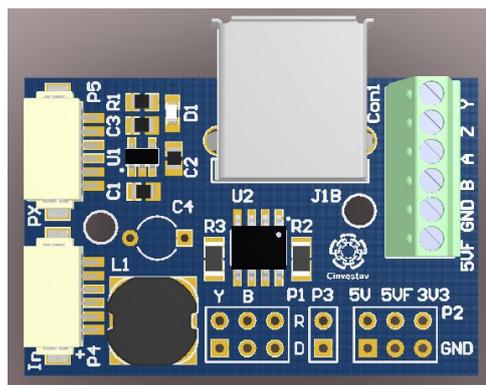


Figura 3.1: Modelo 3D de la PCB

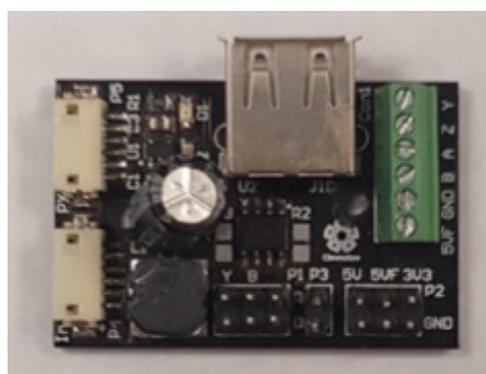


Figura 3.2: PCB ensamblada

## 3.2. Autopiloto

El autopiloto empleado dentro del prototipo consiste en un controlador de vuelo *Pixhawk PX4*, el cual contiene los elementos necesarios para obtener los estados (orientación y posición) del prototipo. El autopiloto utilizado dentro de la plataforma se puede observar en la Figura 3.3.



Figura 3.3: Autopiloto *Pixhawk PX4*

Las características lógicas del autopiloto son las siguientes:

### 3.3. ESQUEMA DE LA PLATAFORMA EXPERIMENTAL: LÍDER-SEGUIDOR

---

- Procesador ARM Cortex M4 de 32 bits *STM32F427* de la empresa *ST Microelectronics*
- Co-procesador de 32 bits *STM32F103* de la empresa *ST Microelectronics* para posibles fallas

Los sensores con los que se cuenta son:

- Giroscopio de 16 bits *L3GD20H* de la empresa *ST Microelectronics*
- Acelerómetro y Magnetómetro de 14 bits *LSM303D* de la empresa *ST Microelectronics*
- Acelerómetro y Giroscopio de 3 ejes *MPU6000* de la empresa *Invensense*
- Barómetro *MS5611* de la empresa *MEAS*

## 3.3. Esquema de la plataforma experimental: Líder-Seguidor

### 3.3.1. Agente "Líder"

Dentro de la plataforma que funge como "Líder" a lo largo de la formación, se cuenta con una *SBC*, del inglés *Single Board Computer*, *Odroid XU4*, la cual contiene precargado un sistema operativo basado en la distro de *Linux*, *Ubuntu*, sobre el cual se ejecutan diversos scripts de *Python*. Por otro lado, la principal tarea de la *SBC* dentro de esta plataforma consiste en ejecutar el algoritmo de visión artificial encargado de detectar un posible obstáculo dentro del rango de visión del prototipo a lo largo de la trayectoria del mismo.

A su vez, la misma sirve como punto intermedio entre la *GCS* y el autopiloto *Pixhawk*, esto debido a que recibe información proveniente del autopiloto vía *USB*, del inglés *Universal Serial Bus*, y posteriormente la redirige al módulo de radiofrecuencia con el cual cuenta el prototipo, para su recepción y análisis tanto en la *GCS* como por parte del agente "Seguidor". El autopiloto recibe información de la *GCS* siguiendo el mismo razonamiento pero en sentido inverso.

Las especificaciones de la *SBC Odroid* son las siguientes:

- Microprocesador Samsung Exynos-5422 Cortex-A15 y Cortex-A7 de ocho núcleos
- *GPU* Mali-T628 MP6 a 600 MHz
- 2 Gb de RAM LPDDR3
- Puerto de almacenamiento dual por medio de tarjetas Micro-SD o tarjetas eMMC 5.0
- Puerto *HDMI* tipo A
- 2 puertos *USB* 3.0
- 1 puerto *USB* 2.0
- Puerto Ethernet

En la figura 3.4 se puede observar la *SBC* antes descrita



Figura 3.4: Tarjeta *Odroid XU4*

El diagrama de flujo del funcionamiento e implementación de la plataforma experimental que funge como "Líder", considerando los distintos elementos que las conforman y sus respectivas interconexiones, se puede observar en la Figura 3.5.

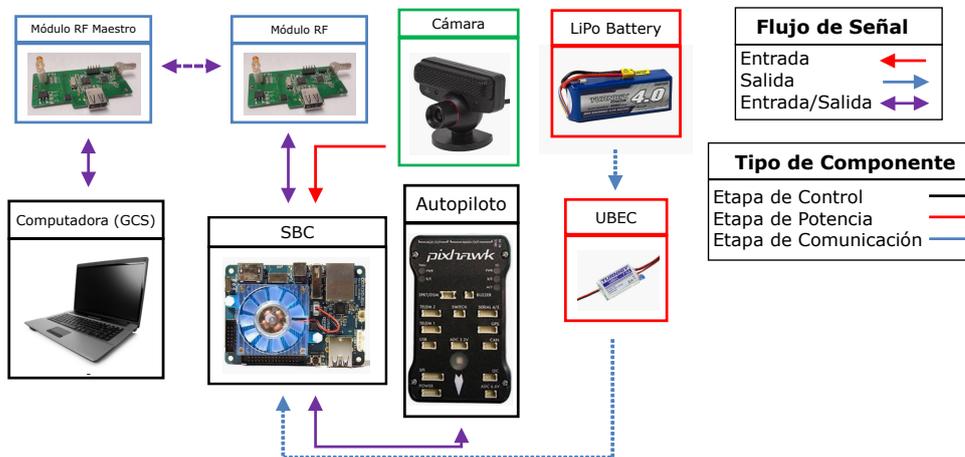


Figura 3.5: Diagrama de flujo del funcionamiento e implementación de la plataforma "Líder"

Dentro del mismo se omiten ciertos componentes como lo son:

- Motores BLDC
- *GPS*
- *ESC*, del inglés *Electronic Speed Controller*

### 3.3. ESQUEMA DE LA PLATAFORMA EXPERIMENTAL: LÍDER-SEGUIDOR

---

debido a que son componentes básicos que debe poseer cualquier sistema con las características que cuenta el prototipo y no es necesaria su inclusión dentro del diagrama.

#### 3.3.2. Agente "Seguidor"

Para el caso de la plataforma que actúa como "Seguidor" dentro del escuadrón, se cuenta con una *SBC Raspberry Pi 3*, la cual al igual que el agente líder, cuenta con un sistema operativo basado en una distro de *Linux*, llamado *Raspbian*, la cual se encuentra basada en *Debian*, sobre la cual, al igual que en la plataforma descrita anteriormente, se ejecutan diversos scripts de *Python*, los cuales se encargan de procesar la información recibida por parte del agente "Líder" para posteriormente ser transmitida al autopiloto y de esta manera variar la dinámica de la plataforma través de la acción de control.

La información proveniente del autopiloto es redirigida por la *SBC* al módulo de radiofrecuencia con el cual cuenta el prototipo, a través de la *PCB* encargada de la transducción de niveles lógicos.

Del mismo modo, las especificaciones de la tarjeta *Raspberry Pi 3* son las siguientes:

- SoC Broadcom BCM2837
- CPU Quad-core ARM Cortex A53 a 1.2 GHz
- GPU Broadcom VideoCore IV a 400 MHz
- 1 Gb de SDRAM LPDDR2-900
- Puerto de almacenamiento por medio de tarjetas Micro-SD
- Puerto *HDMI* tipo A
- 4 puertos *USB 2.0*
- Puerto Ethernet

En la figura 3.6 se puede observar la *SBC* antes descrita

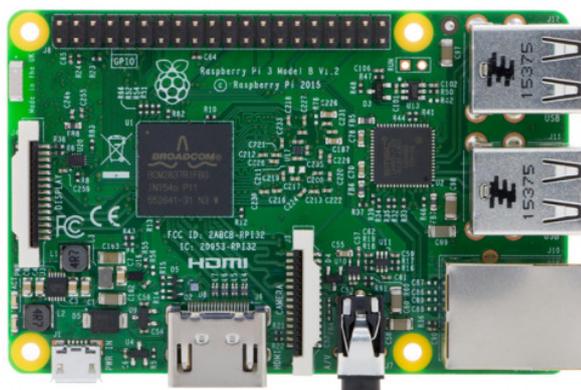


Figura 3.6: Tarjeta *Raspberry Pi 3*

El diagrama de flujo del funcionamiento e implementación para la plataforma que actúa como "Seguidor" dentro de la formación, se puede observar en la Figura 3.7.

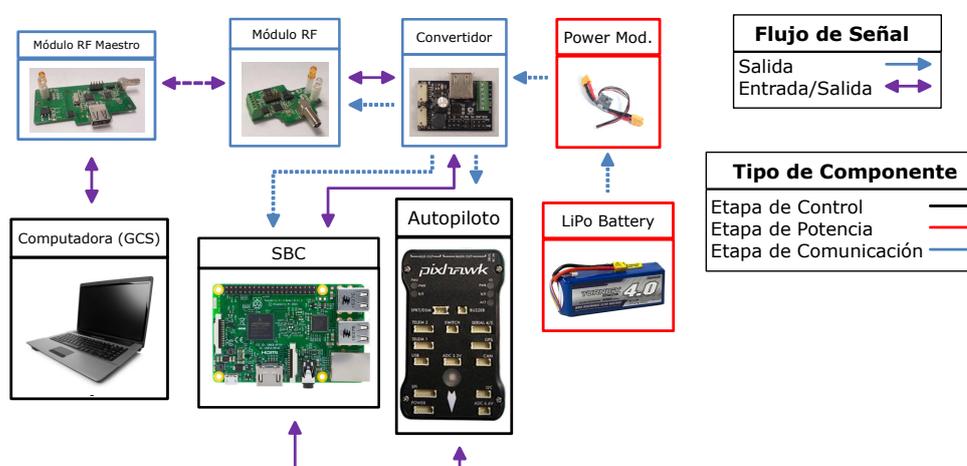


Figura 3.7: Diagrama de flujo del funcionamiento e implementación de la plataforma "Seguidor"

Al igual que para el diagrama de flujo de la plataforma "Líder", algunos componentes se omiten debido a la misma razón antes descrita.

### 3.3.3. ROS

Para llevar a cabo el procesamiento en paralelo de las distintas tareas de las cuales se encuentran encargadas las *SBC*, como son la recepción de información por parte de la *GCS*, el envío de comandos al autopiloto, el algoritmo de visión artificial, así como el envío bilateral de información entre diversos cuadri-rotors, se hace uso del esquema basado en nodos por parte del middleware *ROS*, del inglés *Robot Operating System*.

Como se menciona en la página y documentación oficial de *ROS*, soportada por la *Open Source Robotics Foundation*, "ROS es un framework flexible utilizado para escribir software para robots. Es una colección de herramientas, bibliotecas y convenciones que buscan simplificar la tarea de crear comportamiento robótico complejo y robusto a lo largo de una gran variedad de plataformas robóticas" [12].

## 3.4. Módulo de radiofrecuencia

Para la comunicación entre la *GCS* con los diversos agentes dentro de la formación y a su vez, entre ellos mismos, se hace uso de un módulo de radiofrecuencia diseñado en la Sección de Comunicaciones del Departamento de Ingeniería Eléctrica del CINVESTAV, el cual se encuentra basado en el *IC CC2530F256* de la empresa *Texas Instruments*, cuyas principales especificaciones se plantean en la Tabla 3.1.

### 3.4. MÓDULO DE RADIOFRECUENCIA

---

Banda de operación	2.4 GHz
Potencia de salida máxima	4.5 dBm
Voltaje de alimentación	2-3.6 V
Microcontrolador	Núcleo 8051
Protocolo de Acceso al Medio soportado	CSMA/CA

Tabla 3.1: Especificaciones del IC *CC2530F256*

Debido al propósito principal con el cual fueron diseñados los módulos de radiofrecuencia, se cuenta con un receptor/transmisor *Full-Duplex* para comunicación alámbrica por medio del protocolo *RS-485* dentro de la *PCB*, con matrícula *SN65HVD30*. El módulo de radiofrecuencia descrito se puede observar en la Figura 3.8.



Figura 3.8: Módulo de radiofrecuencia

A su vez, el módulo de radiofrecuencia cuenta con una antena dipolo *GW.59*, la cual es idónea para su uso en aplicaciones inalámbricas. Dicha antena, así como sus principales especificaciones, se pueden observar en la Figura 3.9 y en la Tabla 3.2, respectivamente.



Figura 3.9: Antena dipolo *GW.59*

Banda de operación	2.4 GHz
Ganancia	2.04 dBi
Eficiencia	75 %
Radiación	Omnidireccional
Potencia	1 W
Impedancia	50 $\Omega$

Tabla 3.2: Especificaciones de la antena dipolo GW.59

### 3.5. Interfaz de Control en Tierra

Para la recepción y envío de información hacia los agentes dentro de la formación por parte del operador, se cuenta con una *GCS* diseñada y programada por medio del framework *Qt*, la cual es capaz de reconocer todos los módulos de radiofrecuencia cercanos y diferenciar entre los que corresponden a un cuadri-rotor y a los que puedan ser utilizados por otra *GCS*, a partir de la dirección *MAC*, del inglés *Medium Access Control*, del dispositivo. Una vez reconocido y clasificado cada uno de los módulos, es posible desplegar la información enviada por ellos a la *GCS*.

La acciones de recepción y envío de los paquetes de datos entre la *GCS* y los agentes se realiza por medio de un módulo de radiofrecuencia similar al descrito en la sección 3.4, con la funcionalidad de poder comunicarse con una computadora por medio del protocolo *USB* a través del *IC TUSB3410I* de la empresa *Texas Instruments*. El módulo de radiofrecuencia antes mencionado se puede observar en la Figura 3.5.

Figura 3.10: Módulo de radiofrecuencia empleado en la *GCS*

En la Figura 3.11 se puede observar la ventana principal de la interfaz gráfica, en la cual es posible configurar las características requeridas por el puerto serial asociado al módulo de radiofrecuencia con el cual se realice la comunicación entre los diversos agentes y la *GCS*. Una vez realizada la configuración requerida, es posible llevar a cabo una búsqueda de los módulos de radiofrecuencia dentro de la red, para así desplegar la información asociada a los mismos.

### 3.5. INTERFAZ DE CONTROL EN TIERRA

---



Figura 3.11: Ventana de inicio

Una vez detectados los módulos de radiofrecuencia, se realiza la clasificación de los mismos de acuerdo a la información obtenida. De esta manera, como se puede observar en la Figura 3.12, se despliega el número de quadri-rotosres que se encuentren dentro de la red, de los cuales es posible desplegar la información asociada al estado de cada uno de ellos.



Figura 3.12: Ventana de Menú

A su vez, dentro de esta ventana se cuenta con el botón "Devices", por medio del cual es posible visualizar en una ventana desplegable los diversos nodos dentro de la red, tanto GCS o UAV, como se puede observar en la Figura 3.13.

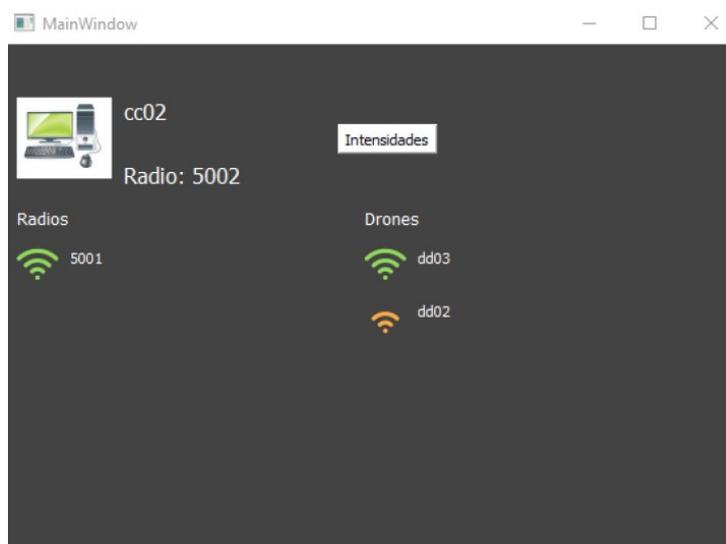


Figura 3.13: Ventana de Dispositivos dentro de la red

Por otro lado, por medio del botón "Console" es posible desplegar la ventana de Consola, mostrada en la Figura 3.14, en la cual se lleva un registro de las tramas de datos transmitidas entre los *UAV* y la *GCS* con la finalidad de contar con un registro de la actividad dentro de la red y en caso de presentarse alguna falla, poder analizar cuándo ocurrió y tener una idea general de lo sucedido.

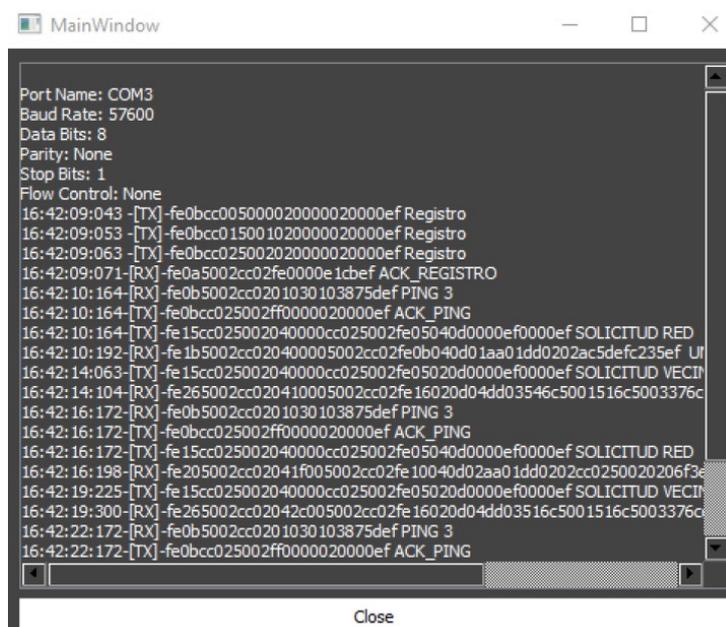


Figura 3.14: Ventana de Consola

Como se mencionó, una vez detectados los *UAV* dentro de la red, es posible seleccionar alguno de ellos para desplegar en una ventana emergente la información relacionada con su estado actual, como es orientación (ángulos de *yaw* ( $\psi$ ), *pitch* ( $\theta$ ) y *roll* ( $\phi$ )), posición (*X*, *Y* y *Z*), velocidad lineal (*X* y *Y*), coordenadas geodésicas (*Latitud* y *Longitud*) y número de

### 3.5. INTERFAZ DE CONTROL EN TIERRA

---

satélites localizados por parte del *GPS*. La ventana antes descrita se puede observar en la Figura 3.15.

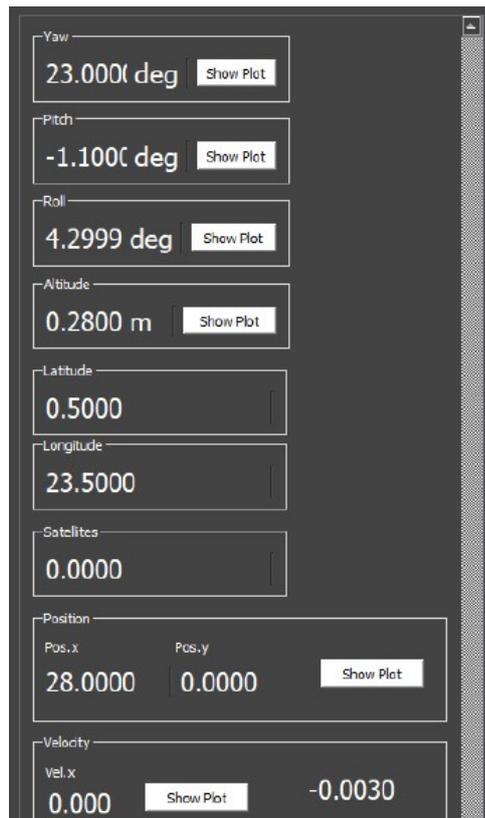


Figura 3.15: Ventana de Datos del estado del *UAV*

Dentro de la ventana de datos del *UAV* se cuenta con la opción de desplegar una gráfica en tiempo real de los mismos, los cuales a su vez, son almacenados en un *logger* a través de archivos *.txt* que cuentan con la fecha y la hora en la cual se realizó la captura de datos, con la finalidad de analizar posteriormente el comportamiento del *UAV* durante el vuelo. Ambas funcionalidades se pueden observar en las Figuras 3.16 y 3.17 respectivamente.

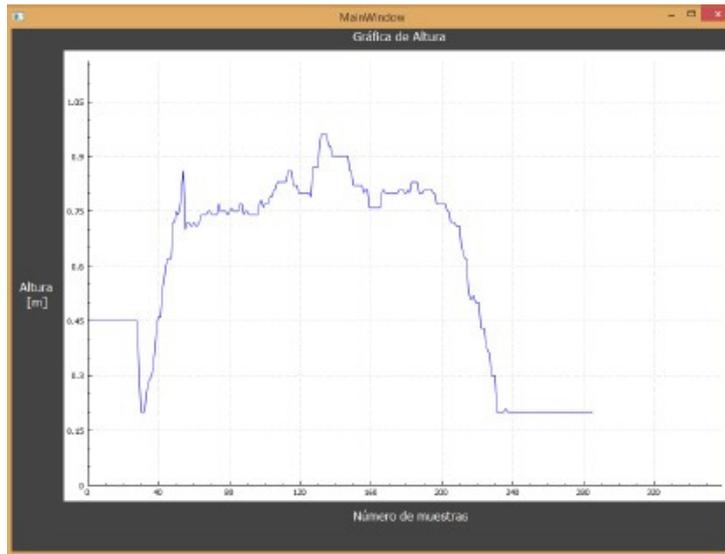


Figura 3.16: Gráfica en tiempo real de los datos de *pitch* ( $\theta$ )

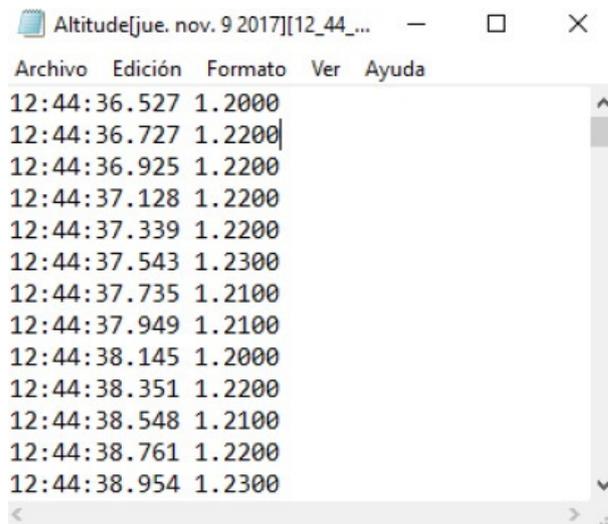


Figura 3.17: Ejemplo de un archivo *logger* .txt

---

# Estrategia de control

---

## 4.1. Procesamiento de señales

Una "señal" puede ser definida como una cantidad física que varía con relación a una o más variables independientes, como puede ser el tiempo, y que puede ser descrita matemáticamente como una función de dichas variables [13].

Las señales pueden ser clasificadas en dos categorías dependiendo de las características de las variables independientes con las que se encuentran relacionadas:

- Señales Continuas o Analógicas
- Señales Discretas

A su vez, los valores que puede adoptar una señal, tanto en tiempo continuo como en tiempo discreto, pueden ser continuos o discretos, dependiendo de si toma todos los valores posibles en un intervalo finito o infinito, o si toma valores que se encuentren dentro de un conjunto finito de valores, respectivamente. Con base en lo anterior, se llega a la definición de "señal digital", la cual corresponde a una señal en tiempo discreto que adquiere valores pertenecientes a un conjunto discreto.

### 4.1.1. Concepto de Frecuencia para señales en tiempo continuo y tiempo discreto

El concepto de frecuencia se encuentra relacionado con un tipo de movimiento armónico llamado "oscilación armónica", la cual puede ser descrita a partir de funciones sinusoidales. Dado que la frecuencia se encuentra relacionada de manera inversamente proporcional al tiempo, es de esperarse que la naturaleza del mismo (continuo o discreto) afecte a la frecuencia. Lo anterior puede ser ejemplificado a partir de la representación y las propiedades de las señales sinusoidales tanto en tiempo continuo como en tiempo discreto.

#### 4.1.1.1. Señales sinusoidales en tiempo continuo

Una oscilación armónica en tiempo continuo puede ser descrita a partir de

$$x_a(t) = A \cos(\Omega t + \theta) \quad -\infty < t < \infty \quad (4.1)$$

en donde

- $x_a(t) \rightarrow$  Señal analógica
- $A \rightarrow$  Amplitud de la senoide
- $\Omega \rightarrow$  Frecuencia en rad/s
- $\theta \rightarrow$  Fase en radianes

Dado que la frecuencia puede ser descrita como  $\Omega = 2\pi F$ , en donde  $F$  viene dada en Hertz (Hz), es posible reescribir 4.1 en terminos de  $F$  como

$$x_a(t) = A \cos(2\pi Ft + \theta) \quad -\infty < t < \infty \quad (4.2)$$

#### 4.1.1.2. Señales sinusoidales en tiempo discreto

Para el caso discreto, una señal sinusoidal puede expresarse como

$$x(n) = A \cos(\omega n + \theta) \quad -\infty < n < \infty \quad (4.3)$$

en donde

- $x(n) \rightarrow$  Señal discreta
- $A \rightarrow$  Amplitud de la senoide
- $n \rightarrow$  Número de muestra
- $\omega \rightarrow$  Frecuencia en rad/muestra
- $\theta \rightarrow$  Fase en radianes

Debido a que la frecuencia puede definirse como  $\omega = 2\pi f$ , 4.3 se convierte en

$$x_a(t) = A \cos(2\pi fn + \theta) \quad -\infty < n < \infty \quad (4.4)$$

### 4.1.2. Conversión Analógico-Digital

Debido a que la mayoría de las señales analizadas y utilizadas de manera práctica son señales analógicas, es necesario convertirlas a un formato digital, para posteriormente poder ser procesadas por medios digitales. A este proceso de conversión se le conoce como "conversión analógico-digital (A/D)" y se basa en el uso de conversores A/D. El procedimiento consta de 3 etapas:

1. Muestreo
2. Cuantificación
3. Codificación

### 4.1.3. Muestreo

Esta etapa corresponde a la conversión de una señal en tiempo continuo a una señal en tiempo discreto, la cual se obtiene al conseguir muestras de la señal en tiempo continuo en instantes de tiempo discreto. El tipo de muestreo utilizado más a menudo de manera práctica es el llamado "muestreo periódico", el cual puede ser descrito mediante la relación:

$$x(n) = x_a(nT) \quad -\infty < n < \infty \quad (4.5)$$

en donde

- $x(n)$  → Señal en tiempo discreto
- $x_a(t)$  → Señal en tiempo continuo
- $T$  → Periodo o intervalo de muestreo

El muestreo periódico de una señal analógica a una señal en tiempo discreto se puede observar en la Figura 4.1.

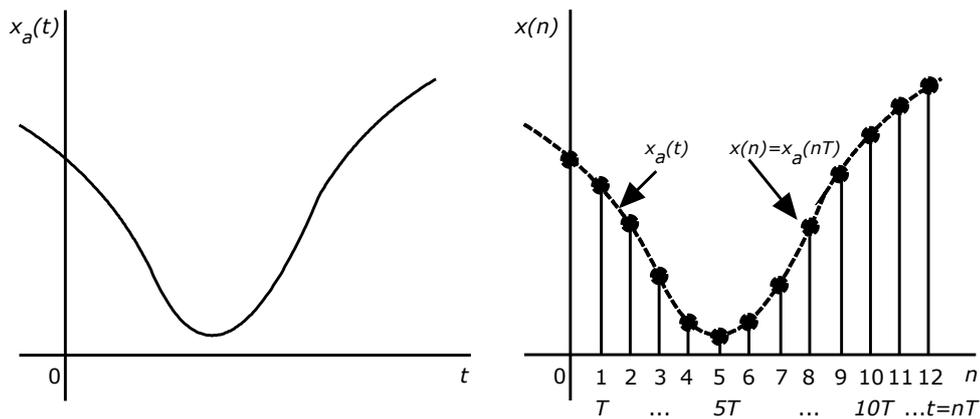


Figura 4.1: Muestreo periódico de una señal en tiempo continuo

El recíproco del período de muestreo se conoce como "frecuencia de muestreo" y se encuentra definido por

$$F_s = \frac{1}{T} \quad (4.6)$$

El muestreo periódico entrega una relación directa entre las variables  $t$  y  $n$ , las cuales se relacionan linealmente a partir del período de muestreo  $T$ , a partir de

$$t = nT = \frac{n}{F_s} \quad (4.7)$$

Debido a lo anterior, existe una relación entre la variable de frecuencia  $F$  de señales analógicas y la frecuencia  $f$  de señales discretas. Es posible demostrar lo anterior considerando la señal analógica

$$x_a(t) = A \cos(2\pi Ft + \theta) \quad (4.8)$$

en la cual, al realizar un muestro periódico a una frecuencia de muestreo de  $F_s = \frac{1}{T}$ , se tiene

$$\begin{aligned} x_a(nT) \equiv x(n) &= A \cos(2\pi F nT + \theta) \\ &= A \cos\left(\frac{2\pi n F}{F_s} + \theta\right) \end{aligned} \quad (4.9)$$

Considerando lo establecido en 4.4, se tiene

$$f = \frac{F}{F_s} \quad (4.10)$$

o de manera equivalente

$$\omega = \Omega T \quad (4.11)$$

De acuerdo a [13], el rango de la variable  $F$  o  $\Omega$  para tiempo continuo es

$$\begin{aligned} -\infty < F < \infty \\ -\infty < \Omega < \infty \end{aligned} \quad (4.12)$$

y para tiempo discreto, corresponde a

$$\begin{aligned} -\frac{1}{2} < f < \frac{1}{2} \\ -\pi < \omega < \pi \end{aligned} \quad (4.13)$$

Sustituyendo 4.10 y 4.11 en 4.13, se tiene que la frecuencia de la señal senoidal en tiempo continuo, cuando se muestrea a una  $F_s = \frac{1}{T}$  se debe encontrar en el rango

$$\begin{aligned} -\frac{1}{2T} = -\frac{F_s}{2} \leq F \leq \frac{F_s}{2} = \frac{1}{2T} \\ -\frac{\pi}{T} = -\pi F_s \leq \Omega \leq \pi F_s = \frac{\pi}{T} \end{aligned} \quad (4.14)$$

De acuerdo a lo anterior, es posible concluir la diferencia fundamental entre las señales en tiempo conitnuo y las señales en tiempo discreto, la cual corresponde con el rango de valores para las variables  $F$  y  $f$ , y  $\Omega$  y  $\omega$ . Debido a que la frecuencia máxima de una señal en tiempo discreto es  $\omega = \pi$  o  $f = \frac{1}{2}$ , los valores máximos que pueden admitir  $F$  y  $\Omega$  para una velocidad de muestreo  $F_s$  son

$$\begin{aligned} F_{max} &= \frac{F_s}{2} = \frac{1}{2T} \\ \Omega_{max} &= \pi F_s = \frac{\pi}{T} \end{aligned} \quad (4.15)$$

### 4.1.4. Cuantificación

El proceso de convertir una señal en tiempo discreto con amplitud continua, en una señal digital es llamado "cuantificación", el cual puede ser descrito por:

$$x_q(n) = Q[x(n)] \tag{4.16}$$

en donde

- $x(n) \rightarrow$  Muestras
- $x_q(n) \rightarrow$  Muestras cuantificadas

Dentro de este proceso, el error que se presenta al representar la señal de valores continuos por medio de un conjunto de valores discretos se llama "error de cuantificación", el cual puede ser definido como la diferencia entre el valor cuantificado y el valor de la muestra original. El error de cuantificación puede ser expresado como:

$$e_q(n) = x_q(n) - x(n) \tag{4.17}$$

Debido a que la cuantificación es en esencia un proceso de aproximación, es posible realizarlo por redondeo o truncamiento.

### 4.1.5. Codificación

En el proceso de cuantificación, se asigna un número binario único a cada distinto nivel de cuantificación. Si se cuenta con  $N$  niveles de cuantificación, se necesitan cuando menos  $N$  niveles o valores binarios diferentes. Si se tiene una longitud de palabra de  $b$  bits, es posible crear  $2^b$  números binarios diferentes, con lo cual se tiene  $2^b \geq N$ .

Las distintas etapas del proceso de conversión analógico-digital (A/D) antes descritas, se pueden observar en la Figura 4.2.

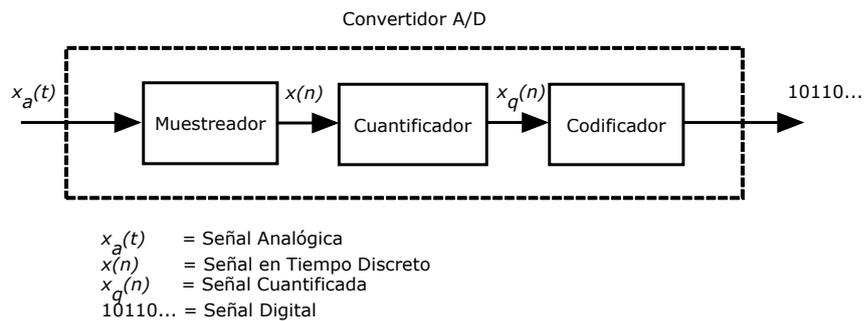


Figura 4.2: Proceso y elementos de la conversión analógico-digital

## 4.2. Sistemas Discretos

De acuerdo con [14], "un sistema discreto es una regla para asignar a una secuencia  $f[n]$  otra secuencia  $g[n]$ . Así, un sistema discreto es un mapeo (transformación) de la secuencia  $f[n]$

a la secuencia  $g[n]$ ". Dentro de la notación anterior,  $n$  representa un índice en tiempo discreto. La secuencia  $f[n]$  corresponde a la entrada y la secuencia  $g[n]$  es la salida o respuesta. Para determinar el valor de la salida  $g[n]$  para algún  $n$  específico, es necesario conocer la entrada  $f[n]$  para cualquier  $n$ , aunque existen casos en los cuales no es necesario.

### 4.3. Controlador *PID* continuo

El controlador *PID* analógico o en tiempo continuo se basa en la manipulación de la variable a controlar por medio de tres acciones de control: proporcional, derivativa e integral. La acción proporcional actúa equitativamente sobre la señal de error, que corresponde con la diferencia entre la señal de entrada y la señal realimentada, la acción integral actúa de manera correspondiente a la integral de la señal de error y por último, la acción de control derivativa actúa de manera proporcional a la derivada de la señal de error ([15], Cap. 3). La acción de control *PID* viene dada por la expresión

$$y(t) = k \left[ e(t) + \frac{1}{T_i} \int_0^t e(t)dt + T_d \frac{de(t)}{dt} \right] \quad (4.18)$$

en donde

- $e(t)$  → Señal de error
- $y(t)$  → Señal de salida del controlador
- $K$  → Ganancia Proporcional
- $T_i$  → Tiempo integral
- $T_d$  → Tiempo derivativo

### 4.4. Sistemas de Control en tiempo Discreto

Los sistemas de control en tiempo discreto son aquellos en los cuales una o más variables pueden presentar cambios en su valor solo en valores discretos de tiempo. El intervalo de tiempo entre dichos instantes debe ser lo suficientemente corto para tener la mejor aproximación de la señal discreta a la señal continua ([15], Cap. 1).

### 4.5. Controlador *PID* discreto

Una vez descrito el controlador *PID* analógico, es posible obtener la expresión del mismo en tiempo discreto por medio del uso de la transformada Z y su operación inversa. Desarrollando la ecuación 4.18, se tiene

$$y(t) = ke(t) + \frac{k}{T_i} \int_0^t e(t)dt + kT_d \frac{de(t)}{dt} \quad (4.19)$$

Aplicando la transformada Z y factorizando los términos comunes  $k$  y  $E(z)$

$$Y(z) = k \left[ 1 - \frac{T}{2T_i} + \frac{T}{T_i} \frac{1}{1 - z^{-1}} + \frac{T_d}{T} (1 - z^{-1}) \right] E(z) \quad (4.20)$$

Agrupando términos

$$Y(z) = k \left[ 1 - \frac{T}{2T_i} \right] E(z) + \left[ \frac{kT}{T_i} \frac{1}{1 - z^{-1}} \right] E(z) + \left[ \frac{kT_d}{T} (1 - z^{-1}) \right] E(z) \quad (4.21)$$

$$Y(z) = \frac{\left[ k \left[ 1 - \frac{T}{2T_i} \right] [1 - z^{-1}] \right] E(z) + \left[ \frac{kT}{T_i} \right] E(z) + \left[ \frac{kT_d}{T} \right] [1 - z^{-1}] [1 - z^{-1}] E(z)}{(1 - z^{-1})} \quad (4.22)$$

$$(1 - z^{-1})Y(z) = \left[ k \left[ 1 - \frac{T}{2T_i} \right] [1 - z^{-1}] \right] E(z) + \left[ \frac{kT}{T_i} \right] E(z) + \dots \quad (4.23)$$

$$\dots + \left[ \frac{kT_d}{T} \right] [1 - z^{-1} + z^{-2}] E(z)$$

Considerando que las ganancias proporcional, derivativa e integral vienen dadas por

$$k_p = k$$

$$k_d = kT_d$$

$$k_i = \frac{k}{T_i}$$

Se tiene

$$Y(z) - Y(z)z^{-1} = \left[ \left[ k_p - \frac{k_i T}{2} \right] [1 - z^{-1}] \right] E(z) + [k_i T] E(z) + \dots \quad (4.24)$$

$$\dots + \left[ \frac{k_d}{T} \right] [1 - z^{-1} + z^{-2}] E(z)$$

Aplicando la transformada Z inversa a la ecuación 4.24, se tiene

$$y(k) - y(k-1) = \left[ \left[ k_p - \frac{k_i T}{2} \right] \right] [e(k) - e(k-1)] + [k_i T] e(k) + \dots \quad (4.25)$$

$$\dots + \left[ \frac{k_d}{T} \right] [e(k) - 2e(k-1) + e(k-2)]$$

Agrupando y reduciendo términos

$$y(k) = y(k-1) + \left[ k_p + \frac{k_i T}{2} + \frac{k_d}{T} \right] [e(k)] + \left[ -k_p + \frac{k_i T}{2} - 2\frac{k_d}{T} \right] [e(k-1)] + \dots \quad (4.26)$$

$$\dots + \left[ \frac{k_d}{T} \right] [e(k-2)]$$

En donde se puede observar que la acción de control viene dada por el valor del error en estado estable actual, el valor del mismo en uno y dos instantes pasados y por la acción de control en un instante pasado.

## 4.6. Control de Traslación: Linealización exacta

Con el fin de obtener el valor de los ángulos deseados para trasladar el cuadri-rotor de un cierto punto a otro, es decir  $\theta_d$  y  $\phi_d$  se realizó una linealización exacta sobre el modelo del mismo. Considerando la representación en el espacio de estado del modelo simplificado obtenido anteriormente, los subsistemas relacionados con la posición y la orientación pueden ser reescritos como

$$\begin{aligned}\dot{x}_2 &= -\frac{1}{m} \sin \theta_1 u_1 \\ \dot{y}_2 &= \frac{1}{m} \cos \theta_1 \sin \phi_1 u_1 \\ \dot{z}_2 &= \frac{1}{m} \cos \theta_1 \cos \phi_1 u_1 - g\end{aligned}\quad (4.27)$$

$$\begin{aligned}\dot{\psi}_2 &= \tau_\phi \\ \dot{\theta}_2 &= \tau_\theta \\ \dot{\psi}_2 &= \tau_\psi\end{aligned}\quad (4.28)$$

Se parte del sistema de posición para obtener los ángulos deseados, en donde se traduce el mismo a un sistema de errores

$$\begin{aligned}e_{x1} &= x_1 - x_{1r} \\ e_{y1} &= y_1 - y_{1r} \\ e_{z1} &= z_1 - z_{1r}\end{aligned}\quad (4.29)$$

en donde  $x_1$  representa la posición actual en el eje coordenado  $x$  y  $x_{1r}$  representa la posición deseada dentro del mismo eje. La misma metodología se emplea para el resto de los ejes coordenados. Derivando dos veces la ecuación 4.29, se tiene

$$\begin{aligned}\dot{e}_{x1} &= e_{x2} = \dot{x}_1 - \dot{x}_{1r} \\ \dot{e}_{y1} &= e_{y2} = \dot{y}_1 - \dot{y}_{1r} \\ \dot{e}_{z1} &= e_{z2} = \dot{z}_1 - \dot{z}_{1r}\end{aligned}\quad (4.30)$$

$$\begin{aligned}\dot{e}_{x2} &= -\frac{1}{m} \sin \theta_1 u_1 - \delta_x \\ \dot{e}_{y2} &= \frac{1}{m} \cos \theta_1 \sin \phi_1 u_1 - \delta_y \\ \dot{e}_{z2} &= \frac{1}{m} \cos \theta_1 \cos \phi_1 u_1 - \delta_z\end{aligned}\quad (4.31)$$

en donde

- $\delta_x = \ddot{x}_{1r}$

#### 4.6. CONTROL DE TRASLACIÓN: LINEALIZACIÓN EXACTA

---

- $\delta_y = \ddot{y}_{1r}$
- $\delta_z = g + \ddot{z}_{1r}$

Como en [16], se parte del subsistema de altura, del cual se obtiene un control sintético empleado para eliminar las no linealidades del modelo. Dicho control sintético se encuentra dado por

$$u_1 = \frac{m}{\cos \theta_{1r} \cos \phi_{1r}} (V_z + \delta_z) \quad (4.32)$$

en donde  $V_z = -k_{pz}e_{z1} - k_{dz}e_{z2} - k_{iz} \int_0^t e_{z1}(t)dt$ . La expresión anterior corresponde con un controlador *PID*, el cual se encuentra considerado para usos posteriores para  $V_x$  y  $V_y$ , los cuales son los controladores de posición para  $x$  y  $y$  respectivamente.  $k_{pn}$ ,  $k_{dn}$  and  $k_{in}$  son constantes positivas para el controlador *PID* con  $n = \{x, y, z\}$ .

Considerando que  $\theta_1 \rightarrow \theta_{1r}$  y  $\phi_1 \rightarrow \phi_{1r}$  conforme  $t \rightarrow \infty$  y sustituyendo el control sintético  $u_1$  en la ecuación para el subsistema  $y$ , se tiene

$$\begin{aligned} \dot{e}_{y2} &= \cancel{\cos \theta_1} \sin \phi_1 \frac{1}{\cancel{m}} \left[ \frac{m}{\cancel{\cos \theta_{1r}} \cos \phi_{1r}} (V_z + \delta_z) \right] - \delta_y \\ \dot{e}_{y2} &= \tan \phi_{1r} (V_z + \delta_z) - \delta_y \end{aligned} \quad (4.33)$$

en donde se busca que  $\dot{e}_{y2} = V_y$ , por lo cual se debe calcular el valor deseado de  $\phi_1$ , es decir,  $\phi_{1r}$ , teniendo la expresión

$$\phi_{1r} = \tan^{-1} \left[ \frac{V_y + \delta_y}{V_z + \delta_z} \right] \quad (4.34)$$

Del mismo modo, considerando el control sintético obtenido anteriormente y sustituyéndolo en el subsistema para  $x$  se tiene

$$\begin{aligned} \dot{e}_{x2} &= -\sin \theta_1 \frac{1}{m} \left[ \frac{m}{\cos \theta_{1r} \cos \phi_{1r}} (V_z + \delta_z) \right] - \delta_x \\ \dot{e}_{x2} &= -\frac{\tan \theta_{1r}}{\cos \phi_{1r}} (V_z + \delta_z) - \delta_x \end{aligned} \quad (4.35)$$

en donde se busca que  $\dot{e}_{x2} = V_x$ , por lo cual se debe calcular el valor deseado de  $\theta_1$ , es decir,  $\theta_{1r}$ , con lo que se tiene

$$\theta_{1r} = \tan^{-1} \left[ \cos \phi_{1r} \frac{V_x + \delta_x}{V_z + \delta_z} \right] \quad (4.36)$$

De esta manera, se obtienen las expresiones necesarias para calcular los ángulos deseados  $\phi$  y  $\theta$  para realizar una traslación del cuadri-rotor a un cierto punto en el plano  $X$ - $Y$ .

Dado que el error en estado estable se encuentra dado por  $e = PV - SP$ , en donde  $PV$  es la variable del proceso o el estado a observar y  $SP$  corresponde a la referencia o valor deseado de la variable del proceso, se busca que  $x \rightarrow x_d$ ,  $y \rightarrow y_d$  y  $z \rightarrow z_d$ , el controlador implementado busca hacer que el error de estado estable  $e \rightarrow 0$ , lo cual se logra cuando solo se encuentra presente la acción de control, como se puede observar a continuación

$$\begin{aligned}\dot{e}_{x2} &= V_x = -kp_x e_{x1} - kd_x e_{x2} - ki_x \int_0^t e_{x1}(t) dt \\ \dot{e}_{y2} &= V_y = -kp_y e_{y1} - kd_y e_{y2} - ki_y \int_0^t e_{y1}(t) dt \\ \dot{e}_{z2} &= V_z = -kp_z e_{z1} - kd_z e_{z2} - ki_z \int_0^t e_{z1}(t) dt\end{aligned}\quad (4.37)$$

## 4.7. Análisis de estabilidad

De acuerdo a [10], se considera la dinámica del sistema dada por

$$\mathbb{I}(\eta)\ddot{\tilde{\eta}} + \mathcal{C}(\eta, \dot{\tilde{\eta}})\dot{\tilde{\eta}} = \tau \quad (4.38)$$

Se propone la función candidata de Lyapunov

$$V(\tilde{\eta}, \dot{\tilde{\eta}}_p) = \frac{1}{2}\dot{\tilde{\eta}}^T \mathbb{I}(\eta)\dot{\tilde{\eta}} + \sum_{i=1}^n k_{a_i} \ln(\cosh(k_{p_i} \tilde{\eta}_i)) \quad (4.39)$$

en donde  $\tilde{\eta}_i$  es el  $i$ -ésimo elemento del vector  $\tilde{\eta} = \eta - \eta_d$ ,  $k_{a_i}$  y  $k_{p_i}$  son los  $i$ -ésimos elementos de la diagonal de las matrices positivas  $k_a \in \mathbb{R}^{3 \times 3}$  y  $k_p \in \mathbb{R}^{3 \times 3}$ , respectivamente. La función candidata de Lyapunov  $V(\eta, \dot{\eta})$  es definida positiva debido a que  $\mathbb{I} > 0$  y  $\ln(\cosh(\cdot))$  es una función positiva radialmente no acotada. Derivando 4.39, se tiene

$$\dot{V}(\tilde{\eta}, \dot{\tilde{\eta}}) = \frac{1}{2}\dot{\tilde{\eta}}^T \mathbb{I}(\eta)\dot{\tilde{\eta}} + \dot{\tilde{\eta}}^T \mathbb{I}(\eta)\ddot{\tilde{\eta}} + \sum_{i=1}^n \dot{\tilde{\eta}}_i k_{a_i} k_{p_i} \tanh(k_{p_i} \tilde{\eta}_i) \quad (4.40)$$

que puede ser reescrita como

$$\dot{V}(\tilde{\eta}, \dot{\tilde{\eta}}) = \frac{1}{2}\dot{\tilde{\eta}}^T \mathbb{I}(\eta)\dot{\tilde{\eta}} + \dot{\tilde{\eta}}^T \mathbb{I}(\eta)\ddot{\tilde{\eta}} + \dot{\tilde{\eta}}^T k_a k_p \text{Tanh}(k_p \tilde{\eta}) \quad (4.41)$$

en donde  $\text{Tanh}$  es el vector que contiene los ángulos de Euler saturados y que es igual a

$$\text{Tanh}(k_p \tilde{\eta}) = (\tanh(k_{p1} \tilde{\phi}), \tanh(k_{p2} \tilde{\theta}), \tanh(k_{p3} \tilde{\psi}))^T \quad (4.42)$$

Sustituyendo 4.38 en 4.41, se tiene

$$\dot{V}(\tilde{\eta}, \dot{\tilde{\eta}}) = \dot{\tilde{\eta}}^T \left( \frac{1}{2}\dot{\mathbb{I}}(\eta) - \mathcal{C}(\eta, \dot{\tilde{\eta}}) \right) \dot{\tilde{\eta}} + \dot{\tilde{\eta}}^T (\tilde{\tau} + k_a k_p \text{Tanh}(k_p \tilde{\eta})) \quad (4.43)$$

Considerando que  $\dot{\mathbb{I}}(\eta) - 2\mathcal{C}(\eta, \dot{\tilde{\eta}})$  es una matriz antisimétrica, se tiene

$$\dot{V}(\tilde{\eta}, \dot{\tilde{\eta}}) = \dot{\tilde{\eta}}^T (\tilde{\tau} + k_{s_p} \text{Tanh}(k_p \tilde{\eta})) \quad (4.44)$$


---

## 4.7. ANÁLISIS DE ESTABILIDAD

---

Se introduce el control  $PD$  saturado

$$\tilde{\tau} = -k_{sp} \text{Tanh}(k_p \tilde{\eta}) - k_{sv} \text{Tanh}(k_v \dot{\tilde{\eta}}) \quad (4.45)$$

en donde  $k_{sp} = k_a k_p$  y  $k_{sv} = k_b k_v$  son matrices diagonales positivas. Sustituyendo la ley de control en 4.44, se tiene

$$\dot{V}(\tilde{\eta}, \dot{\tilde{\eta}}) = -\dot{\tilde{\eta}}^T k_{sv} \text{Tanh}(k_v \dot{\tilde{\eta}}) \quad (4.46)$$

En 4.46 se puede observar que  $\dot{V} \leq 0$ , por lo cual el origen es estable y las soluciones  $\eta t$  y  $\dot{\eta}(t)$  son acotadas.



---

# Sistema de Visión Artificial

---

Los sistemas de visión artificial constituyen una parte fundamental de la mayoría de los UAV debido a que aumentan la capacidad del sistema de realizar tareas autónomas u operadas a distancia de una manera mucho más eficiente. En comparación con otros sistemas de navegación y de otro tipo de sensores, los sistemas de visión artificial cuentan con ciertas características únicas ([24], Cap. 11), como son:

- Proveen amplia información sobre ciertos objetos de interés y sobre el entorno que se encuentra dentro de su campo de visión
- Solo requieren de luz natural para operar
- Son de bajo costo y generalmente son muy ligeros en comparación a otros sistemas de detección

## 5.1. Modelo de Cámara

### 5.1.1. Representación en 2-D

Una manera de aproximar la manera en que se forma una imagen con respecto al mundo tridimensional se realiza por medio de transformaciones de perspectiva, por medio de la cual se proyectan puntos sobre un plano [17]. El proceso de formación de una imagen se puede observar en la Figura 5.1, en donde se tiene el sistema de coordenadas de una cámara, denotado por  $(X_c, Y_c, Z_c)$ , y el sistema de coordenadas de la imagen, el cual corresponde con  $(x_i, y_i, z_i)$ . Debido a que el plano imagen es paralelo al plano  $X_c, Z_c$ , el eje óptico coincide con el eje  $y_i$ . De esta forma, el centro del plano imagen se encuentra en el origen y el centro de la lente de la cámara se encuentra en las coordenadas  $(0, -f, 0)$ , en donde  $f$  corresponde a la distancia focal de la lente.

Considerando cualquier punto en el espacio tridimensional con coordenadas globales  $(x_0, y_0, z_0)$ , es necesario obtener una relación que proporcione las coordenadas  $(x_i, y_i)$  de la proyección de dicho punto sobre el plano imagen [17] [18].

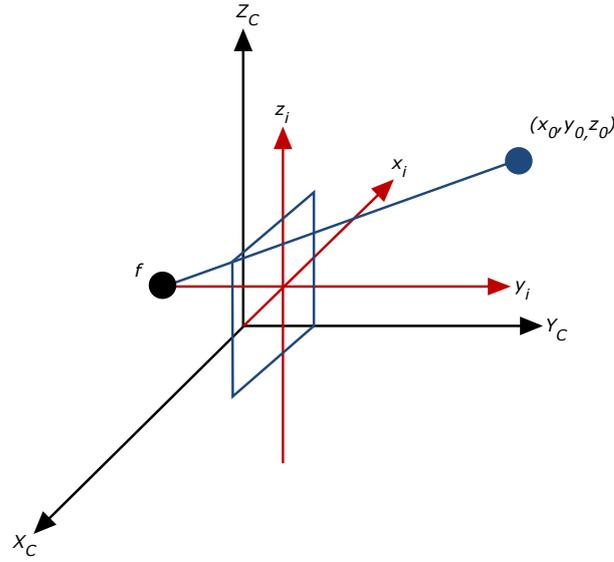


Figura 5.1: Proyección de imagen del espacio tridimensional al plano imagen

### 5.1.2. Representación en 3-D

Considerando que el prototipo es un móvil, sobre el cual se monta una plataforma en la que se encuentra posicionada la cámara, es necesario obtener un sistema de posicionamiento de los puntos en el espacio tridimensional sobre el plano imagen. El sistema de posicionamiento antes mencionado se puede observar en la Figura 5.2, en donde  $R_e$  representa el sistema coordenado del marco del mundo,  $R_p$  representa el sistema de la plataforma. Las coordenadas  $x_r$  y  $y_r$  son obtenidas a partir del sistema de localización implementado dentro del prototipo, con lo cual es posible generar la posición del móvil dentro del sistema coordenado del mundo real.

Debido a que la rotación de la cámara se realiza con respecto a los ejes de la plataforma  $(X_p, Y_p, Z_p)$ , la posición de la misma en el marco  $R_p$  se encuentra dada por  $(e_x, e_y, e_z)$ . Se realiza la suposición de que el plano imagen  $(X_c, Y_c)$  es paralelo al plano plataforma  $(X_p, Y_p)$ .

Utilizando las matrices de mapeo de un marco hacia otro, es posible posicionar cualquier punto de un objeto  $M(x_0, y_0, z_0)$  sobre el plano imagen. Las coordenadas del punto de la imagen  $(x_i, z_i)$  se encuentran representadas por:

$$x_i = f \frac{x_0 \cos(\Theta) - y_0 \sin(\Theta) + A}{x_0 \sin(\Theta) \cos(\Phi) + y_0 \cos(\Theta) \cos(\Phi) - z_0 \sin(\Phi) + B + f} \quad (5.1)$$

$$z_i = f \frac{x_0 \sin(\Theta) \sin(\Phi) + y_0 \cos(\Theta) \sin(\Phi) + z_0 \cos(\Phi) + C}{x_0 \sin(\Theta) \cos(\Phi) + y_0 \cos(\Theta) \cos(\Phi) - z_0 \sin(\Theta) + B + f} \quad (5.2)$$

en donde

- $\Theta \rightarrow$  rotación con respecto al eje  $Z_p$
- $\Phi \rightarrow$  rotación con respecto al eje  $X_p$
- $A = -x_r \cos(\Theta) + y_r \sin(\Theta) - e_x$

## 5.2. DETECCIÓN DE PUNTOS DE INTERÉS Y EMPAREJAMIENTO

---

- $B = -x_r \sin(\Theta) \cos(\Phi) - y_r \cos(\Theta) \cos(\Phi) + h \sin(\Phi) - e_y$
- $C = x_r \sin(\Theta) \sin(\Phi) - y_r \cos(\Theta) \sin(\Phi) - h \cos(\Phi) - e_z$

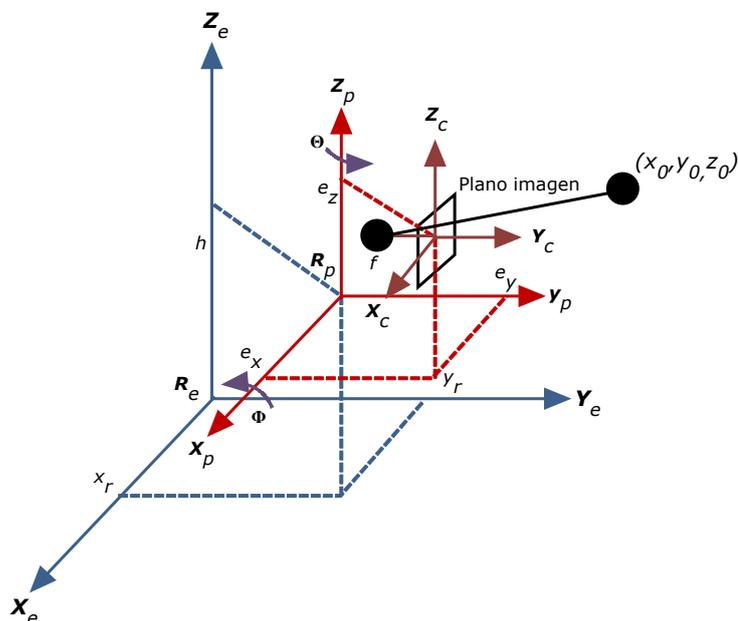


Figura 5.2: Sistema coordenado 3D

## 5.2. Detección de Puntos de Interés y Emparejamiento

La detección de ciertas características y su emparejamiento consecutivo entre dos o más imágenes y por ende, en un video, es un punto clave dentro de diversos problemas de visión artificial, como pueden ser la detección de escenas y objetos, correspondencia estereoscópica, seguimiento de un objetivo, entre otros [19]. Estas características corresponden con ciertas zonas específicas de la imagen y son comúnmente llamados "puntos de interés" y son generalmente descritos por la apariencia e información de un parche o grupo de píxeles que rodean la localización del punto de interés [7]. A los mecanismos utilizados para localizar puntos de interés dentro de una imagen se les denomina comúnmente "detectores".

Un detector de puntos de interés ideal es capaz de localizar las regiones más sobresalientes dentro de una imagen, las cuales son capaces de ser detectadas sin problemas dentro de imágenes sucesivas, siendo de esta manera robusto ante todas las transformaciones posibles que pueda sufrir la imagen [20].

Existen principalmente dos métodos para localizar puntos característicos dentro de una imagen y su correspondencia en imágenes consecutivas: el primero consiste en utilizar una técnica de rastreo local como pueden ser mínimos cuadrados o la operación de correlación, por otro lado, el segundo método consiste en parear los puntos característicos

localizados, basándose en su apariencia local. Se pueden considerar 3 etapas dentro del proceso de detección y emparejamiento de puntos característicos, teniendo a su vez una cuarta etapa opcional, con lo cual, se tienen las siguientes etapas [7]:

- Detección de características
- Descripción de características
- Emparejamiento de características
- Seguimiento de características

## 5.3. Escala y Pirámides Gaussianas

### 5.3.1. Pirámides Gaussianas

Una pirámide gaussiana consiste en una colección de representaciones de una imagen, en donde típicamente cada capa de la pirámide es de la mitad de ancho y la mitad de alto que la capa anterior, con lo cual se podría construir una pirámide si se empalmara una capa sobre otra y así sucesivamente. Dentro de una pirámide gaussiana, cada capa es suavizada por medio de un kernel gaussiano simétrico y remuestreada para obtener la capa siguiente. La imagen más pequeña corresponde con la más suavizada y generalmente las capas son referidas como "escalas aproximadas" de la imagen original.

### 5.3.2. Espacio escala

Tipicamente, se tienen puntos dentro de una imagen, los cuales se desea encontrar correctamente en imágenes sucesivas; una manera eficiente y comúnmente utilizada para realizar este emparejamiento consiste en buscar una coincidencia en copias de la imagen original, las cuales se encuentran altamente suavizadas y remuestreadas, para posteriormente buscar dicha coincidencia en una versión más detallada de dicha imagen, y así sucesivamente [21].

Al trabajar sobre escalas mucho menores (o gruesas, por así llamarlas) dentro de una imagen, se tiene información relevante sobre la señal obtenida a nivel de pixel, por otro lado, al trabajar sobre escalas finas, es posible obtener información detallada sobre la imagen en general. Para una señal en 1D, es posible representar el comportamiento de ciertas características a medida que los parámetros de la escala van cambiando, como se puede observar en la Figura 5.3. Estos bosquejos y sus respectivas representaciones representaciones son comúnmente conocidas como *espacio escala*.

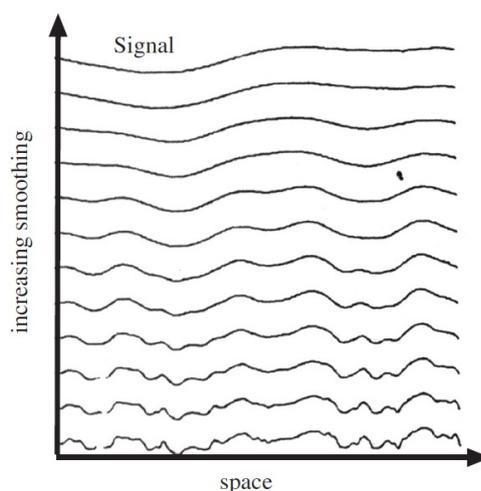


Figura 5.3: Comportamiento de las señales dentro de una imagen a medida que aumenta la escala

Si se plantea la señal como un conjunto de parches paramétricos que se encuentran alrededor de los puntos característicos, se puede obtener una descomposición distinta de la señal a distintas escalas. Es posible trasladar el concepto de espacio escala al caso en 2D, como es la implementación usual.

### 5.4. Descriptores Binarios

Un descriptor es un método para codificar información acerca de un parche o grupo de píxeles dentro de una imagen, el cual se encuentra alrededor de un punto característico dentro de la misma. Un descriptor ideal es capaz de capturar la información más importante contenida en las regiones más sobresalientes dentro de una imagen, de manera que la misma estructura pueda ser reconocida si se es encontrada en imágenes sucesivas [20]. El uso de descriptores binarios se ha popularizado debido a que son capaces de describir los parches alrededor de un punto característico utilizando cadenas de bits.

En esencia, un descriptor binario escoge un patrón dentro de los parches en la imagen y realiza comparaciones por parejas entre las intensidades medias a lo largo de parejas de elementos predeterminados en el patrón; a partir de estas comparaciones se genera una cadena de bits para cada parche.

### 5.5. Reconocimiento de patrones

El reconocimiento de patrones consiste en "el descubrimiento automático de regularidades dentro de datos a través del uso de algoritmos de computadora y al uso de dichas regularidades para realizar acciones tales como clasificar los datos en diferentes categorías" ([22], Cap. 1).

El emplear técnicas heurísticas para reconocer ciertos patrones dentro de un conjunto de datos resulta en un conjunto de reglas y excepciones a las mismas, que en consecuencia generan otro conjunto de reglas y así sucesivamente, lo cual presenta resultados pobres para dicho procedimiento. Un acercamiento que entrega mucho mejores resultados consiste en adoptar un enfoque basado en una máquina de aprendizaje automatizado con un amplio conjunto de  $N$  datos  $(\mathbf{x}_1, \dots, \mathbf{x}_N)$  llamado "Conjunto de Entrenamiento", el cual sirve para ajustar los parámetros de un modelo adaptativo. Las categorías de cada uno de los datos dentro del conjunto anterior se conocen *a priori*, etiquetando a cada una de manera manual.

Es posible expresar la categoría de cada dato entrante a partir de un "Vector de Objetivo"  $\mathbf{t}$ , el cual representa la identidad de dicho dato con respecto a los datos almacenados dentro del conjunto de entrenamiento.

El resultado obtenido de una máquina de aprendizaje automatizado puede ser expresado como una función  $y(\mathbf{x})$ , la cual adopta un vector de entrada  $\mathbf{x}$  y genera un vector de salida  $\mathbf{y}$ , el cual se encuentra codificado de la misma manera que los vectores de objetivo correspondientes. La forma precisa de dicha función  $y(\mathbf{x})$  se define durante la "etapa de aprendizaje" con base en los datos de entrenamiento proporcionados. Una vez que el modelo se encuentra entrenado, es posible determinar la identidad de los nuevos datos entrantes con respecto a los datos de entrenamiento.

A partir de lo anterior, se conoce como "generalización" a la capacidad de clasificar de manera correcta a los nuevos datos entrantes que sean distintos a los usados para realizar el entrenamiento. Durante la implementación, es una práctica común el pre procesar los datos de entrada para transformarlos a un espacio de trabajo por medio del cual se espera que el problema de reconocimiento de patrones se simplifique en gran manera. A la etapa antes descrita se le conoce como "detección de características" ([22], Cap. 1).

## 5.6. El problema de clasificación

### 5.6.1. Modelos lineales para clasificación

El problema de clasificación consiste en analizar un vector de entrada  $\mathbf{x}$  y clasificar al mismo en alguna de las  $C_k$  clases discretas posibles dentro del problema, con  $k = 1, \dots, N$ . Para realizar lo anterior, el espacio de entrada es dividido en "regiones de decisión", cuyas fronteras son llamadas "fronteras de decisión". Al emplear modelos lineales para realizar la clasificación de los elementos antes mencionados, las fronteras de decisión son funciones lineales del vector de entrada  $\mathbf{x}$ , motivo por el cual son definidas por hiperplanos de dimensión  $(D - 1)$  contenidos dentro del espacio de entrada de dimensión  $D$ . Los conjuntos de datos cuyas clases pueden ser separadas de una manera exacta por fronteras de decisión lineales se dice que son linealmente separables ([22], Cap. 4).

### 5.6.2. Máquina de Soporte Vectorial (SVM)

El problema de clasificación para dos clases distintas puede ser visto a partir del modelo lineal de la forma

$$y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x} + b) \quad (5.3)$$

en donde  $\phi(\mathbf{x})$  denota una transformación con una característica espacial fija y el parámetro de sesgo  $b$  se define de manera explícita. Para este caso, el conjunto de entrenamiento consiste en  $N$  vectores de entrada  $(\mathbf{x}_1, \dots, \mathbf{x}_N)$  con valores objetivo  $t_1, \dots, t_N \in \{-1, 1\}$ , y donde los nuevos datos de entrada  $\mathbf{x}$  se encuentran clasificados de acuerdo al signo de  $y(\mathbf{x})$ .

Se puede asumir que el conjunto de datos para el entrenamiento del clasificador son linealmente separables dentro del espacio de trabajo, por lo cual se tiene un valor de  $\mathbf{w}$  y  $b$ , tal que la ecuación (5.1) cumple que  $y(\mathbf{x}_n) > 0$  para los puntos cuyo objetivo sea  $t_n = +1$  y  $y(\mathbf{x}_n) < 0$  para los puntos en donde  $t_n = -1$ , de manera que  $t_n y(\mathbf{x}_n) > 0$  para todos los datos de entrenamiento.

Debido a que existen múltiples soluciones que son capaces de separar las clases de manera exacta, el problema recae en encontrar la adecuada que entregue el menor error de generalización. Una SVM resuelve esta problemática por medio del concepto de "margen", el cual se entiende como la mínima distancia posible entre la frontera de decisión y cualquiera de las muestras, como se puede observar en la Figura 5.4. Dentro de las SVM, se busca que la frontera de decisión se localice de tal manera que el margen sea el máximo posible.

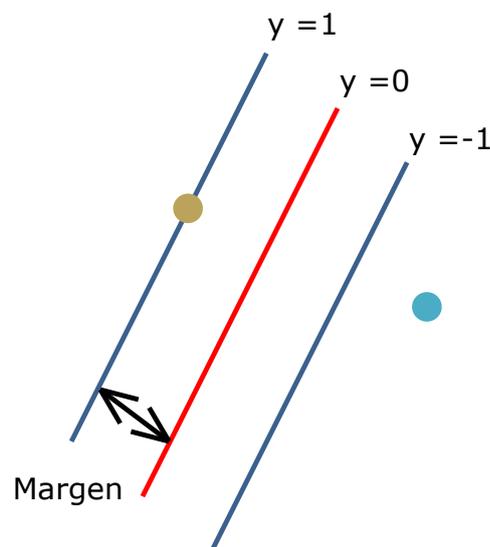


Figura 5.4: Detección del punto de interés en el espacio escalar

Considerando que la distancia de un punto  $\mathbf{x}$  a un hiperplano definido por  $y(\mathbf{x}) = 0$  en donde  $y(\mathbf{x})$  adopta la forma declarada en la ecuación (5.1) viene dada por  $|y(\mathbf{x})|/||\mathbf{w}||$  y que se busca

una solución en la cual todos los datos se clasifican correctamente, teniendo que  $t_n y(\mathbf{x}_n) > 0 \forall n$ , la distancia desde un punto  $\mathbf{x}_n$  a la frontera de decisión viene dada por

$$\frac{t_n y(\mathbf{x}_n)}{\|\mathbf{w}\|} = \frac{t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b)}{\|\mathbf{w}\|} \quad (5.4)$$

en donde el valor del margen se obtiene a partir de la distancia perpendicular hacia el punto  $\mathbf{x}_n$  más cercano. Como se mencionó anteriormente, se busca maximizar el valor dicho margen, por lo cual es necesario optimizar los valores de los parámetros  $\mathbf{w}$  y  $b$ , Dicha optimización se obtiene a partir de

$$\arg \max_{\mathbf{w}, b} \left\{ \frac{1}{\|\mathbf{w}\|} \min_n [t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b)] \right\} \quad (5.5)$$

Para simplificar la solución al problema de optimización, se realiza un reescalamiento de  $\mathbf{w} \rightarrow k\mathbf{w}$  y  $b \rightarrow kb$ , con lo cual la distancia desde cualquier punto  $\mathbf{x}_n$  hacia la frontera de decisión permanece sin cambios. Partiendo de lo anterior, se puede declarar

$$t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + 1) = 1 \quad (5.6)$$

para el punto que se encuentre más cercano a la frontera de decisión, con lo cual todos los puntos satisfacen las restricciones

$$t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geq 1, \quad n = 1, \dots, N \quad (5.7)$$

De la ecuación 5.5, se busca maximizar  $\|\mathbf{w}\|^{-1}$ , lo cual se considera equivalente a minimizar  $\|\mathbf{w}\|^2$ , con lo que se tiene el problema de optimización

$$\arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 \quad (5.8)$$

el cual se encuentra sujeto a las restricciones establecidas en la ecuación ???. Para resolver el problema de optimización con restricciones planteado, es necesario hacer uso de multiplicadores de Lagrange  $a_n \geq 0$ , en donde se tiene un multiplicador para cada una de las restricciones establecidas previamente, con lo cual se obtiene la función Lagrangiana

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{n=1}^N a_n \{t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) - 1\} \quad (5.9)$$

en donde  $\mathbf{a} = (a_1, \dots, a_N)^T$  corresponde con el vector columna de multiplicadores de Lagrange. Diferenciando parcialmente el Lagrangiano con respecto a  $\mathbf{w}$  y  $b$ , se tiene las condiciones

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{n=1}^N a_n t_n \phi(\mathbf{x}_n) = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{n=1}^N a_n t_n \phi(\mathbf{x}_n) \quad (5.10)$$

## 5.6. EL PROBLEMA DE CLASIFICACIÓN

---

$$\frac{\partial L}{\partial b} = -\sum_{n=1}^N a_n t_n = 0 \Rightarrow \sum_{n=1}^N a_n t_n = 0 \quad (5.11)$$

Considerando las condiciones determinadas en las ecuaciones 5.10 y (5.11) dentro del Lagrangiano de la ecuación (5.9), se tiene

$$\begin{aligned} L(\mathbf{w}, b, \mathbf{a}) &= \frac{1}{2} \sum_{n=1}^N a_n t_n \phi(\mathbf{x}_n) \sum_{m=1}^N a_m t_m \phi(\mathbf{x}_m) - \dots \\ &\dots - \sum_{n=1}^N a_n t_n \phi(\mathbf{x}_n) \sum_{m=1}^N a_m t_m \phi(\mathbf{x}_m) - \underbrace{\sum_{n=1}^N a_n t_n b}_{0} + \sum_{n=1}^N a_n \end{aligned} \quad (5.12)$$

Al eliminar  $\mathbf{w}$  y  $b$  del Lagrangiano  $L(\mathbf{w}, b, \mathbf{a})$  bajo las condiciones antes descritas en las ecuaciones 5.10 y ??, reduciendo términos semejantes y reacomodando la ecuación ??, se tiene la representación dual siguiente

$$\tilde{L}(\mathbf{a}) = \sum_{n=1}^N a_n - \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N a_n a_m t_n t_m \phi(\mathbf{x}_n) \phi(\mathbf{x}_m) \quad (5.13)$$

en donde se maximiza con respecto al vector de multiplicadores de Lagrange  $\mathbf{a}$ , sujeto a las restricciones

$$a_n \geq 0, \quad n = 1, \dots, N \quad (5.14)$$

$$\sum_{n=1}^N a_n t_n = 0 \quad (5.15)$$

La función kernel se encuentra definida como  $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}')$ . Al realizar la formulación doble, el problema de optimización original con  $M$  variables mostrado en la ecuación 5.8 se convierte en uno dual con  $N$  variables, como se define en la ecuación 5.13, por medio del cual es posible reformular el modelo utilizando kernels, permitiendo de esta manera que el máximo margen para la clasificación pueda ser aplicado a espacios característicos cuyas dimensiones excedan el número de datos.

Para poder clasificar nuevos datos utilizando el modelo previamente entrenado, es necesario evaluar el signo de  $y(\mathbf{x})$  de acuerdo a la ecuación 5.3, cuya expresión puede definirse a partir de los parámetros  $a_n$  y la función kernel antes descrita, sustituyendo el valor de  $\mathbf{w}$  obtenido en la ecuación 5.10. La expresión antes mencionada corresponde con

$$y(\mathbf{x}) = \sum_{n=1}^N a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b \quad (5.16)$$

Una optimización como la antes descrita satisface las condiciones de *Karush-Kuhn-Tucker*, en donde para este caso, se requiere que se mantengan las siguientes propiedades

$$a_n \geq 0 \quad (5.17)$$

$$t_y(\mathbf{x}_n) - 1 \geq 0 \quad (5.18)$$

$$a_n \{t_n y(\mathbf{x}_n) - 1\} = 0 \quad (5.19)$$

con lo que se tiene que para cada dato,  $a_n = 0$  o  $t_n y(\mathbf{x}_n) = 1$ . Para el caso en el cual para algún dato  $a_n = 0$ , dicho dato no se presentaría en la ecuación 5.16. Los datos restantes son llamados "vectores de soporte" y dado que cumplen que  $t_n y(\mathbf{x}_n) = 1$ , los mismos corresponden con puntos que se localizan los hiperplanos de margen máximos dentro del espacio, como se mostró en la Figura 5.4. Debido a lo anterior, una vez que el modelo se encuentra entrenado, una gran cantidad de los datos utilizados pueden ser descartados y solo los vectores de soporte se mantienen.

Para calcular el valor de  $b$ , se utiliza la propiedad de que cualquier vector de soporte  $\mathbf{x}_n$  cumple que  $t_n y(\mathbf{x}_n) = 1$ . Tomando esta premisa y sustituyendo la ecuación 5.16 en la misma, se tiene

$$t_n \left( \sum_{m \in \mathcal{S}} a_m t_m k(\mathbf{x}_n, \mathbf{x}_m) + b \right) = 1 \quad (5.20)$$

en donde  $\mathcal{S}$  denota al conjunto de índices de los vectores de soporte. La solución para  $b$  parte de multiplicar en un principio la ecuación 5.20 por el factor  $t_n$ , posteriormente utilizar el hecho de que  $t_n^2 = 1$  y finalmente promediando estas ecuaciones a lo largo de todos los vectores de soporte, teniendo como resultado para  $b$

$$b = \frac{1}{N_{\mathcal{S}}} \sum_{n \in \mathcal{S}} \left( t_n - \sum_{m \in \mathcal{S}} a_m t_n k(\mathbf{x}_n, \mathbf{x}_m) \right) \quad (5.21)$$

en donde  $N_{\mathcal{S}}$  corresponde al número de vectores de soporte.

Debido a la posibilidad de que se presente el caso en el cual los datos a clasificar no son linealmente separables, se considera el trasladar el espacio de trabajo hacia una dimensión superior, con lo que se pretende facilitar la tarea de clasificación.

## 5.7. Algoritmo de detección de personas

El detectar a una persona dentro de una escena constituye una tarea complicada debido a las diversas apariencias que poseen y las diversas poses que las mismas pueden adoptar, motivos por los cuales es necesario desarrollar un conjunto de características robusto que permita

## 5.7. ALGORITMO DE DETECCIÓN DE PERSONAS

---

diferenciar la forma humana del resto de la escena, de la manera más eficiente.

Para abordar el problema de detección de un objetivo, específicamente de personas, se considera el trabajo presentado en [23], en donde se adopta un procedimiento de detección basado en la obtención de descriptores de un objeto a partir del uso de histogramas de gradientes orientados para posteriormente alimentar con dicha información a una *SVM* lineal encargada de realizar la clasificación de dicho objeto para saber si se trata o no de una persona.

### 5.7.1. Histograma de Gradientes Orientados (*HOG*)

De acuerdo a [23], "los descriptores generados a partir del *HOG* son un recuerdo de los empleados dentro de los Histogramas de Orientación de Bordes y *SIFT* pero se procesan dentro de una malla de celdas espaciadas, utilizando normalizaciones de contrastes locales superpuestos para mejorar su rendimiento". A su vez, las características de una persona son descritas de manera global por un solo vector de características, en lugar de emplear múltiples vectores para describir pequeñas partes de la persona.

La fundamentación del algoritmo utilizado se basa en la idea de que la apariencia local de un objeto y su forma pueden ser caracterizados de una manera eficiente a partir de la distribución de los gradientes de intensidad local o las direcciones de los bordes. Lo anterior es logrado al dividir la imagen en pequeñas regiones llamadas "celdas", en donde para cada una de las mismas se obtiene un histograma unidimensional basado en las dirección de los gradientes a lo largo de los píxeles que conforman la celda.

Para obtener un cierto grado de invarianza con respecto a la iluminación, se acumula una medida de la energía del histograma local a lo largo de regiones espaciales mayores llamadas "bloques", utilizando de esta manera los resultados para normalizar todas las celdas que conforman a dicho bloque. Los bloques de descriptores normalizados son llamados "Descriptores de **HOG**". Una vez obtenidos los descriptores, los mismos son suministrados a una *SVM* para su clasificación. El proceso antes descrito se puede observar en la Figura 5.5.



Figura 5.5: Proceso de detección del objeto

Las características del detector utilizado para llevar a cabo este tipo de clasificación pueden ser consultadas en [23].



## Vuelo en formación

La idea fundamental de los sistemas conformados por múltiples robots consiste en la realización de distintas tareas de manera más eficiente por parte de un solo robot, o en dado caso, de tareas que un solo agente sería incapaz de realizar. Dentro de los sistemas multi-robot, una parte crucial consiste en el movimiento coordinado por parte de los agentes que conforman al sistema, de manera que los mismos mantengan una configuración aceptable a lo largo de la realización de la tarea especificada [4]. Como se estableció en la Sección 1.2, del Capítulo 1, lo anterior es logrado a partir del acoplamiento de los estados dinámicos de los agentes por medio de una ley de control común.

De manera particular, cuando menos un agente dentro de la formación debe por lo menos realizar alguna de las siguientes acciones [1]:

1. Seguir un estado deseado con respecto a algún otro agente dentro de la formación
2. La ley de control de seguimiento debe de depender mínimamente del estado del miembro antes mencionado

### 6.1. Teoría de Grafos

Un grafo  $X(\mathcal{V}, \mathcal{E})$  consiste en un conjunto de vértices o nodos  $\mathcal{V}(X)$  y un conjunto de aristas  $\mathcal{E}(X)$  en donde se define una arista como un par de vértices distintos pertenecientes al grafo  $X$ . Si se tiene una arista  $xy$ , se dice que que  $x$  y  $y$  son adyacentes, denotando esta relación con  $x\tilde{y}$ . Dentro de una arista, se dice que un nodo es "indicente" si es parte de dicha arista [25].

Para los casos en donde se busca representar una relación asimétrica, es necesario hacerlo mediante "grafos dirigidos", los cuales consisten en un par  $(\mathcal{V}, \mathcal{E})$ , en donde  $\mathcal{V} = \{1, \dots, n\}$  es un conjunto finito de nodos y  $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$  es un conjunto de aristas de pares ordenados de nodos. Se tiene que la arista  $(i, j) \in \mathcal{E}$  define que uno de los agentes en la formación, denominado  $j$ , puede obtener información del agente  $i$ , pero no en sentido contrario. Para el caso anterior, el agente o nodo  $i$  funge como "nodo padre" y el nodo  $j$  como el "nodo hijo"

[26]. La representación de un grafo dirigido se puede observar en la Figura 6.1

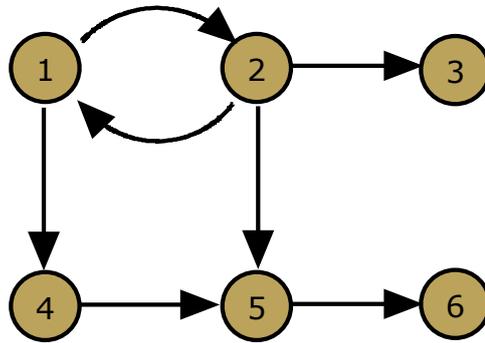


Figura 6.1: Representación de un grafo dirigido

Por otro lado, existe el caso "no dirigido", en donde las aristas  $(i, j)$  no se encuentran ordenadas, denotando de esta manera que el nodo  $i$  puede intercambiar información con el nodo  $j$  y viceversa.

### 6.1.1. Grafo dirigido de Dependencia

De acuerdo con [1], es posible definir un "grafo dirigido de dependencia", cuyos vértices o nodos representan a los agentes dentro de la formación. Una arista dirigida  $(i, j)$ , en donde el nodo  $i$  actúa como el agente "líder" y el nodo  $j$  funge como el agente "seguidor", se introduce dentro del grafo dirigido en dado caso de que la acción de control por parte de agente  $j$  sea una función del estado del agente  $i$ . La dependencia antes mencionada puede darse de 3 maneras:

1. Una función del estado relativo entre el agente  $i$  y el agente  $j$  es seguida por parte del agente  $j$ .
2. La trayectoria de referencia del agente  $j$  es una función del estado del agente  $i$ .
3. La acción de control retroalimentada del agente  $i$  es utilizada en el controlador del agente  $j$ .

Una representación de un grafo dirigido de dependencia se puede observar en la Figura 6.2.

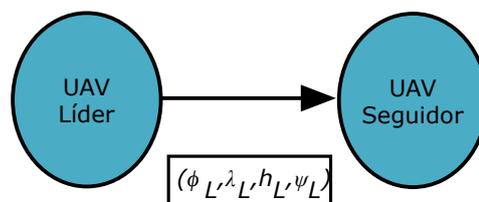


Figura 6.2: Representación de un grafo dirigido

en donde

## 6.2. NAVEGACIÓN EN EXTERIORES

---

- $\varphi_L \rightarrow$  Latitud del agente líder
- $\lambda_L \rightarrow$  Longitud del agente líder
- $h_L \rightarrow$  Altura del agente líder
- $\psi_L \rightarrow$  Ángulo *yaw* del agente líder

la cual es la información que es enviada desde el agente líder hacia el agente seguidor con la finalidad de actuar como referencia para el controlador implementado dentro del agente líder, con una cierta compensación para el caso del posicionamiento en el plano  $X$  y  $Y$  a partir de los datos de latitud y longitud.

## 6.2. Navegación en exteriores

### 6.2.1. Sistema de coordenadas geodésicas

El sistema de coordenadas geodésicas corresponde con el sistema utilizado por los sistemas de *GPS*, basado en el *WGS84*, siglas en inglés de *World Geodetic System-84*, con el cual es posible caracterizar y localizar un punto coordenado en cualquier posición sobre la superficie terrestre por medio de tres unidades dadas: longitud, latitud y altura, denotadas por  $\lambda$ ,  $\varphi$  y  $h$ , respectivamente. En dicho modelo, la Tierra es representada por medio de un elipsoide, teniendo un error máximo de 2 cm en la localización de un punto ([24], Cap. 2).

La medición de la longitud de un punto coordenado adoptada actualmente se toma a partir del Meridiano Primario, el cual corresponde actualmente con el Meridiano de Referencia *IERS* y donde se tiene un ángulo de  $0^\circ$ , y hasta otro meridiano que pase por dicho punto, teniendo valores que se encuentran en el rango de  $[-180^\circ, 180^\circ]$  en dirección hacia el Este u Oeste. A su vez, la medición de la latitud de un punto se considera a partir del plano ecuatorial en el cual se tiene un ángulo de  $0^\circ$  y el paralelo o plano perpendicular al eje de rotación, teniendo valores que se encuentran en el rango de  $[-90^\circ, 90^\circ]$  en dirección hacia el Norte o Sur. Por último, la altura corresponde con la distancia vertical entre el punto medido y el elipsoide de referencia.

Los parámetros asociados al marco geodésico son:

- $R_{Ea} = 6,378,137 \text{ m} \rightarrow$  Semi-eje mayor
- $R_{Eb} = 6,356,752.3142 \text{ m} \rightarrow$  Semi-eje menor
- $f = \frac{1}{298.257223563} \rightarrow$  Factor de Achatamiento
- $e = 0.08181919 \rightarrow$  Primera excentricidad
- $M_e = \frac{R_{Ea}(1-e^2)}{(1-e^2 \sin^2 \varphi)^{\frac{3}{2}}} \rightarrow$  Radio de curvatura del meridiano
- $N_e = \frac{R_{Ea}}{\sqrt{1-e^2 \sin^2 \varphi}} \rightarrow$  Radio de curvatura vertical primario

El sistema antes descrito se puede observar a continuación en la Figura 6.3.

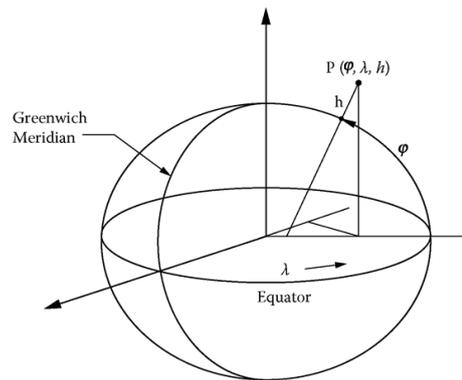


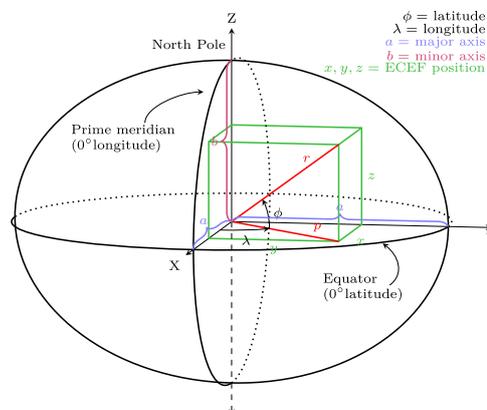
Figura 6.3: Sistema de Coordenadas Geodésicas

### 6.2.2. Sistema *ECEF*

El sistema de coordenadas *ECEF*, del inglés *Earth-Centered Earth-Fixed* consiste en un sistema cartesiano, el cual mantiene un movimiento de rotación con respecto a la Tierra, por lo cual las coordenadas de un punto en la superficie terrestre no cambian. Los ejes, así como el origen de este sistema coordenado se encuentran posicionados de la siguiente manera:

- Origen → Localizado en el centro de la Tierra
- Eje  $z$  → Localizado a lo largo del eje de rotación de la Tierra y en dirección al Polo Norte
- Eje  $x$  → Intersecta la Tierra en Latitud  $0^\circ$  y Longitud  $0^\circ$
- Eje  $y$  → Ortogonal tanto al eje  $x$  como al eje  $z$  de manera que se complete el sistema coordenado convencional al estilo de la regla de la mano derecha

El sistema antes descrito se puede observar a continuación en la Figura 6.4.

Figura 6.4: Sistema de Coordenadas *ECEF*

### 6.2.3. Sistemas de Coordenadas Fijos a la Superficie Terrestre

#### 6.2.3.1. Sistema de Coordenadas NED

Al igual que el sistema de coordenadas ECEF, este sistema de coordenadas plantea el uso de un sistema cartesiano, en donde se mantiene la regla de la mano derecha para el posicionamiento de sus ejes, siendo localizados de la siguiente manera:

- Origen  $\rightarrow$  Localizado en un punto arbitrario de la superficie terrestre
- Eje  $x$   $\rightarrow$  Apuntando hacia el elipsoide Norte
- Eje  $y$   $\rightarrow$  Apuntando hacia el elipsoide Este
- Eje  $z$   $\rightarrow$  Apuntado hacia abajo en dirección del elipsoide normal

Este sistema es el utilizado convencionalmente por los *UAV*, utilizando el mismo como punto de inicialización de los sensores abordo y de origen del mismo, obteniendo la distancia relativa del aeronave con respecto a dicho punto. El sistema antes descrito se puede observar a continuación en la Figura 6.5.

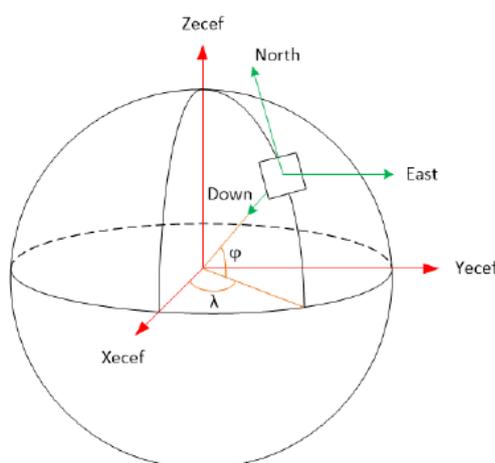


Figura 6.5: Sistema de Coordenadas *ECEF*

#### 6.2.3.2. Sistema de Coordenadas ENU

El sistema de coordenadas *ENU*, del inglés *East-North-Up*, es el adoptado convencionalmente como marco de referencia para vehículos terrestres, manteniendo la regla de la mano derecha para el posicionamiento de sus ejes, localizándose de la siguiente manera:

- Origen  $\rightarrow$  Localizado en un punto arbitrario de la superficie terrestre
- Eje  $x$   $\rightarrow$  Apuntando hacia el elipsoide Este
- Eje  $y$   $\rightarrow$  Apuntando hacia el elipsoide Norte
- Eje  $z$   $\rightarrow$  Apuntado hacia arriba

El marco coordenado descrito anteriormente se puede observar en la Figura 6.6.

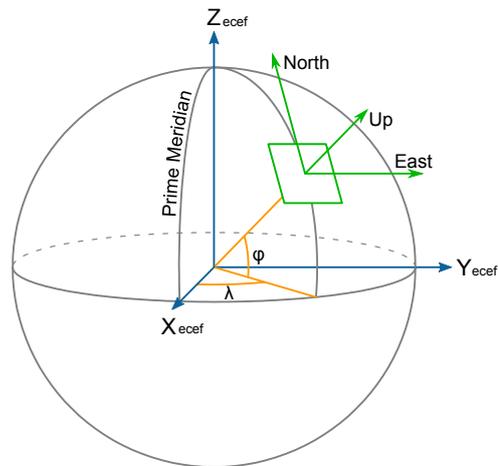


Figura 6.6: Sistema de Coordenadas *ENU*

Como fue mencionado anteriormente, los *UAV* utilizan convencionalmente el sistema de coordenadas *NED* como marco de referencia, por lo que es necesario realizar una transformación del sistema de coordenadas geodésico al marco *NED*. La transformación antes mencionada se puede analizar en el Apéndice A.

---

# Protocolo de Comunicación

---

## 7.1. Modelo de Referencia (*OSI*)

En la década de los setentas, debido al creciente uso de las computadoras y la subsecuente creación y establecimiento de redes de comunicación entre las mismas, la consolidación de estándares internacionales para el trabajo coordinado dentro de una red por parte de los agentes de la misma se volvió un tema de investigación y desarrollo muy importante; esto debido a la necesidad de emplear los recursos compartidos dentro de la red de la manera más eficiente y a todo su potencial. Para que dos o más sistemas puedan comunicarse, es necesario establecer previamente un conjunto de reglas en común por medio de las cuales pueda ser posible generar e interpretar la información que envíe o reciba cada uno de ellos [27].

En consecuencia de lo anterior, la *ISO*, del inglés *International Organization for Standardization*, creó en 1978 al subcomité *OSI*, del inglés *Open Systems Interconnection*, que para 1983, y tras diversas consideraciones, desarrolló y estableció el "Modelo de Referencia Básico" que se convirtió en un Estándar Internacional (*ISO 7498*), y por medio del cual se ideó una arquitectura basada en capas, por la que era posible descomponer un problema general en piezas más pequeñas y manejables. De acuerdo a lo establecido en el *ISO 7498*, los problemas dentro de una red de comunicación son tratados por medio de niveles de abstracción, de un nivel superior a uno inferior, teniendo de esta manera tres niveles que siguen este esquema, los cuales se pueden observar en la Figura 7.1 y corresponden con:

- Arquitectura o Modelo de Referencia *OSI*
- Especificaciones de Servicio *OSI*
- Especificaciones de Protocolo *OSI*

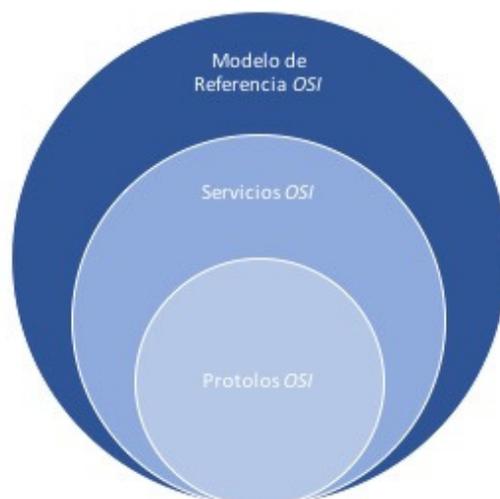


Figura 7.1: Niveles de abstracción en *OSI*

Dentro del punto de vista de *OSI*, se busca eliminar cualquier posible impedimento dentro de la comunicación establecida entre dos o más sistemas, sin importar el origen de los mismos, enfocándose de manera primordial en el intercambio de información en los puntos de interconexión de los mismos [27].

Aunque el modelo *OSI* define los componentes necesarios para llevar a cabo el intercambio de información entre varios sistemas, sin importar el origen de cada uno de ellos, el mismo no es implementado dentro de redes prácticas, debido a que funciona simplemente como un marco de trabajo por medio del cual es posible coordinar el desarrollo de estándares *OSI*, debido a su naturaleza de estratificación de tareas.

### 7.1.1. Arquitectura

Dentro de este nivel de abstracción "se definen los tipos de objetos que son utilizados para describir un sistema abierto, las relaciones generales entre los mismos y las restricciones generales entre estos tipos de objetos y en sus relaciones" [28]. A su vez, dentro de la especificación de la arquitectura según *OSI*, se especifica y analiza un modelo de siete capas para la comunicación entre procesos, el cual se encuentra construido a partir de los objetos antes mencionados, sus relaciones y sus restricciones, esto con la finalidad de establecer un marco de trabajo para el desarrollo de estándares que trabajen sobre el *ISO 7498*.

#### 7.1.1.1. Estratificación o Sistemas de Capas

De acuerdo a *OSI*, el uso de capas dentro de la arquitectura puede ser visto en una secuencia vertical, en donde cada una provee de servicios "inferiores" a cada uno de los servicios "superiores" a ella. De manera convencional, las capas se numeran de manera ascendente, de modo que la capa  $N$  utiliza los servicios denominados  $N - 1$  y a su vez provee de los servicios  $N$ , teniendo el uso completo de todos los servicios de capas inferiores al trabajar dentro de capas superiores. Dentro de cada capa existen diversas entidades que

## 7.1. MODELO DE REFERENCIA (OSI)

---

se encuentran distribuidas a lo largo de sistemas interconectados dentro de esa misma capa [27].

La representación de las siete capas que conforman el Modelo de Referencia *OSI* se puede observar en la Figura 7.2

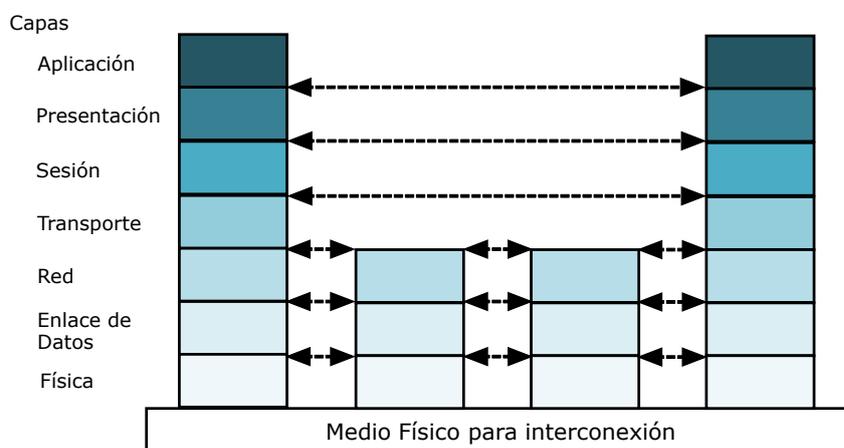


Figura 7.2: Arquitectura *OSI* de capas

### 7.1.1.1.1. Capa de Aplicación

Dado que es la capa de mayor abstracción, no provee de servicios a ninguna otra capa. Su principal tarea se centra en la semántica de la aplicación, entregando de esta manera la información recibida por capas inferiores a través de la interfaz de usuario. Todos los procesos de la aplicación residen dentro de la capa de aplicación [28].

### 7.1.1.1.2. Capa de Presentación

La capa de presentación se encarga de proveer de independencia a procesos de la aplicación con respecto a la diferencias en la representación de los datos, es decir, en la sintaxis. Para el proceso antes descrito, se realizan 3 operaciones elementales, las cuales son:

- Reformateo de datos
- Cifrado y Descifrado
- Compresión

El protocolo de la capa de presentación habilita al usuario para seleccionar un "Contexto de Presentación" deseado, el cual puede ser específico y dedicado para cada tipo de sistema [28].

### 7.1.1.1.3. Capa de Sesión

Esta capa provee de mecanismos para estructurar y organizar las posibles interacciones que se pueden presentar entre procesos dentro de una aplicación, de manera que a partir

de la misma se establecen, administran y finalizan las conexiones de usuario. Los mecanismos que se proveen permiten operaciones en ambos sentidos tanto simultáneas como alternas, la declaración de puntos de sincronización máximos y mínimos y la definición de tokens para intercambios estructurales [28].

### 7.1.1.1.4. Capa de Transporte

La finalidad de la capa de transporte consiste en proveer de transferencia de datos al sistema de manera transparente. Se encarga de asegurar que la información llegue sin error de un sistema a otro, dividiendo grandes paquetes en paquetes más pequeños para realizar una transmisión eficiente, los cuales son reensamblados, verificados para cualquier error posible y reconocidos en el punto de arribo [28].

### 7.1.1.1.5. Capa de Red

Dentro de esta capa se suministra independencia del tipo de tecnología implementada para la transmisión de información e independencia de consideraciones de retransmisión y enrutamiento. A su vez, esta capa se encarga de enmascararle a la capa de transferencia todas las características concernientes al medio de transferencia [28].

### 7.1.1.1.6. Capa de Enlace de Datos

La capa de enlace de datos provee de medios funcionales y procesales para transferir datos entre las entidades de una red y para detectar los posibles errores que se pudieran presentar dentro de la capa física. Los protocolos de enlace de datos y sus respectivos servicios se encuentran fuertemente relacionados y afectados por la tecnología física empleada para llevar a cabo la transferencia; debido a esto, es necesario el uso de protocolos que trabajen específicamente con las características requeridas por cada tecnología[28].

### 7.1.1.1.7. Capa de Física

La capa física provee de los medios mecánicos, eléctricos, funcionales y procesales para acceder al medio físico, permitiendo de esta manera la transmisión a nivel de bit entre diferentes dispositivos con tal de lograr una comunicación sincronizada entre los mismos [28].

## 7.1.2. Especificaciones de Servicio

Las especificaciones de servicio se basan en las demarcaciones de funciones entre cada una de las capas establecidas dentro de la arquitectura *OSI*. Considerando que los protocolos pueden ser vistos como una secuencia, teniendo un nivel de abstracción que va disminuyendo conforme se llega a las piedras angulares del mismo, los servicios son los encargados de proveer a las partes del sistema que se encuentra por encima de cierta demarcación, y que son llamadas "Usuarios del Servicio", una descripción del conjunto de capacidades, que las partes inferiores a

## 7.1. MODELO DE REFERENCIA (OSI)

---

dicha demarcación, son capaces de ofrecer. A su vez, para las partes inferiores de esa demarcación, llamadas "Proveedores del Servicio", sirve como un conjunto de restricciones de diseño.

El punto de interacción entre un Usuario del Servicio y un Proveedor del Servicio es llamado un "Punto de Acceso al Servicio", por medio del cual, a partir de la dirección del mismo, el usuario puede solicitar al proveedor el uso de los servicios de los cuales este dispone. Una representación esquemática de la interacción entre un Usuario y un Proveedor se puede observar en la Figura 7.3.

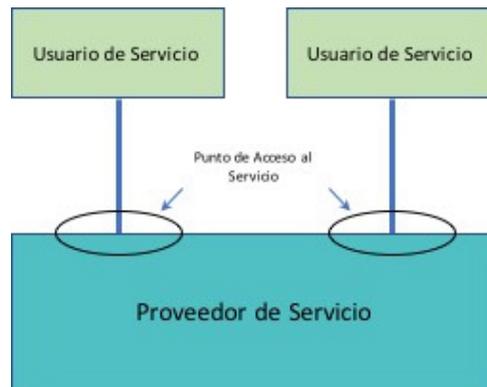


Figura 7.3: Modelo de Servicio OSI

### 7.1.3. Especificaciones de Protocolo

Para la correcta cooperación entre sistemas dentro de un servicio de comunicación distribuido es necesario establecer reglas por medio de las cuales dicha cooperación se lleve a cabo, las cuales constituyen un "Protocolo". Dentro de OSI, dicho protocolo dicta el procedimiento para utilizar servicios de capas inferiores con la finalidad de obtener servicios de capas superiores. En esencia, los protocolos "son una manera formal de verificar que las propiedades definidas de un servicio superior puedan ser derivadas de las reglas del protocolo y de las propiedades definidas de un servicio inferior" [27].

El llamado "Modelo de Protocolo" para OSI se puede observar en la Figura 7.4

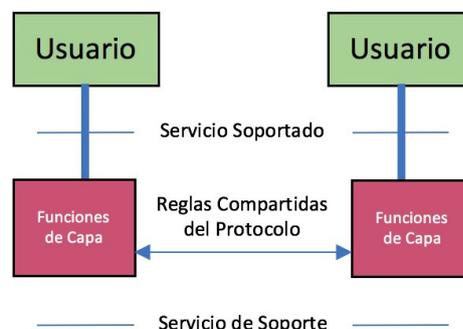


Figura 7.4: Modelo de Protocolo OSI

Una representación general de la Arquitectura, los Servicios y los Protocolos utilizados con base en *OSI* se puede observar en la Figura 7.5, en donde, a su vez, se muestran ciertos elementos básicos como son los Puntos de Acceso al Servicio y las Entidades.

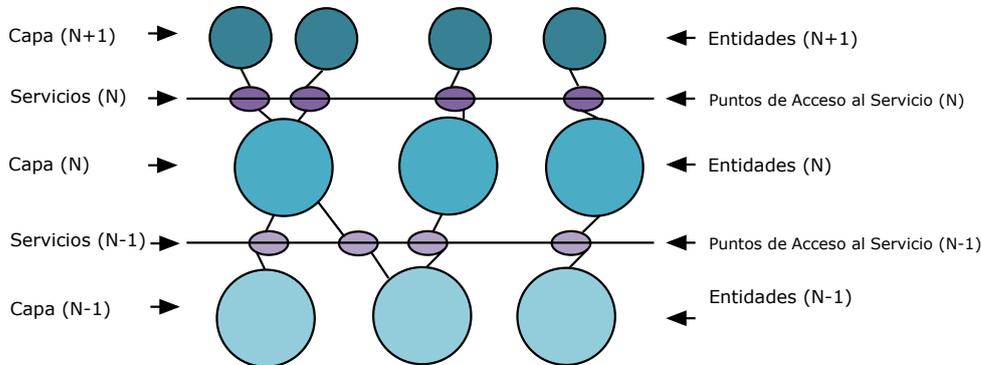


Figura 7.5: Representación de elementos de *OSI*

## 7.2. Protocolos de Acceso Múltiple

Existen principalmente tres componentes a considerar dentro de cualquier red de comunicación, los cuales corresponden a:

- Nodos
- Canales
- Protocolos

Los nodos son las estaciones físicas que alojan los datos a comunicar y sus funciones. Los protocolos corresponden a las reglas y acuerdos que rigen el comportamiento de los nodos dentro de la red. Y por último, los canales son el medio físico por el cual las señales, que representan la información transmitida, son enviadas de un nodo a otro.

Las redes de comunicación convencionales hacen uso de canales punto a punto, de manera que un nodo se comunica con otro nodo, sin ningún tipo de interferencia, inclusive si alguno de dichos nodos se encuentra comunicado con algún otro nodo. La desventaja inherente dentro de este tipo de canales recae en la posible modificación de la topología implementada, la cual se encuentra planteada de manera fija desde la etapa de diseño y por ende, cualquier tipo de cambio dentro de dicha topología implicaría un costo mayor y un trabajo arduo para poder ser llevado a cabo. Debido al inconveniente antes mencionado y a la posible necesidad de realizar transmisiones de información a múltiples usuarios dentro de la red de comunicación, la implementación de un canal de difusión es una posibilidad adoptada dentro de la etapa de diseño de la red.

La posibilidad de que una transmisión dentro del canal coincida en tiempo con alguna otra es una característica a considerar durante el uso de este tipo de esquemas de comunicación, debido a que se puede dar el caso en que ninguna de ellas llegue a su destino, perdiendo de esta

## 7.2. PROTOCOLOS DE ACCESO MÚLTIPLE

---

manera robustez dentro de la red. A su vez, debido a que de manera convencional un paquete de información se encuentra destinado a un cierto nodo o receptor, el que dicho paquete sea recibido por diversos nodos para quienes no se encuentra destinado, resulta en el uso de recursos por parte de dichos nodos con la intención de procesar dicha información y finalmente desecharla.

Con base en lo anterior, el concepto de "método de acceso múltiple" toma relevancia, debido a que el mismo se refiere a la manera en la cual diversas terminales o nodos, conectados a un medio de transmisión multipunto, son capaces de compartir los recursos y las capacidades del mismo, pudiendo de esta manera enviar información de un nodo a otro.

Es necesario evitar o cuando menos controlar, en la medida de lo posible, cualquier tipo de interferencia que se pueda presentar dentro del canal, ya que este representa el recurso que es compartido por cada uno de los nodos dentro de la red. Dicho lo anterior, la implementación de esquemas de acceso al canal por parte de los nodos es un elemento importante dentro de la red de comunicación. Dichos esquemas son conocidos como *Protocolos de Acceso Múltiple* ([29], Cap. 1).

Considerando el modelo de referencia *OSI* descrito en secciones anteriores, los Protocolos de Acceso Múltiple forman parte de una capa intermedia situada entre la Capa de Enlace de Datos y la Capa Física. Dicha subcapa se conoce como "Capa de Control de Acceso al Medio".

Dentro de los protocolos de acceso múltiple se realiza la consideración de que el canal representa el punto central en los mismos y los nodos representan los usuarios que transmiten información a través del primero. Tomando en cuenta lo anterior, es posible clasificar de manera esquematizada los tipos de protocolos implementados, como se puede observar en la Figura 7.6.

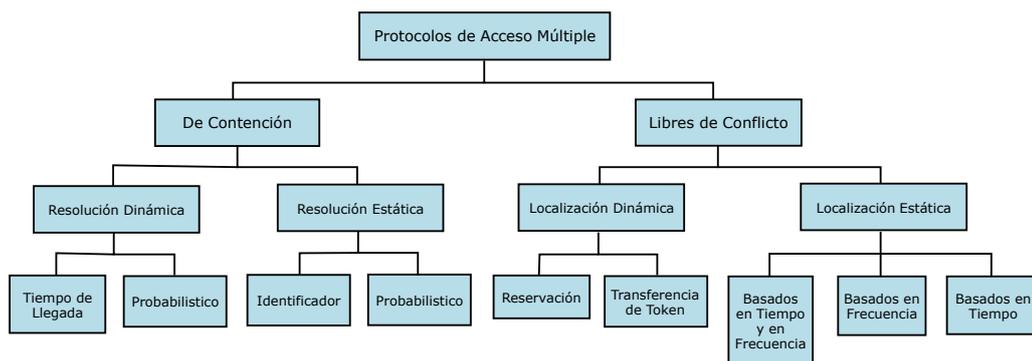


Figura 7.6: Clasificación de Protocolos de Acceso Múltiple

Dentro de esta clasificación se pueden diferenciar dos tipos de protocolos:

- De Contención
- Libres de Conflicto

### 7.2.1. De Contención

En este tipo de protocolos, un usuario que se encuentra transmitiendo no tiene garantía de una operación exitosa, debido a que en el momento en el cual el mismo, junto con otros usuarios dentro de la red, se encuentra transmitiendo información dentro del canal compartido de manera simultánea, se puede presentar una colisión y la información no puede ser recibida de manera correcta. Cuando acontece el caso antes descrito, es necesario retransmitir nuevamente el paquete de datos hasta que sean recibidos correctamente de manera eventual, en donde el tiempo de espera para realizar la operación es elegido de manera aleatoria ([29], Cap 3).

En los esquemas de contención, los usuarios que se encuentran disponibles no transmiten información, motivo por el cual no consumen ningún porcentaje de los recursos del canal.

### 7.2.2. Protocolos Libres de Conflicto

Dentro de este tipo de protocolos se asegura que la transmisión sea exitosa, sin importar cuándo sea realizada, es decir, no se presenta interferencia entre la misma y alguna otra dentro del canal. Lo anterior se logra asignando el canal para los usuarios, sin ninguna superposición entre las porciones del canal destinadas a diferentes usuarios.

Una ventaja que se presenta en este tipo de protocolos consiste en la habilidad de asegurar equidad entre los diversos usuarios del canal y la capacidad de controlar el retraso de paquetes de información [29].

### 7.2.3. Protocolos de Detección de Portadora

Dentro de los protocolos de contención antes mencionados, los esquemas de detección de portadora se basan en un comportamiento comúnmente llamado "Escuchar antes de enviar", en el cual cada usuario dentro de la red "escucha" si algún otro usuario está utilizando el canal en esos momentos, antes de enviar información y hacer uso del mismo. Esto con la finalidad de evitar una posible colisión del paquete que se pretende enviar con respecto al que está siendo enviado en esos momentos.

El proceso de escuchar el canal no implica recibir la información enviada por otro usuario, simplemente consiste en detectar la portadora que se encuentra presente cuando las señales son transmitidas. A pesar de la ventaja que presentan este tipo de protocolos para no ocupar el canal si este se encuentra ocupado, estos no garantizan que no se produzcan colisiones dentro del mismo, debido a que si el canal se encuentra disponible, dos o más usuarios pueden detectar el canal libre e intentar transmitir información, resultando en colisión de paquetes. Esto se debe al retardo presente en la transmisión por parte de un usuario, lo cual conlleva a que la portadora de dicho usuario no sea detectada de manera inmediata por algún otro usuario dentro de la red.

Existen diversas versiones de protocolos de acceso múltiple por detección de portadora (CSMA), pero todos mantienen la misma filosofía: en el momento en el cual un usuario genera un nuevo paquete y pretende transmitirlo, primeramente se detecta el canal para determinar su estado. Si

## 7.2. PROTOCOLOS DE ACCESO MÚLTIPLE

---

el canal se encuentra disponible, el paquete es transmitido de manera inmediata. En el caso de que se presente una colisión, todos los usuarios que se encuentran transmitiendo reagendan una retransmisión del paquete que colisionó, en donde el tiempo de espera para llevar a cabo dicha retransmisión se elige de manera aleatoria ([29], Cap 4).

### 7.2.3.1. Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA)

De las diversas problemáticas que pueden presentarse dentro de las comunicaciones inalámbricas, se tienen dos que son tratadas hasta cierto punto a través de la implementación del protocolo CSMA/CA:

- Estación Escondida
- Estación Visible

#### 7.2.3.1.1. Estación Escondida

Para la primera, se tiene una estación que puede no ser visible para alguna otra o múltiples dentro de la red, pero si para el resto. Debido a lo anterior, es posible que una transmisión que está siendo realizada en una parte de la red puede que no sea recibida por alguna otra parte de la misma y de manera errónea, algún agente en esa parte de la red comience a transmitir.

Es posible ejemplificar lo antes descrito por medio de la Figura 7.7 (a), en donde la estación C se encuentra transmitiendo a la estación B. Dado que la estación A se encuentra fuera del rango de transmisión de la estación C, en el momento en que detecte el canal, no detectará la portadora de dicha estación, por lo cual concluirá que el canal se encuentra disponible y comenzará a transmitir a la estación B.

#### 7.2.3.1.2. Estación Visible

Para el caso de la segunda problemática, un agente de la red puede que intente transmitir a otro agente dentro de su rango pero al escuchar el canal, este detecte que el mismo está siendo usado, por lo cual cesa su transmisión, sin darse cuenta de que el agente que se encuentra transmitiendo, lo hace a algún otra parte de la red.

Para ejemplificar esta posible situación, se hace uso de la Figura 7.7 (b), en donde la estación B pretende transmitir a la estación C, por lo cual escucha el canal. Dado que B detecta la portadora de A, el mismo decide posponer su transmisión, sin estar al tanto de que la estación A puede que se encuentre transmitiendo a alguna otra estación no visible para B.

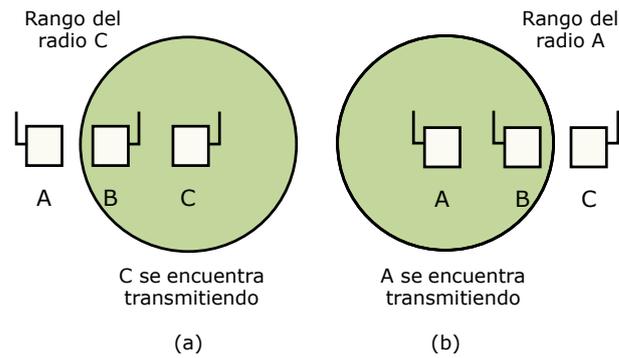


Figura 7.7: (a) Problema de la estación escondida. (b) Problema de la estación visible

Al implementar el protocolo *CSMA/CA*, se realiza tanto una detección del canal de manera física como de manera virtual. La detección física consiste en escuchar el canal para saber si se encuentra disponible u ocupado, si se encuentra disponible, la estación comienza a transmitir, proceso durante el cual no detecta nuevamente el canal, lo que puede ocasionar que el paquete sea corrompido. Si el canal se encuentra ocupado, el transmisor reagenda la transmisión hasta que detecte que el canal se encuentra disponible. La potencial corrupción del paquete de datos, puede evitarse haciendo uso de la detección virtual, donde se hace uso de algunos paquetes de control adicionales, como se explica a partir del ejemplo siguiente.

Como se puede observar en la Figura 7.8, se plantea el escenario en el cual el agente *B* se encuentra en el rango del agente *A*. El agente *C* también se encuentra en el rango de *A*. El agente *D* se encuentra en el rango de *B* pero no de *A*.

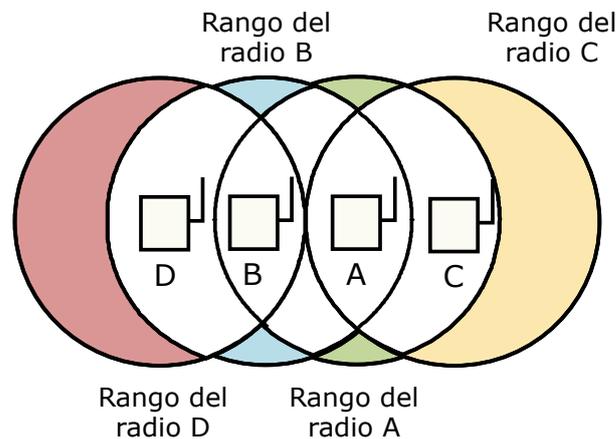


Figura 7.8: Ejemplo de la detección virtual

El uso del protocolo comienza en el momento en que *A* se propone a realizar una transmisión de información a *B*, para lo cual envía en un principio una trama denominada *RTS*, del inglés *Request To Send*, hacia *B* informándole de su intención de enviarle información. Al recibir dicha trama, *B* puede permitir la recepción de información, enviando de regreso a *A* una trama denominada *CTS*, del inglés *Clear To Send*. Una vez que *A* recibe la trama anterior, el mismo

### 7.3. CONSTITUCIÓN DE LA TRAMA DE LA CAPA DE ENLACE DE DATOS

comienza la transmisión de información, dando inicio a un contador *ACK*, del inglés *Acknowledge*. Si la información fue recibida de manera correcta, *B* responde a *A* con una trama *ACK*.

Considerando el mismo proceso visto desde los agentes *C* y *D*, se tiene que *C*, al encontrarse en el rango de *A*, recibe la trama *RTS*, por lo cual decide no transmitir información para no ocasionar una posible colisión. De acuerdo a la información integrada en la trama *RTS*, es posible estimar el tiempo que tomará la transmisión, incluyendo la trama *ACK*, en finalizarse, por lo cual comienza un proceso de "canal virtual ocupado" para si mismo, en donde se genera el llamado *NAV*, del inglés *Network Allocation Vector*.

Debido a que el agente *D*, no se encuentra dentro del rango de *A*, no recibe la trama *RTS*, pero al encontrarse dentro del rango de *B*, recibe la trama *CTS*, motivo por el cual comienza el mismo proceso que el agente *C* [31]. La representación de las tramas enviadas y recibidas, a lo largo del tiempo, por parte de los nodos dentro de la red se puede observar en la Figura 7.9.

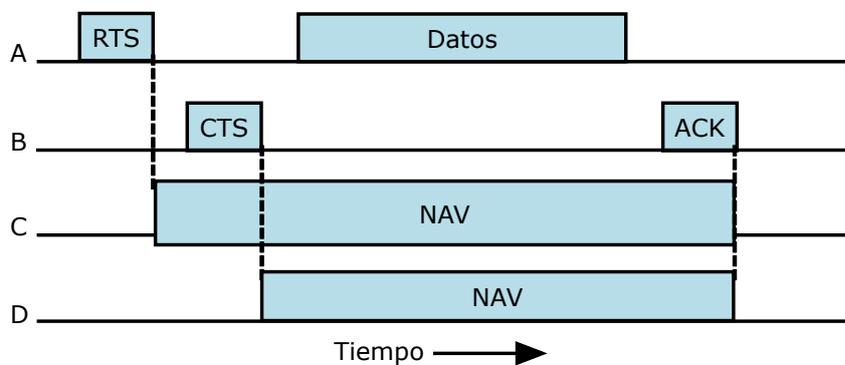


Figura 7.9: Detección virtual de canal utilizando *CSMA/CA*

### 7.3. Constitución de la trama de la capa de enlace de datos

Considerando la arquitectura de capas establecida en el modelo de referencia *OSI* el intercambio de información a nivel de bit se da entre los agentes de una red por medio de la capa física, en la cual no se le da importancia al significado de los ceros y unos transmitidos o si ocurrió algún error durante la transmisión. Dado que la información transmitida será utilizada por el resto de las capas superiores en el o los nodos receptores, y por ende por los servicios intermedios entre ellas, es necesario brindar cierto orden a la misma, de manera que los posibles errores dentro de una transmisión puedan ser corregidos.

La manera común de realizar esto, de acuerdo al modelo *OSI*, es agrupando la información en tramas de datos, los cuales cuentan con campos específicos que cumplen algún propósito dentro de la red ([30], Cap. 1). Dicho orden se establece dentro de la capa de enlace de datos por medio de la operación de conmutación de paquetes.

Un posible bloque de datos puede estar conformado por los campos mostrados en la Figura 7.10, en donde se cuenta con un campo destinado a alojar la dirección del nodo destino, uno para la dirección del nodo que envía la información, uno o más campos destinados a almacenar

la información a enviar y posteriormente un campo utilizado para detectar algún posible cambio o corrupción de la información dentro del bloque de datos.

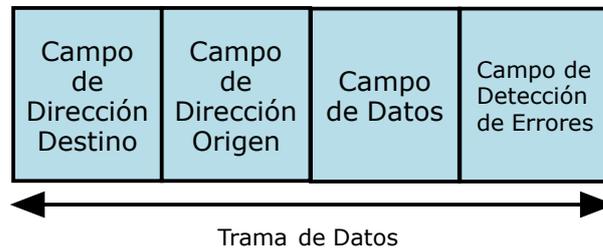


Figura 7.10: Formato de bloque de datos

La idea principal detrás de las tramas de datos es el proporcionar ciertas entidades dentro del mensaje transmitido, las cuales puedan ser revisadas para detectar posibles errores a lo largo de la transmisión. Para lograr lo anterior, es necesario que los nodos dentro de la red se comporten de acuerdo a una serie de reglas preestablecidas para el envío y recepción de tramas de datos.

### 7.3.1. Tramas orientadas a bits

La definición de orientación a bits consiste en que cada trama de datos puede contener cualquier número de bits, sin contar necesariamente con algún tipo de delimitador entre los diversos campos que forman la trama, pero si con un patrón binario único al principio y al final de la trama. Estos patrones son denominados "bandera de apertura" y "bandera de cierre". En la Figura 7.11 se puede observar una trama de datos orientada a bit, en donde se tienen campos destinados a alojar cierta información ([30], Cap 4).

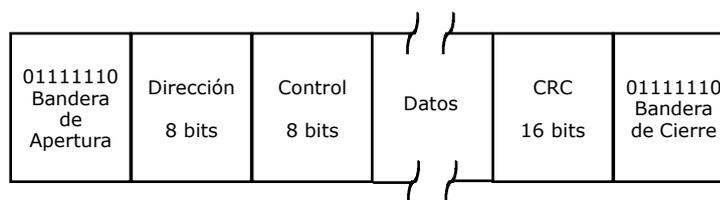


Figura 7.11: Trama orientada a bits

### 7.3.2. Trama de datos

Una trama de datos corresponde a un paquete de bits de información codificados, el cual se envía desde un transmisor a través de un medio de transmisión (guiado o no guiado) hacia un receptor, para que este realice la operación de decodificación con la finalidad de posteriormente trabajar sobre la información recibida.

La información concerniente al estado, tanto en posición como en orientación, de cada una de las plataformas utilizadas es la que se encuentra compartida entre las mismas y, a su vez,

### 7.3. CONSTITUCIÓN DE LA TRAMA DE LA CAPA DE ENLACE DE DATOS

---

con la *GCS* por medio de una serie de tramas de datos interpretados de forma hexadecimal, en donde cada campo dentro de la misma tiene asignada alguna función. Un ejemplo de lo anterior corresponde a:

**FE 15 CC 01 DD 01 04 00 00 CC 01 CC 02 FE 05 01 02 03 04 EF 00 00 EF**

En la Tabla 7.1 se describe el funcionamiento de cada campo dentro de la trama antes mostrada.

<b>Campo</b>	<b>Ejemplo</b>	<b>Función</b>
1	FE	Bandera de apertura
2	15	Tamaño de la trama
3 y 4	CC 01	Dirección <i>MAC</i> del dispositivo que envía
5 y 6	DD 01	Dirección <i>MAC</i> del módulo de radiofrecuencia
7	04	Comando
8 y 9	00	Reservados
10 y 11	CC 01	Dirección <i>MAC</i> del dispositivo que envía
12 y 13	CC 02	Dirección <i>MAC</i> del dispositivo destino
14	FE	Bandera de apertura de la trama interna
15	05	Tamaño de la trama interna
16, 17, 18, y 19	01 02 03 04	Información enviada
20	EF	Bandera de cierre de la trama interna
21 y 22	00 00	<i>CRC</i>
23	FE	Bandera de cierre

Tabla 7.1: Constitución de la trama de datos y función de cada campo

El ejemplo anterior considera una longitud de 4 bytes para la información que se desea enviar. Este tamaño puede variar dependiendo de la cantidad de datos que se requieran enviar de un agente a otro.

#### 7.3.3. Control de flujo de datos

El control de flujos de datos es una técnica implementada para evitar que un agente transmisor sobrecargue con una cantidad excesiva de datos a un agente receptor, esto debido a que el mismo cuenta con una zona de memoria temporal destinada a almacenar los datos que están siendo recibidos. Antes de que dichos datos puedan ser redirigidos a niveles superiores de la aplicación para su uso, el receptor realiza cierto procesamiento sobre los mismos, los cuales podrían perderse debido al posible desbordamiento de la memoria temporal, si es que no existieran procedimientos para el control de flujo de datos [32].

Como se especificó en la subsección (7.2.3.1.2), dentro del protocolo de acceso múltiple utilizado en el prototipo (*CSMA/CA*) se cuenta con un método de control de flujo de datos basado en el envío y recepción de las tramas *RCS*, *CTS* y *ACK* por parte de los diversos nodos de la red.

### 7.3.4. Control de Errores

El control de errores se centra en los mecanismos que pueden ser implementados para detectar y corregir los posibles errores que puedan presentarse en la transmisión de tramas de un nodo a otro.

La detección de errores consiste en percibir alguna posible corrupción dentro de un paquete de datos enviados de un transmisor a un receptor. Como se estableció anteriormente, a partir de la conmutación de paquetes se establece cierto orden a los datos transmitidos de un nodo a otro, de manera que los posibles errores que se pudiesen presentar dentro de la transmisión, los cuales pueden deberse a ruido eléctrico o algún otro tipo de interferencia dentro del canal, puedan ser detectados y corregidos.

Todas las técnicas de detección de errores consisten en añadir bits adicionales a una trama de datos que se pretende enviar por parte del transmisor, con la finalidad de generar un código con capacidad de detectar errores. Dicho código se calcula de acuerdo a los bits que se desean transmitir. Una vez recibidos los datos, el receptor efectúa el mismo cálculo y compara ambos resultados, detectando un error en la transmisión si ambos resultados no coinciden [32]. En la Figura 7.12 se puede observar una representación del proceso de detección de errores.

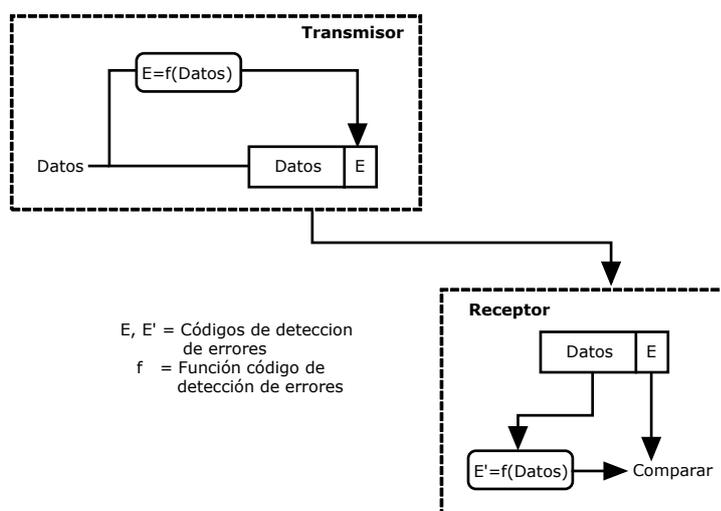


Figura 7.12: Detección de errores

Dentro del protocolo de comunicación implementado en el prototipo descrito en el Capítulo 3, se cuenta con un método de detección de errores basado en un *CRC*, del inglés *Cyclic Redundancy Check* y un método de corrección de errores que utiliza una *ARQ*, del inglés *Automatic Repeat Request*.

#### 7.3.4.1. Solicitud de Repetición Automática (ARQ)

Este tipo de método se basa en el uso de mensajes de reconocimiento enviados del nodo receptor al nodo transmisor, con la finalidad de validar la recepción correcta de los paquetes de datos

### 7.3. CONSTITUCIÓN DE LA TRAMA DE LA CAPA DE ENLACE DE DATOS

---

transmitidos. De presentarse el caso en que un paquete haya sido corrompido durante el proceso de transmisión, el nodo receptor es capaz de detectar dicha falla a través de un método de detección de errores, para posteriormente solicitar una retransmisión del paquete por parte del nodo transmisor. Existen 3 variantes del *ARQ*, que corresponden a:

- *ARQ* con parada-y-espera
- *ARQ* con vuelta-atrás-N
- *ARQ* con repetición selectiva

#### 7.3.4.2. Código de Redundancia Cíclica (*CRC*)

Los códigos de redundancia cíclica se basan en la creación e implementación de una firma única que caracteriza a una trama específica de ceros y unos. Si se tiene un bloque de  $k$ -bits, el transmisor genera una secuencia de  $n$ -bits llamada "secuencia de comprobación de trama", de manera que la trama a enviar, la cual contiene  $k + n$  bits, se divide por algún número previamente determinado. Una vez que se reciben los datos, el receptor divide la trama recibida entre ese número y si no se cuenta con resto en dicha operación, se puede considerar que no se han presentado errores en la transmisión. El proceso anterior se encuentra detallado en el Apéndice B.

Para el protocolo de comunicación establecido dentro del prototipo, se hace uso del *CRC-16-IBM*, el cual usa el polinomio  $1 + x^2 + x^{15} + x^{16}$ , cuya representación hexadecimal corresponde con *0xA001*.

### 7.3.5. Características del protocolo implementado

#### 7.3.5.1. Calendarización de envío de paquetes

Para el envío de paquetes de un nodo a otro dentro de la red formada por los prototipos cuadrirrotors, líder y seguidor, y la *GCS*, se realiza una calendarización de dicha operación, en donde la prioridad más alta se le asigna a ambos *UAV*, teniendo un tiempo de envío de datos entre los mismos de 50 ms, para posteriormente enviar y recibir información por parte de la *GCS* cada 200 ms.

Una representación del control de flujo de datos implementado se puede observar en la Figura 7.13, esquematizada por medio de un grafo no dirigido.

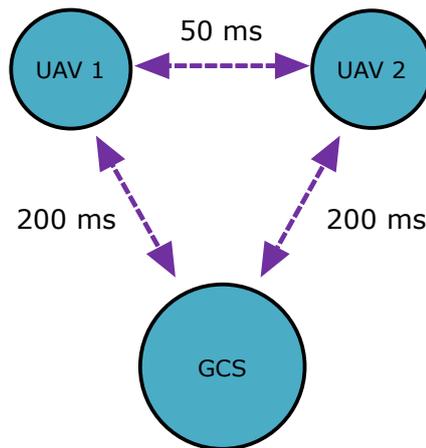


Figura 7.13: Diagrama del control de flujo de datos

### 7.3.5.2. Información de los dispositivos dentro de la red

A partir de los módulos de radiofrecuencia, es posible obtener información sobre el estado de los demás nodos en la red, como es:

- Potencia de la señal recibida
- Calidad del enlace

A partir de la potencia de la señal recibida, es posible estimar la distancia entre dos módulos de radiofrecuencia a partir del modelo de propagación de espacio libre (Ver Apéndice C). Con dicha distancia estimada es posible retroalimentar la posición de un *UAV* con respecto al otro y conjuntar dicha estimación con la presentada en el Apéndice B, para realizar la tarea de seguimiento entre ambos vehículos.

# Simulación y Resultados Experimentales

## 8.1. Controlador

Como se mencionó con anterioridad, el controlador implementado para la estabilización del prototipo tanto en orientación como en posición consiste en un controlador de tipo *PID* discreto; la acción de dicho controlador sobre el modelo simplificado del cuadri-rotor que funge como el agente líder dentro de la formación y posteriormente, la acción del mismo tipo de controlador sobre el cuadri-rotor que actúa como el agente seguidor dentro de la misma configuraciónn fue simulada por medio del entorno de programación *Simulink* de *MATLAB*. El esquemático a bloques del modelo desarrollado se puede observar en la Figura 8.1.

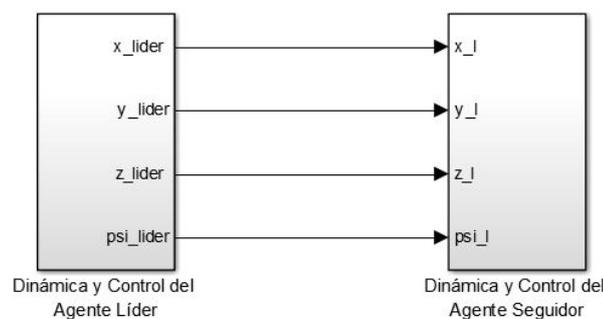


Figura 8.1: Esquemático a bloques del modelo Líder-Seguidor

Debido a que dentro de la configuración de líder-seguidor se considera el estado actual del líder, tanto en orientación como en posición, para ser observado y seguido con cierta compensación por parte de los agentes dentro de la misma, se considera una trayectoria a seguir por parte de los agentes dentro de la formación, la cual consta de ciertas etapas, las cuales son:

1. Despegue controlado
2. Modo Hover
3. Traslación de un punto A a un punto B
4. Modo Hover

## 5. Aterrizaje controlado

Debido a lo anteriormente planteado, la dinámica a seguir por parte de ambos cuadri-rotors se ve constantemente afectada debido al cambio de referencia necesario para completar cada una de las etapas planteadas para la trayectoria a seguir, por lo cual dentro de las distintas gráficas mostradas en los siguientes apartados se puede apreciar un cambio de referencia tanto para las relacionadas con la posición tridimensional, como para la orientación angular.

## 8.2. Controlador de Orientación

### 8.2.1. Controlador de Orientación para el agente Líder

Dentro de la simulación realizada, se consideró una condición inicial de  $30^\circ$  para el ángulo de yaw ( $\psi$ ), esto con la finalidad de comprobar la acción del controlador para llevar de un estado actual  $\psi$  a un estado  $\psi_{des}$ , buscando que  $\psi \rightarrow \psi_{des}$  a medida que  $t \rightarrow \infty$ . La curva de respuesta para el ángulo de yaw del agente líder se puede observar en la Figura 8.2. Tanto para el ángulo de pitch ( $\theta$ ) como de roll ( $\phi$ ) se consideró una condición inicial de  $0^\circ$ . Las curvas de respuesta para el ángulo de pitch y de roll del agente líder se pueden observar en las Figuras 8.3 y 8.4 respectivamente.

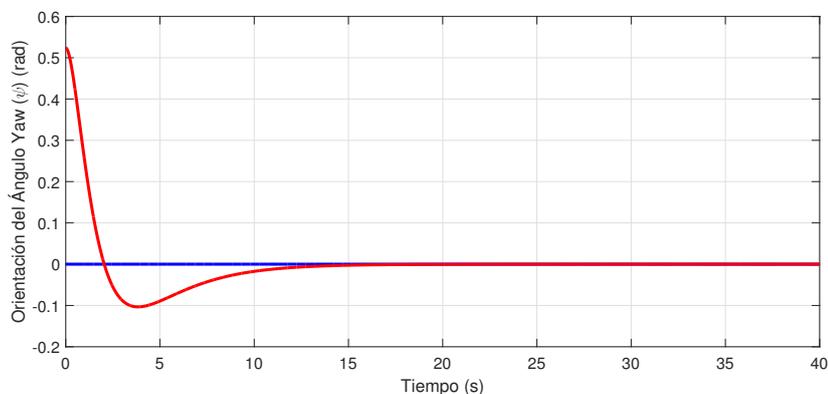


Figura 8.2: Respuesta del controlador para orientación del Ángulo Yaw ( $\psi$ ) del agente Líder

### 8.3. CONTROLADOR DE POSICIÓN

---

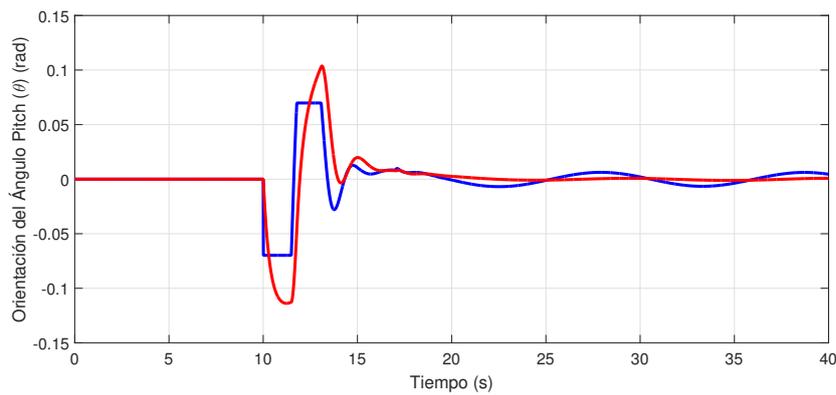


Figura 8.3: Respuesta del controlador para orientación del Ángulo Pitch ( $\theta$ ) del agente Líder

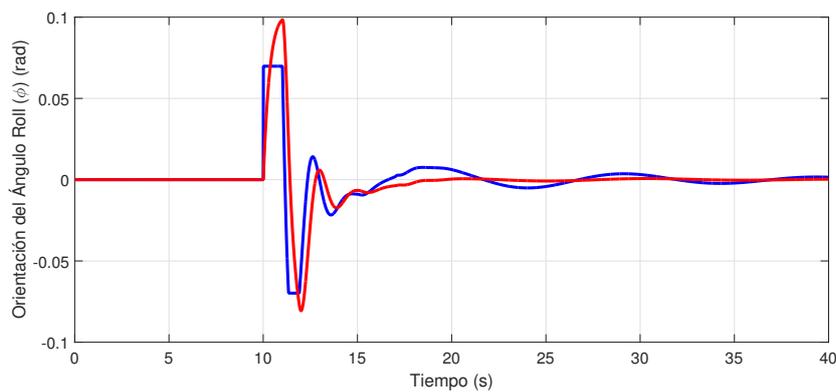


Figura 8.4: Respuesta del controlador para orientación del Ángulo Roll ( $\phi$ ) del agente Líder

## 8.3. Controlador de Posición

### 8.3.1. Controlador de Posición para el agente Líder

Para el controlador de posición del agente líder, se toman en cuenta los ángulos pitch ( $\theta_{des}$ ) y roll ( $\phi_{des}$ ) calculados anteriormente en el Capítulo 4 para realizar la traslación del UAV líder a una cierta posición deseada. La curva de respuesta para la posición en el eje  $X$ , el eje  $Y$  y el eje  $Z$ , se pueden observar en las Figuras 8.5, 8.6 y 8.7, respectivamente.

## 8. SIMULACIÓN Y RESULTADOS EXPERIMENTALES

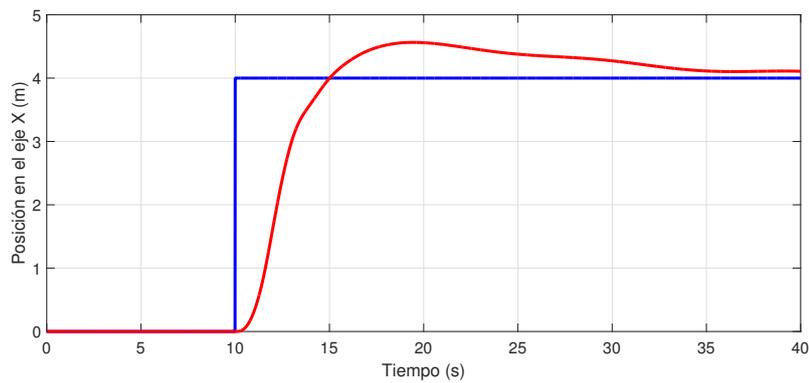


Figura 8.5: Respuesta del controlador para posición en el Eje X del agente Líder

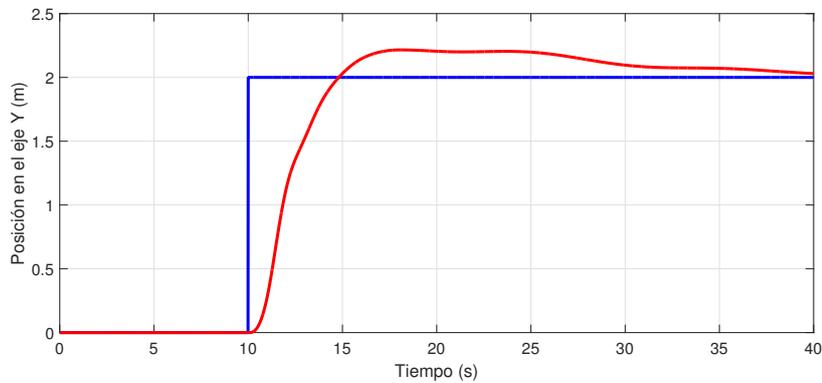


Figura 8.6: Respuesta del controlador para posición en el Eje Y del agente Líder

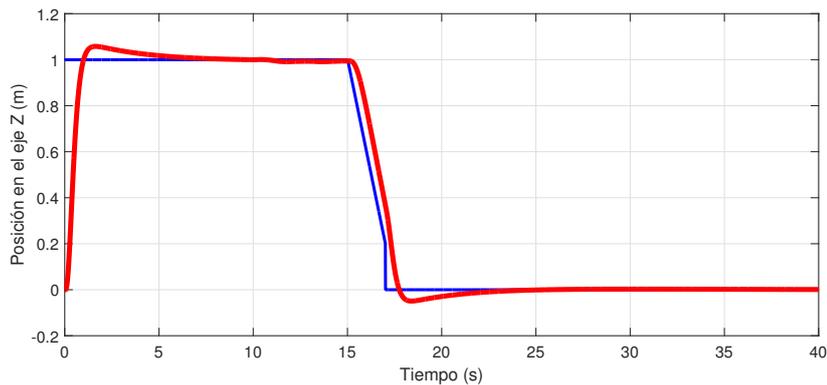


Figura 8.7: Respuesta del controlador para posición en el Eje Z del agente Líder

La posición tridimensional de ambos agentes (Líder y Seguidor) se puede observar en la Figura 8.8, en la cual se puede apreciar que existe una cierta compensación del estado del agente Seguidor con respecto al estado del agente Líder, como se establece dentro de la arquitectura implementada. A su vez, la posición bidimensional de los mismos agentes se puede apreciar en la Figura 8.9, en la cual se cumple la consideración antes establecida.

### 8.3. CONTROLADOR DE POSICIÓN

---

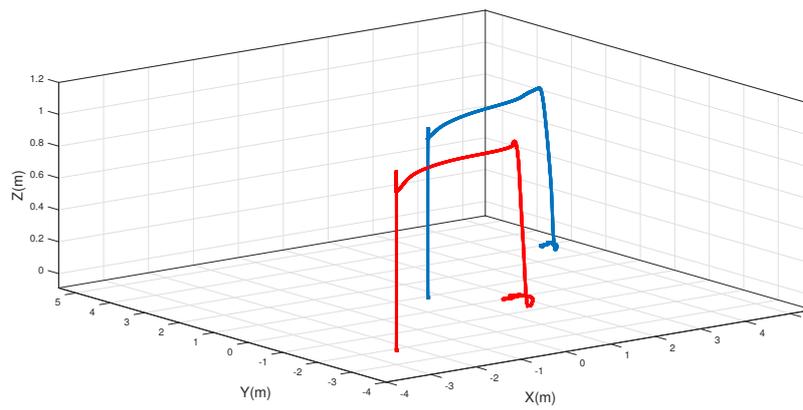


Figura 8.8: Posición tridimensional de los agentes Líder y Seguidor dentro de la formación

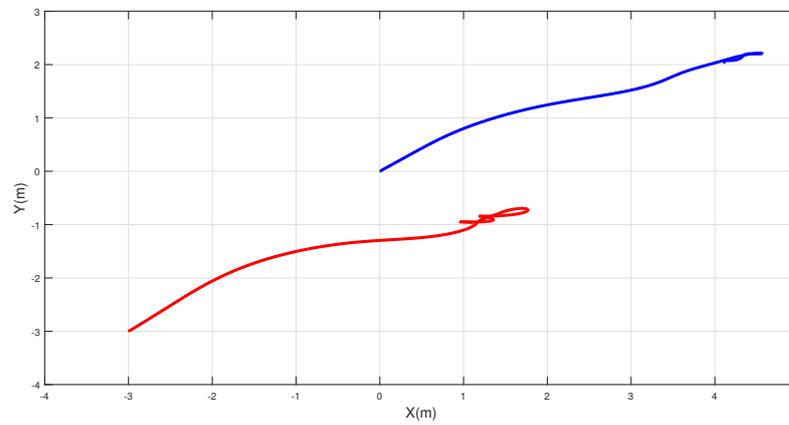


Figura 8.9: Posición biimensional de los agentes Líder y Seguidor dentro de la formación



---

## Conclusiones y Trabajo a Futuro

---

Dentro del presente trabajo de tesis se realizó la obtención de datos en tiempo real a través de la interfaz descrita en el Capítulo 3, con lo cual fue posible analizar el comportamiento del prototipo en vuelo, eliminando la necesidad almacenar los datos de vuelo dentro de la tarjeta *microSD*, teniendo que terminar la tarea en ejecución para poder analizar los datos *offline*. A su vez, a partir de la interfaz fue posible sustituir el uso de un radiocontrol para pilotear la aeronave, pudiendo de esta manera establecer comandos y tareas específicas a partir de tramas de información enviadas desde la *GCS* hacia el *UAV*.

La ley de control antes descrita realizó la tarea de regulación de posición y orientación del prototipo de manera satisfactoria, trabajando en conjunto con el cálculo de los ángulos deseados para realizar la traslación del prototipo a una posición deseada, teniendo como restricción la saturación de dichos ángulos con la finalidad de mantener al prototipo dentro de un área de operación lineal.

El algoritmo de visión artificial entregó muy buenos resultados en el momento de detectar una persona dentro del área de visión del prototipo, teniendo las restricciones de contar con una representación de cuerpo completo de la persona o no encontrarse obstruida por algún otro objeto. Dentro de este algoritmo, es necesario considerar el costo computacional que representa el analizar una mayor cantidad de escalas dentro de la pirámide formada dentro del detector, así como el paso que presenta la ventana deslizante encargada de obtener los Histogramas de Gradientes Orientados.



## 10.1. Transformación del sistema de coordenadas geodésicas al sistema NED

Para convertir las coordenadas geodésicas a coordenadas *NED*, primeramente se realiza el cambio de *WGS-84* a coordenadas *ENU* para posteriormente realizar la traducción de las mismas a coordenadas *NED*. Los pasos a seguir para el primer cambio son:

1.- Expresar las coordenadas geodésicas dentro del sistema *ECEF* de acuerdo a las siguientes ecuaciones

$$e^2 = 1 - \left(\frac{b}{a}\right)^2 \quad (10.1)$$

$$\chi = \sqrt{1 - e^2 \sin^2 \varphi} \quad (10.2)$$

$$\begin{aligned} x &= \left(\frac{a}{\chi} + h\right) \cos(\phi) \cos(\lambda) \\ y &= \left(\frac{a}{\chi} + h\right) \cos(\phi) \sin(\lambda) \\ z &= \left(\frac{a(1 - e^2)}{\chi} + h\right) \sin(\phi) \end{aligned} \quad (10.3)$$

en donde

- Semi-eje mayor  $\rightarrow R_{Ea} = 6,378,137$  m
- Semi-eje menor  $\rightarrow R_{Eb} = 6,356,752.3142$  m
- Primera excentricidad  $\rightarrow e = 0.08181919$
- $\varphi \rightarrow$  Latitud

- $\lambda \rightarrow$  Longitud
- $h \rightarrow$  Altura

2.- Convertir los diferenciales de las coordenadas geográficas al sistema *ECEF* por medio de la expansión de la serie de Taylor de las ecuaciones (10.1).

$$\begin{aligned}
 dx &= \left( -\frac{a \cos(\lambda) \sin(\varphi)(1-e^2)}{\chi^3} - h \cos(\lambda) \sin(\varphi) \right) d\varphi - \dots \\
 &\dots - \left( \frac{a \sin(\lambda) \cos(\varphi)}{\chi} + h \sin(\lambda) \cos(\varphi) \right) d\lambda + \cos(\varphi) \cos(\lambda) dh + \dots \\
 &\dots + \left( \frac{1}{4} a \cos(\varphi) \cos(\lambda) (-2 - 7e^2 + 9e^2 \cos^2(\varphi)) - \frac{1}{2} h \cos(\lambda) \cos(\varphi) \right) d\varphi^2 + \dots \\
 &\dots + \left( -\frac{a \sin(\lambda) \sin(\varphi)(1-e^2)}{\chi^2} + h \sin(\lambda) \sin(\varphi) \right) d\varphi d\lambda - \dots \\
 &\dots - \cos(\lambda) \sin(\varphi) dh d\varphi + \left( -\frac{a \cos(\lambda) \cos(\varphi)}{2\chi} - \frac{1}{2} h \cos(\lambda) \cos(\varphi) \right) d\lambda^2 - \dots \\
 &\dots - \sin(\lambda) \cos(\varphi) dh d\lambda + 0(d\varphi^3) + 0(dh d\varphi^2) \\
 dy &= \left( -\frac{a \sin(\lambda) \sin(\varphi)(1-e^2)}{\chi^3} - h \sin(\lambda) \sin(\varphi) \right) d\varphi - \dots \\
 &\dots - \left( \frac{a \cos(\lambda) \cos(\varphi)}{\chi} + h \cos(\lambda) \cos(\varphi) \right) d\lambda + \sin(\varphi) \cos(\lambda) dh + \dots \\
 &\dots + \left( \frac{1}{4} a \cos(\varphi) \sin(\lambda) (-2 - 7e^2 + 9e^2 \cos^2(\varphi)) - \frac{1}{2} h \sin(\lambda) \cos(\varphi) \right) d\varphi^2 + \dots \\
 &\dots + \left( -\frac{a \cos(\lambda) \sin(\varphi)(1-e^2)}{\chi^2} + h \cos(\lambda) \sin(\varphi) \right) d\varphi d\lambda - \dots \\
 &\dots - \sin(\lambda) \sin(\varphi) dh d\varphi + \left( -\frac{a \sin(\lambda) \cos(\varphi)}{2\chi} - \frac{1}{2} h \sin(\lambda) \cos(\varphi) \right) d\lambda^2 - \dots \\
 &\dots - \cos(\lambda) \cos(\varphi) dh d\lambda + 0(d\varphi^3) + 0(dh d\varphi^2) \\
 dz &= \left( \frac{a(1-e^2) \cos(\varphi)}{\chi^2} + h \cos(\varphi) \right) d\varphi + \sin(\varphi) dh + \cos(\varphi) dh d\varphi + \dots \\
 &\dots + \left( \frac{1}{4} a \sin(\varphi) (-2 - e^2 + 9e^2 \cos^2(\varphi)) - \frac{1}{2} h \sin(\varphi) \right) d\varphi^2 + 0(dh d\varphi^2)
 \end{aligned} \tag{10.4}$$

3.- Convertir de *ECEF* a *ENU* mediante la matriz de rotación

$$\begin{bmatrix} de \\ dn \\ du \end{bmatrix} = \begin{bmatrix} -\sin(\lambda) & \cos(\lambda) & 0 \\ -\sin(\varphi) \cos(\lambda) & -\sin(\varphi) \sin(\lambda) & \cos(\varphi) \\ \cos(\varphi) \cos(\lambda) & \cos(\varphi) \sin(\lambda) & \sin(\varphi) \end{bmatrix} \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} \tag{10.5}$$

4.- Sustituir 10.4 en 10.5

## 10.1. TRANSFORMACIÓN DEL SISTEMA DE COORDENADAS GEODÉSICAS AL SISTEMA NED

---

$$\begin{aligned}
 de &= \left( \frac{a}{\chi} + h \right) \cos(\varphi) d\lambda - \left( \frac{a(1-e^2)}{\chi^3} + h \right) \sin(\varphi) d\varphi d\lambda + \cos(\varphi) d\lambda dh \\
 dn &= \left( \frac{a(1-e^2)}{\chi^3} + h \right) d\varphi + \frac{3}{2} a \cos(\varphi) \sin(\varphi) e^2 d\varphi^2 + dh d\varphi + \dots \\
 &\dots + \frac{1}{2} \sin(\varphi) \cos(\varphi) \left( \frac{a}{\chi} + h \right) d\lambda^2 \\
 du &= dh - \frac{1}{2} a \left( 1 - \frac{3}{2} e^2 \cos(\varphi) + \frac{1}{2} e^2 + \frac{h}{a} \right) d\varphi^2 - \frac{1}{2} \left( a \cos^2 \left( \frac{\varphi}{\chi} \right) - h \cos^2(\varphi) \right) d\lambda^2
 \end{aligned} \tag{10.6}$$

5.- Por último, se realizan dos rotaciones sucesivas con respecto al sistema ENU, con un cierto ángulo sobre los ejes  $U$  y  $N$ , los cuales corresponden con los ejes  $Z$  e  $Y$  respectivamente. La rotación con respecto al eje  $U$  se realiza de  $90^\circ$ , para posteriormente realizar una rotación de  $-180^\circ$  con respecto al eje  $N$ .

$$[NED] = R_y(-180)R_z(90)[ENU] \tag{10.7}$$

$$\begin{bmatrix} X_N \\ Y_E \\ Z_D \end{bmatrix} = \begin{bmatrix} \cos(-180) & 0 & \sin(-180) \\ 0 & 1 & 0 \\ -\sin(180) & 0 & \cos(-180) \end{bmatrix} \begin{bmatrix} \cos(90) & -\sin(90) & 0 \\ \sin(90) & \cos(90) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_E \\ Y_N \\ Z_U \end{bmatrix} \tag{10.8}$$

$$\begin{bmatrix} X_N \\ Y_E \\ Z_D \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_E \\ Y_N \\ Z_U \end{bmatrix} \tag{10.9}$$

$$\begin{bmatrix} X_N \\ Y_E \\ Z_D \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} -Y_N \\ X_E \\ Z_U \end{bmatrix} \tag{10.10}$$

$$\begin{bmatrix} X_N \\ Y_E \\ Z_D \end{bmatrix} = \begin{bmatrix} Y_N \\ X_E \\ -Z_U \end{bmatrix} \tag{10.11}$$



## 11.1. Representación y Cálculo del CRC

El CRC puede ser representado y calculado de 3 distintas maneras, las cuales son:

1. Aritmética Módulo 2
2. Polinomial
3. Lógica Digital

Considerando el cálculo por medio de la aritmética de módulo dos, se tienen las siguientes definiciones que se utilizarán durante la explicación.

1.  $T \rightarrow$  Trama de  $(k + n)$  bits a transmitir, con  $n < k$
2.  $M \rightarrow$  Mensaje de  $k$  bits, los primeros bits de  $T$
3.  $F \rightarrow n$ -bits de la secuencia de comprobación de trama, los últimos bits de  $T$
4.  $P \rightarrow$  Patrón de  $n + 1$  bits, divisor elegido

Se tiene que

$$T = 2^n M + F \quad (11.1)$$

en donde el multiplicar  $M$  por  $2^n$  corresponde con desplazar hacia la izquierda  $n$  bits, agregando ceros al resultado. Al sumar  $F$  se están concatenando  $M$  y  $F$ . El objetivo principal de este método es hacer a  $T$  divisible entre  $P$ . Supóngase que se divide  $2^n M$  entre  $P$

$$\frac{2^n M}{P} = Q + \frac{R}{P} \quad (11.2)$$

Se obtiene un cociente y un resto, el cual será siempre al menos un bit más pequeño que el divisor. La secuencia de comprobación de la trama será igual al resto de la división, con lo que se tiene







## 12.1. Modelo de propagación de espacio libre

Un modelo de propagación se enfoca en estimar la potencia a través de una señal recibida a una cierta distancia del transmisor. así como las variaciones de la potencia de dicha señal en la cercanía de un punto de interés. Si una antena transmisora se encuentra posicionada en el espacio libre (alejada de la Tierra u alguna otra obstrucción), la misma posee una ganancia  $G_t$  en dirección a una antena receptora. La densidad de potencia a una cierta distancia  $d$  en la dirección elegida es

$$W = \frac{P_t G_t}{4\pi d^2} \quad (12.1)$$

La potencia en la antena receptora, que posee un área efectiva  $A$  viene dada por

$$\begin{aligned} P_r &= \frac{P_t G_t}{4\pi d^2} A \\ P_r &= \frac{P_t G_t G_r}{4\pi d^2} \left( \frac{\lambda^2 G_r}{4\pi} \right) \end{aligned} \quad (12.2)$$

con lo que se obtiene

$$P_r = P_t G_t G_r \left( \frac{\lambda}{4\pi d} \right)^2 \quad (12.3)$$

en donde

- $P_r$  → Potencia recibida en mW
- $P_t$  → Potencia transmitida en mW

- $G_r$  Ganancia de la antena receptora
- $G_t$  Ganancia de la antena transmisora
- $\lambda$  Longitud de onda en m
- $d$  Distancia, medida en metros, entre la antena receptora y la antena transmisora

Una manera alternativa de representar la ecuación de Friis consiste en considerar la relación  $c = f\lambda$ , obteniendo

$$\frac{P_r}{P_t} = G_t G_r \left( \frac{c}{4\pi f d} \right)^2 \quad (12.4)$$

# Bibliografía

---

- [1] D. P. Scharf, F. Y. Hadaegh, S. R. Ploen, "A Survey of Spacecraft Formation Flying Guidance and Control (Part II): Control" in *Proc. 2004 American Control Conf.*, Boston, United States, 2004, pp. 2976
- [2] J. A. Guerrero, R. Lozano, *Flight Formation Control*, Hoboken: Wiley, 2012, pp. 4, 7-9, 11-13
- [3] R. L. Boylestad, L. Nashelsky, "Transistores de Unión Bipolar", in *Electrónica: Teoría de Circuitos y Dispositivos Electrónicos*, 10ma ed., México, 2009, pp.131
- [4] K. Y. Pettersen, J. T. Gravdahl, H. Nijmeijer, "ISS-Based Robust Leader/Follower Trailing Control", in *Group Coordination and Cooperative Control*, Germany, 2006, pp. 1-2, 75
- [5] D. Lee, P. Y. Li, "Formation and Maneuver Control of Multiple Spacecraft" in *American Control Conf. Proc. 2003*, Denver, United States, pp. 278-283
- [6] The British Machine Vision Association and Society for Pattern Recognition, *What is computer vision?*, Available: <http://www.bmva.org/visionoverview>
- [7] R. Szeliski, "Stereo Correspondence", in *Computer Vision: Algorithms and Applications*, Washington, 2011, pp. 183-209, 469
- [8] S. Hrbar, "3D Path Planning and Stereo-based Obstacle Avoidance for Rotorcraft UAVs" in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Nice, France, 2008, pp. 807-809
- [9] R. Lozano, P. Castillo, A. Dzul, "The Quad-Rotor Rotorcraft", in *Modelling and Control of Mini-Flying Machines*, Springer, 2005, pp. 39-46
- [10] R. Lozano, "Two-Rotor VTOL Mini UAV:Design, Modeling and Control", in *Unmanned Aerial Vehicles - Embedded Control*, Hoboken: Wiley, 2010, pp. 50-51
- [11] G. Vianna, "Modelado y Control de un Helicóptero Quadrotor", M.C. Tesis, Univ. de Sevilla, Sevilla, 2007
- [12] Robot Operating System, *About ROS*, Available: <http://www.ros.org/about-ros/>
- [13] J. G. Proakis, D. G. Manolakis, *Tratamiento digital de señales*, 3era ed., Madrid, Spain, Prentice-Hall, 1998, pp. 2-18, 21-30
- [14] A. Papoulis, *Signal Analysis*, Singapur, McGraw-Hill, 1984, pp. 4

- 
- [15] K. Ogata, "Introducción a los Sistemas de Control en Tiempo Discreto", in *Sistemas de Control en Tiempo Discreto*, 2da ed., México, Prentice-Hall, 1996, pp. 3-5, 114-116
- [16] L. R. García, A. E. Dzul, R. Lozano, C. Pégard, "The Quad-Rotor Experimental Platform" in *Quad Rotorcraft Control*, London, England, Springer-Verlag, 2013, pp. 48-51
- [17] A. Anzueto, "Reconocimiento de escenas visuales empleando percepción difusa", M.S. thesis, Dept. Elect. Eng., CINVESTAV-IPN, Ciudad de México, Distrito Federal, 2003
- [18] M. E. Boudihir, M. Dufaut, R. Husson, "A vision system for mobile robot navigation", *Robotica*, vol. 12, no. 1, pp. 77-89, 1994
- [19] P. Lee, A. S. Timmaraju, "Learning Binary Descriptors from Images", Dept. Elec. Eng., Stanford Univ.
- [20] S. Leutenegger, M. Chli, R. Y. Siegwart, "BRISK: Binary Robust Invariant Scalable Keypoints", in *IEEE Int. Conf. on Computer Vision*, 2011, 2548
- [21] D. A. Forsyth, J. Ponce, "Linear Filters", in *Computer Vision: A Modern Approach*, 1era ed., pp. 204-211
- [22] C. M. Bishop, *Pattern Recognition and Machine Learning*, New York,: Springer, 2006, 1, 2, 179
- [23] N. Dalal, B. Triggs, "Histograms of Oriented Gradients for Human Detection" in *Proc. 2005 IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, San Diego, CA, USA, 2005
- [24] G. Cai, B. M. Chen, T. H. Lee, *Unmanned Rotorcraft Systems*, 1era ed., London, England, Springer, 2011, pp. 23-34, 223-227
- [25] C. Godsil, G. Royle, "Graphs", in *Algebraic Graph Theory*, 1era edición, New York, 2001, pp. 1-9
- [26] W. Ren, R. W. Beard, E. M. Atkins, "Information Consensus in Multivehicle Cooperative Control" in *IEEE Control. Syst. Mag.*, vol. 27, no. 2, pp. 71, 72, Apr. 2007
- [27] J. D. Day, H. Zimmermann, "Fundamentals of the Layer Service Definitions and Protocol Specifications" in *Proc. IEEE*, 1983, pp. 1341-1345
- [28] J. D. Day, H. Zimmermann, "The OSI Reference Model" in *Proc. IEEE*, 1983, pp. 1334- 1340
- [29] R. Rom, M. Sidi, *Multiple Access Protocols: Performance and Analysis*, 1era ed., New York: Springer-Verlag, 1990, pp. 1-7, 47, 79
- [30] G. A. King, *Understanding and Designing Computer Networks*, 1era. ed., Bristol, England: Edward Arnold, 1995, pp. 2,3, 44-52
- [31] A. S. Tanenbaum, "The Medium Access Control Sublayer" in *Computer Networks*, 4ta ed., Prentice Hall, 2003, pp. 222,223
- [32] W. Stallings, "Control del Enlace de Datos" in *Comunicaciones y Redes de Computadores*, 6ta ed., Prentice Hall, 2001, pp. 181-184, 188-200