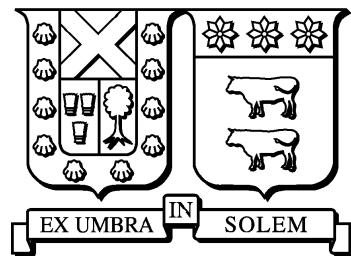


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

DEPARTAMENTO DE INFORMÁTICA  
SANTIAGO – CHILE



“MEJORA DE UN SISTEMA DE PREVENCIÓN DE  
COLISIONES DE VEHICULO AÉREO NO  
TRIPULADO EN ESPACIOS CERRADOS”

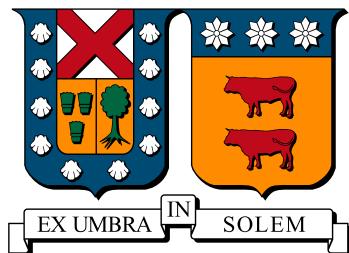
CELESTE FERNANDA BERTIN AGUILAR

MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE  
INGENIERO CIVIL INFORMÁTICO

PROFESOR GUÍA: LIOUBOV DOMBROVSKAIA

ENERO 2018

**UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA  
DEPARTAMENTO DE INFORMÁTICA  
SANTIAGO – CHILE**



**“MEJORA DE UN SISTEMA DE PREVENCIÓN  
DE COLISIONES DE VEHICULO AÉREO NO  
TRIPULADO EN ESPACIOS CERRADOS”**

**CELESTE FERNANDA BERTIN AGUILAR**

**MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE  
INGENIERO CIVIL INFORMÁTICO**

**PROFESOR GUÍA: LIOUBOV DOMBROVSKAIA**

**PROFESOR CORREFERENTE: JOCELYN SIMMONDS**

**ENERO 2018**

**MATERIAL DE REFERENCIA, SU USO NO INVOLUCRA RESPONSABILIDAD DEL AUTOR O DE LA INSTITUCIÓN**

*Dedicado*

*A quienes desarmaron sus juguetes  
y a los padres que lo permitieron.*

# **Agradecimientos**

A las profesoras Lioubov Dombrovskiaia y Jocelyn Simmonds, su orientación, mentoría e infinita paciencia hicieron este trabajo posible, gracias por apoyarme con este tema de memoria tan inusual;

A mi mis padres y hermano, que siempre me apoyaron y por permitirme transformar la casa en una cancha de pruebas para esta memoria, proporcionado el espacio necesario para realizar todas las pruebas que necesitase;

A mis compañeros con los que hemos compartido, jugado, sufrido y crecido durante la carrera;

Y finalmente a Sebastián, quien me ha acompañado en todo este proceso, tu apoyo a toda prueba

Además quiero agradecer a todos quienes me ayudaron a terminar la carrera: profesores y amigos sin los cuales nada de esto sería posible. ¡Muchas gracias!

# Resumen

En este trabajo, un sistema de detección y evasión de colisiones en dos dimensiones, desarrollado por terceros, se extiende para cubrir la tercera dimensión. El sistema es implementado en un vehículo aéreo no tripulado de cuatro hélices, armado en base a un kit DJI y un controlador de vuelo Pixhawk, provisto de sensores ultrasónicos para medición de distancia, conectados a un microcontrolador Arduino. La detección de obstáculos y la ejecución del algoritmo de evasiones es desarrollada en tiempo real. Las mediciones de distancia se limpian mediante filtro de mediana y filtro lineal de Kalman; el microcontrolador toma el control del UAV si detecta un obstáculo en su zona de seguridad, evita la colisión y devuelve el control al piloto. El sistema implementado es capaz de detectar y evadir en tiempo real obstáculos en tres dimensiones, mejorando la seguridad de un UAV que circula en un espacio reducido. La velocidad máxima del UAV que asegura evasión satisfactoria es reducida con respecto al trabajo base, sin llegar a ser demasiado lenta para su uso en espacios reducidos.

# Abstract

This work presents the implementation of a collision detection and avoidance system in three dimensions. It is based on the work of Rethnaraj Rambabu et al. that creates a collision avoidance system in two dimensions, extending it to cover a third dimension. The algorithm is implemented with a quadrotor unmanned aerial vehicle, or UAV, assembled from a DJI kit and a Pixhawk flight controller; the UAV is provided with ultrasonic sensors for distance measurement, connected to an Arduino microcontroller for detecting obstacles and executing the collision avoidance algorithms in real time. The distance measurements are filtered by a median filter and a Kalman linear filter; the microcontroller takes control of the UAV if it detects an obstacle on its safe zone, returning control to the user after avoiding the obstacle. The implementation is tested against the Rambabu et al. implementation, where it is determined that the new system successfully allows real-time collision detection and avoidance in three dimensions, improving the safety of an UAV moving in an enclosed space. The maximum UAV speed that allows successful collision avoidance is reduced, while still being acceptable speeds for usage in enclosed spaces.

# Índice de Contenidos

<b>Lista de Figuras</b>	<b>x</b>
<b>Introducción</b>	<b>1</b>
<b>1 Definición del Problema</b>	<b>3</b>
1.1 Descripción . . . . .	3
1.2 Objetivos . . . . .	4
1.3 Solución Esperada . . . . .	4
1.4 Metodología . . . . .	5
<b>2 Estado del Arte</b>	<b>6</b>
2.1 Técnicas para detectar colisiones . . . . .	7
2.1.1 Espacios en dos dimensiones . . . . .	7
2.1.2 Espacios en tres dimensiones . . . . .	8
2.2 Técnicas para evitar colisiones . . . . .	8
2.3 Comparación de hardware . . . . .	9
2.3.1 Parrot AR.Drone 2.0 . . . . .	9
2.3.2 3DR Pixhawk . . . . .	10
2.3.3 Discusión . . . . .	11

<b>3 Diseño de la Solución</b>	<b>12</b>
3.1 Medición de Distancias . . . . .	12
3.2 Detección de Colisiones . . . . .	15
3.3 Evasión de Colisiones . . . . .	16
3.4 Arquitectura . . . . .	17
3.5 Especificaciones Técnicas . . . . .	18
3.5.1 UAV . . . . .	18
3.5.2 Sensores y Microcontrolador . . . . .	19
3.6 Resumen . . . . .	20
<b>4 Implementación</b>	<b>22</b>
4.1 Medición de Distancias . . . . .	22
4.1.1 Encadenamiento de Sensores . . . . .	24
4.1.2 Sensores Infrarrojos . . . . .	24
4.1.3 Limpieza de mediciones . . . . .	25
4.2 Detección de Colisiones . . . . .	28
4.3 Evasión de Colisiones . . . . .	28
4.3.1 Patrones de Evasión . . . . .	31
4.4 Monitoreo . . . . .	33
4.5 Discusión . . . . .	35
<b>5 Evaluación</b>	<b>37</b>
5.1 Preparación de las pruebas . . . . .	38
5.1.1 Revisión del entorno de pruebas . . . . .	38
5.2 Pruebas . . . . .	39
5.2.1 Obstáculo estático . . . . .	40

<b>6 Resultados</b>	<b>43</b>
6.1 Formato . . . . .	43
6.2 Una Dimensión . . . . .	44
6.3 Dos Dimensiones . . . . .	47
6.4 Tres Dimensiones . . . . .	52
6.5 Análisis de Resultados . . . . .	56
6.5.1 Comparación con Algoritmo en 2 Dimensiones . . . . .	56
<b>Conclusiones</b>	<b>57</b>
<b>Trabajo futuro</b>	<b>59</b>
<b>Bibliografía</b>	<b>61</b>
<b>A Diagramas</b>	<b>64</b>

# Índice de figuras

3.1	Posicionamiento de sensores en el UAV: (a) sensores horizontales y (b) sensores verticales. . . . .	13
3.2	Ciclo del filtro lineal de Kalman [22]. . . . .	15
3.3	Diagrama de bloques del controlador PID [32]. . . . .	16
3.4	Sistema de prevención de colisiones. . . . .	17
4.1	Vista superior del UAV . . . . .	23
4.2	Voltaje sin sensor infrarrojo conectado (a) y voltaje luego de conectarlo un sensor infrarrojo (b). . . . .	25
4.3	Mediciones de distancia sin procesar y luego de ser filtradas, UAV en reposo	26
4.4	Mediciones de distancia sin procesar y luego de ser filtradas, UAV en movimiento . . . . .	27
4.5	Rangos de valores para cada canal del UAV [6]. . . . .	30
4.6	Transformación de señales entre PPM y PWM [27]. . . . .	31
4.7	Ejemplo visualización mediciones enviadas desde Arduino a computador a través de antena XBee a un rango de 10 metros [39]. . . . .	34
5.1	Evaluación frontal de detección y evasión de colisiones . . . . .	40

5.2	Campo de pruebas en espacio cerrado.	41
6.1	Detección y evasión frontal, en una dimensión	45
6.2	Detección y evasión superior, en una dimensión	46
6.3	Detección y evasión frontal, en dos dimensiones	48
6.4	Detección y evasión lateral, en dos dimensiones	49
6.5	Detección y evasión lateral, en dos dimensiones	50
6.6	Detección y evasión superior en dos dimensiones	51
6.7	Detección y evasión frontal, en tres dimensiones	53
6.8	Detección y evasión lateral, en tres dimensiones	54
6.9	Detección y evasión superior, en tres dimensiones	55
A.1	Configuración del hardware a alto nivel.	64
A.2	Encadenamiento de sensores ultrasónicos.	65
A.3	Conexión analógica de sensor infrarrojo a Arduino.	65

# Glosario

- Attitude: En aeronáutica refiere a la inclinación de los tres ejes principales de una aeronave.
- CAC: Collision Avoidance Controller, conjunto de algoritmos que procesan información de sensores para la detección de colisiones, y manipulan comandos de control de vuelo para ejecutar evasión de las mismas.
- Controlador de vuelo: Controlador del UAV, el cual contiene sensores para la estabilización y control del UAV, además de controlar los motores.
- Pitch: Movimiento hacia adelante o hacia atrás de un UAV, realizado inclinando el UAV hacia adelante o hacia atrás.
- PPM: Pulse-position modulation, protocolo de comunicación a través del cual múltiples señales son transmitidas por un solo canal de comunicación.
- PWM: Pulse-width modulation, protocolo de comunicación a través del cual una señal es transmitida como un pulso periódico para la transmisión de información a través de un canal de comunicaciones.
- Roll: Movimiento lateral de un UAV, realizado inclinando el UAV hacia la izquierda o derecha.
- Throttle: Control del movimiento de subida o bajada del UAV.
- UAV: Unmanned Aerial Vehicle (Vehículo Aéreo No Tripulado, por sus siglas en inglés), también conocido como drone, es una aeronave que vuela sin tripulación a bordo.

# Introducción

La robótica ha crecido de forma rápida estos últimos años. Para el funcionamiento seguro de un robot, este debe ser capaz de detectar obstáculos y alterar su recorrido de tal forma que evite colisiones; esto hace que los algoritmos de detección y evasión de colisiones sean áreas de investigación en constante progreso.

Para los vehículos aéreos no tripulados (en adelante UAV por sus siglas en inglés), esto es un problema más complejo que para otros robots, ya que estos funcionan en un ambiente tridimensional. Es crucial que la detección y evasión de colisiones se realice en tiempo real, ya que cualquier demora tiene como resultado la colisión y posible destrucción del UAV, interrumpiendo su labor. Actualmente, para la detección y evasión de colisiones se utilizan algoritmos que funcionan en el plano horizontal, asumiendo que el UAV será utilizado exclusivamente en exteriores, por lo cual la altura no se considera relevante. El aumento del uso en interiores de estos robots hace imperativa la investigación de nuevos algoritmos que tomen en cuenta estas nuevas limitaciones.

En este trabajo, se realiza una extensión al trabajo realizado por Rambabu et al. [36], en el cual se desarrolla un sistema de detección y evasión de colisiones para UAVs en el plano horizontal, utilizando sensores infrarrojos y ultrasónicos para medir distancias, detectar colisiones y realizar maniobras de evasión en tiempo real. Al cubrir las cuatro direcciones del plano horizontal de manera independiente, es posible extender este sistema de detección y evasión de colisiones al agregar el eje vertical, con tal de detectar y evitar colisiones de manera tridimensional.

Se demuestra en la memoria que realizar esta extensión es suficiente para evitar colisiones

en los tres ejes en tiempo real.

La estructura de la presente memoria es la siguiente:

- El capítulo 1 presenta el sistema. Se explica cuál es el sistema, los objetivos a cumplir en este trabajo, y la metodología de trabajo para llegar a cumplir tales objetivos.
- El capítulo 2 muestra la investigación realizada para estudiar las técnicas de detección y evasión de colisiones disponibles en la literatura, y el hardware que se puede utilizar para implementar dichas técnicas.
- El capítulo 3 explica el diseño del sistema: desde la lectura e interpretación de los sensores de distancia, pasando por los algoritmos de detección de colisiones, hasta la intervención de los comandos enviados al UAV para evasión de colisiones; junto con un resumen del hardware usado para ensamblar el UAV.
- El capítulo 4 especifica la forma en que se implementó el diseño mencionado anteriormente: los sensores efectivamente utilizados, detalles sobre los umbrales de detección de colisiones, y la forma en que se monitorea el estado del UAV mientras se lleva a cabo la ejecución de los distintos algoritmos de evasión.
- El capítulo 5 detalla la metodología con la que se desarrollará la evaluación del sistema en comparación a la literatura. Se describen las distintas pruebas con las que se evaluará el sistema, y las condiciones en las cuales estas pruebas deben ser ejecutadas.
- El capítulo 6 realiza la comparación del sistema, tomando los resultados obtenidos en el capítulo anterior y discutiendo las diferencias en rendimiento y eficiacia del sistema, con respecto a la literatura.
- Finalmente, en la conclusión se resume el trabajo realizado y se determina que los objetivos de la presente memoria hayan sido cumplidos.

# **Capítulo 1**

## **Definición del Problema**

### **1.1 Descripción**

Se define un vehículo aéreo no tripulado, también conocido como drone, o UAV, por sus siglas en inglés, a un aparato volador pequeño, impulsado por cuatro o más hélices, generalmente de uso civil, el cual se controla mediante control remoto o navega de manera independiente, guiándose bajo parámetros establecidos por el usuario.

Estos están pensados principalmente para su uso en exteriores, pero su uso en interiores ha ido en aumento por su gran variedad de usos, desde mantenimiento de inventario en bodegas [28] hasta vigilancia de propiedades [9]. Su uso en espacios cerrados significa un gran riesgo de colisiones por las limitaciones de espacio, lo que pueden significar la destrucción del mismo. Últimamente, los UAVs de gama alta que están en el mercado han agregado detección y evasión de obstáculos, pero estos son para vuelo autónomo del UAV y pensado solo para uso en exteriores.

## **1.2 Objetivos**

El objetivo principal de este trabajo es mejorar un algoritmo existente para evitar colisiones de UAVs en espacios cerrados, específicamente colisiones causadas al utilizar el UAV en modo manual en espacios cerrados. De forma secundaria, para lograr dicho objetivo se debe poder detectar colisiones a través de la detección de obstáculos, y al detectarlos, tomar control del UAV y ejecutar maniobras para evitar colisiones, para luego devolver el control manual al piloto. El UAV debe ser capaz de calcular rápidamente la distancia entre sí mismo y los obstáculos que encuentre en cualquiera de las seis direcciones en las que puede moverse, labor que debe ejecutar de forma independiente, y obteniendo resultados suficientemente precisos de forma rápida, con tal de ejecutar maniobras de evasión en tiempo real.

## **1.3 Solución Esperada**

La idea de realizar este trabajo parte de las investigaciones realizadas por Rambabu et al. [36], sobre un sistema de detección y evasión de colisiones utilizando fusión de sensores ultrasónicos e infrarrojos. Dicha investigación trabaja en dos dimensiones, asumiendo que la altura del UAV se mantiene constante. Aunque dicha investigación tenía como objetivo evadir otros UAVs en una formación de UAVs, no lo lograron por ser muy pequeño el obstáculo a evadir, por lo cual su implementación fué evaluada exitosamente utilizando superficies de mayor tamaño como obstáculos y realizando las pruebas en espacios cerrados.

Se desea replicar este trabajo ya que funcionalmente es un sistema de detección y evasión de colisiones en interiores. Luego se añadirá detección de obstáculos de manera vertical, para finalmente evadir obstáculos en tres dimensiones.

Al terminar este trabajo, se espera que el UAV pueda medir, detectar y evadir obstáculos en las seis direcciones de manera suficientemente precisa y en tiempo real. Esto se realizará en hasta tres direcciones perpendiculares simultáneamente. Para esto, se generarán patrones de evasión para las distintas colisiones detectadas.

## **1.4 Metodología**

La ejecución de esta memoria tiene tres fases:

La primera es investigar y construir el UAV, definiendo los sensores necesarios para la medición de distancias y determinando el hardware para procesar el algoritmo de detección y evasión de obstáculos a bordo del UAV.

Posteriormente, se desarrolla un algoritmo de medición de distancia lo suficientemente preciso, el cual será ejecutado en el UAV.

Luego de poder medir distancias, se prueban umbrales de detección de colisiones, para luego programar los distintos patrones de evasiones de colisiones. Los patrones de evasión son los movimientos que realiza el UAV para evitar colisiones, los cuales dependen de las colisiones detectadas.

# **Capítulo 2**

## **Estado del Arte**

La investigación bibliográfica realizada tiene como objetivo ser una colección de artículos que puedan apoyar el desarrollo de la memoria escogida. Para esto, se centró la investigación en:

- Determinar técnicas para detectar colisiones
- Determinar técnicas para evitar colisiones
- Definir el hardware a utilizar

Los trabajos estudiados permiten concluir que para la detección de obstáculos, la detección en dos dimensiones se puede realizar en tiempo real a través de sensores que miden distancias en tiempo real, mientras que en tres dimensiones se detectan colisiones a través del procesamiento de imágenes capturadas por cámaras en el UAV y procesadas externamente. Para realizar detección de colisiones y evitar colisiones, se propone expandir el trabajo de Rambabu et al. [36], en el cual se utilizan tanto sensores infrarrojos como ultrasónicos, fusionando los sensores para medir distancias horizontalmente con mayor exactitud. Este se puede expandir agregando pares de sensores verticales para estimar distancias de altura al suelo y distancia al techo, utilizando un total de 6 pares de sensores, en lugar de 4 pares, lo que permitiría medir distancias en los tres ejes coordinados. De las técnicas para evitar

colisiones, la técnica de campo de repulsión es la más liviana computacionalmente, por lo cual puede ser ejecutada en tiempo real, y es más eficiente que la técnica geométrica. Cabe notar que en el trabajo de Rambabu et al. [36] se utilizó la técnica de campo de repulsión para evitar colisiones.

## 2.1 Técnicas para detectar colisiones

En la literatura se encuentran variadas técnicas para detectar obstáculos, y por ende, detectar colisiones. Estas técnicas dependen de sensores montados en el UAV, cuya información puede ser procesada en el mismo vehículo, o a distancia, en un computador externo. Estas técnicas se pueden separar en su implementación, para espacios en dos dimensiones o en tres dimensiones.

### 2.1.1 Espacios en dos dimensiones

Para la detección de obstáculos en dos dimensiones, se utilizan sensores orientados horizontalmente, con el objetivo de medir distancias y detectar si el UAV está próximo a un obstáculo. Los sensores más utilizados para este fin son:

- Infrarrojo, el cual es útil para medición de cortas distancias hasta 1,5 metros [16]. Estas mediciones son realizadas en linea recta a un punto.
- Ultrasonido, el cual mide distancias hasta de 6,45 metros, pero la distancia mínima que puede medir es de 0,2 metros [31]. Estas mediciones son realizadas en un cono [10] por lo cual este sensor entrega la distancia más cercana dentro del cono.

Por las limitaciones de estos sensores, Rambabu et al. proponen en [36] la fusión de ambos tipos de sensores para obtener una medición de mayor confianza. Este método solo ha sido probado en dos dimensiones en espacios cerrados y asumiendo una altura constante.

## **2.1.2 Espacios en tres dimensiones**

Para la detección de obstáculos en tres dimensiones, se utilizan cámaras monoculares o cámaras en estéreo [21]. Se utilizan las imágenes obtenidas por las cámaras para la determinación de distancias y detección de obstáculos. Los algoritmos de procesamiento de imágenes y detección que utilizan son exigentes computacionalmente, por lo cual deben ser procesados en un computador externo al UAV. El uso de cámaras genera una gran cantidad de información, la cual debe pasar por una fase de procesamiento que puede ser muy lenta [31]. Aun siendo estas procesadas externamente, el tiempo del proceso va desde algunos segundos hasta minutos dependiendo del algoritmo, por lo cual la detección de colisiones no se realiza en tiempo real.

## **2.2 Técnicas para evitar colisiones**

Para evitar colisiones, las técnicas principales encontradas en la literatura son la técnica geométrica, técnica de campo de repulsión y técnica de replanificación de ruta [3].

- Geométrica: utiliza trigonometría para evitar colisiones, asumiendo que el obstáculo es puntual y viaja a velocidad constante [33].
- Campo de repulsión: utilizada ampliamente en robótica para evitar colisiones en dos dimensiones, desde que fue propuesta por Khatib [25]. Esta técnica define una área segura alrededor de un robot en movimiento a través de la detección de distancias a su alrededor, y si detecta algún obstáculo en su área segura, el robot se aleja de este a una velocidad proporcional a la velocidad de acercamiento al obstáculo.
- Replanificación de ruta: se ejecuta a partir de una discretización del ambiente generada con anterioridad y representada como una grilla. Se utiliza esta discretización para calcular un camino libre de obstáculos, y si se detecta alguno, se calcula una nueva ruta utilizando algoritmos de búsqueda en grafos [20].

De las técnicas para evitar colisiones, la geométrica es liviana computacionalmente, pero no

considera que un obstáculo puede ser dinámico; mientras que el algoritmo de replanificación es el más exigente computacionalmente, siendo este imposible de utilizar en tiempo real. La técnica de campo de repulsión es la más liviana computacionalmente, ya que sólo depende de la cantidad de obstáculos en su área segura, pudiendo ésta ser utilizada en tiempo real.

## 2.3 Comparación de hardware

En el mercado existe una gran cantidad de UAVs, los cuales pueden ser tanto kits completos como UAVs armados a partir de componentes individuales. Para este trabajo se hace énfasis en la capacidad de expansión de hardware del procesador de vuelo de cada UAV, ya que es necesario montar sensores adicionales en los UAV para la realización de este trabajo. A continuación se estudian los procesadores de vuelo más utilizados en investigación.

### 2.3.1 Parrot AR.Drone 2.0

Este UAV es usado principalmente en investigaciones que requieren capturar datos desde una ubicación aérea. Esto, debido a que posee cámaras incorporadas, WiFi, y la capacidad de enviar datos en tiempo real a un computador externo; dicho computador puede también controlar el UAV mediante un SDK para desarrolladores. Además, como viene armado de fábrica, lo hace una opción ideal para investigadores, sin necesidad de conocimiento previo de electrónica [26].

Sus usos están relacionados con navegación por procesamiento de imágenes. El caso de uso típico es el de navegación por sensores basados en visión: el UAV capture imágenes mediante dos cámaras, las cuales apuntan hacia adelante y hacia el suelo. Las imágenes son enviadas a un computador externo que las procesa e interpreta [13].

### **2.3.2 3DR Pixhawk**

El 3DR Pixhawk de 3DR es un controlador de vuelo independiente. Puede configurarse para manejar cualquier clase de UAV, ya sea un avión, un drone multi-rotor, o incluso vehículos terrestres [5]. Tiene interfaces de salida PWM/Servo para controlar los motores del UAV, y tiene interfaces de comunicación tipo i<sup>2</sup>c, UART y CAN [2] para conectar toda clase de periféricos.

Adicionalmente, el firmware del Pixhawk es completamente open source. Esto lo hace completamente modificable, existiendo múltiples ejemplos de parte de la comunidad para darle a este controlador diversos usos para el control de cualquier vehículo.

Ahora bien, debido a que el Pixhawk es sólo un controlador de vuelo, requiere el armado de todo el resto del UAV, con el conocimiento electrónico que esto implica. Para simplificar la elección de hardware de este trabajo, se decidió evaluar el conjunto de partes que componen el UAV utilizado por Rambabu et al. [36] en comparación al AR.Drone.

Parte de los periféricos que se pueden utilizar en conjunto con el Pixhawk son, entre otros:

- sensores de movimiento
- medidores de distancia
- interfaces serial Xbee
- interfaces de conexión Bluetooth
- coprocesadores de vuelo, como Arduino
- cámaras de fotos y video

La capacidad de utilizar sensores y coprocesadores en conjunto con el Pixhawk permiten darle usos similares al del AR.Drone, como navegación y procesamiento de imágenes, sin necesidad de utilizar un computador externo; lo cual no impide utilizar procesamiento y control externos mediante las distintas interfaces de comunicación disponibles.

### **2.3.3 Discusión**

Una desventaja importante del AR.Drone es que no posee funcionalidades para detectar obstáculos en tres dimensiones. Además, el hardware por defecto del UAV no es expansible, salvo por unos pocos sensores oficiales [34] y unos cuantos no oficiales [38], que no son compatibles con los alcances de este trabajo.

Debido a esta limitante, se descarta el AR.Drone como opción de forma inmediata, pues no permite desarrollar el objetivo de la memoria.

En comparación, armar un UAV con el Pixhawk como controlador de vuelo, permite conectar suficientes sensores para realizar mediciones en tres dimensiones; estos sensores no necesitan ser cámaras, con lo cual se pueden hacer cálculos más sencillos para obtener mediciones de obstáculos para poder esquivarlos, que es el objetivo de este trabajo.

Además, existe una gran comunidad en línea que discute como usar y modificar el Pixhawk [1], donde se responden preguntas rápidamente y se publica código.

En resumen, la amplia capacidad de expansión y configuración del Pixhawk compensan con creces la dificultad que supone tener que replicar la configuración de hardware utilizada por Rambabu et al. [36]. Se puede asumir con confianza que el hardware escogido en dicho trabajo es suficiente para realizar la mejora del algoritmo que planea en esta memoria.

# **Capítulo 3**

## **Diseño de la Solución**

La solución consiste en que los comandos de control remoto son procesados por el CAC, y en los casos que lo estime conveniente, los reemplaza por los comandos adecuados si es necesario evitar una colisión. Para lograr los objetivos detallados en el capítulo 1 se genera un plan de trabajo que consta de tres etapas:

- Medición de distancias
- Detección de colisiones
- Evasión de colisiones

En el presente capítulo se describe el diseño propuesto para cada una de estas etapas.

### **3.1 Medición de Distancias**

Las acciones que realizará el UAV se basarán en mediciones de la distancia relativa ( $x_r, y_r, z_r$ ) entre el UAV y objetos en el ambiente, por lo cual la medición de dichas distancias debe ser lo más precisa posible. Para lograr esto, se utilizan dos sensores de manera simultánea para la medición de distancias, uno infrarrojo y uno ultrasónico. Como se abordará más

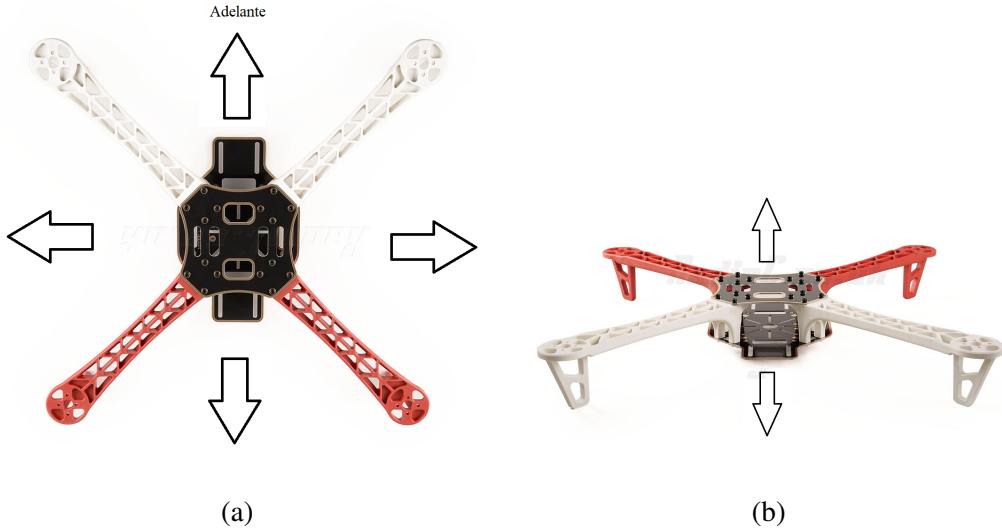


Figura 3.1: Posicionamiento de sensores en el UAV: (a) sensores horizontales y (b) sensores verticales.

adelante en este mismo capítulo, estos tienen distintos rangos de medición de distancias, y son susceptibles a interferencia en sus mediciones de maneras distintas, por lo cual realizar mediciones con ambos sensores y unir dichas mediciones es una forma de obtener distancias precisas y más resistentes a interferencias.

Para la medición de distancias en tres dimensiones, se ubica un par de sensores infrarrojo - ultrasónico realizando mediciones en las tres dimensiones del eje de coordenadas.

Se implementa un algoritmo de unión de mediciones (también conocido como *sensor fusion*) para obtener una distancia a partir de ambas mediciones. Para este caso específico, se utiliza el filtro lineal de Kalman [35].

Las mediciones realizadas por ambos sensores tienen cierto grado de incertidumbre, tanto por su rango de funcionamiento como por la interferencias del ambiente que generan ruido en las mediciones. Para unir estas mediciones, primero se determina los rangos en que un sensor es más confiable que el otro, en este caso el sensor infrarrojo es más confiable para distancias menores, mientras que el sensor ultrasónico es más confiable para distancias de

mayor magnitud.

$$\xi_{US} = \begin{cases} 0 & \text{si } r_s < r_{min} \\ 1 & \text{si } r_s > r_{max} \\ \frac{r_s - r_{min}}{r_{max} - r_{min}} & \text{en otro caso} \end{cases} \quad (3.1)$$

$$\xi_{IR} = 1 - \xi_{US} \quad (3.2)$$

Siendo  $\xi_{US}$  y  $\xi_{IR}$  la confiabilidad del sensor ultrasónico e infrarrojo, respectivamente. Los valores  $r_{min}$  y  $r_{max}$  representan el rango en que se ponderan las mediciones del sensor ultrasónico e infrarrojo [14]. En este caso  $r_{min} = 20$  centímetros, correspondiente a la distancia bajo la cual el sensor ultrasónico no es confiable, y  $r_{max} = 70$  centímetros, correspondiente a la distancia sobre la cual el sensor infrarrojo no es confiable.

Utilizando la ecuación 3.1, se le da relevancia a los valores obtenidos por cada sensor, para luego introducir cada uno al filtro lineal de Kalman. Este filtro considera un conjunto de muestras tomadas desde los sensores, junto con las distintas imperfecciones que se pueden encontrar en las mediciones de los sensores, para producir una estimación más precisa de la distancia relativa del UAV a los obstáculos detectados.

Al aplicar el filtro sobre el conjunto de mediciones, se realiza una predicción basada en el supuesto de que las mediciones tienen un error distribuido según la distribución gaussiana. El algoritmo recibe como parámetros el ruido del proceso  $Q$ , ruido de medición  $R$  y covarianza  $P$ ; el resultado del algoritmo se basa en la medición anterior, la medición actual y los parámetros [19].

Como la estimación se basa en las mediciones actuales y en las mediciones tomadas con anterioridad, el resultado es una estimación precisa de la distancia real del UAV, según la confiabilidad de cada sensor, dada una cierta distancia.

La figura 3.2 muestra como funciona el filtro lineal de Kalman, utilizando información del estado anterior del sistema más los parámetros de entrada, para generar una predicción de la distancia. Luego introduce la medición al filtro, para luego corregir la estimación de la distancia. Esa corrección es utilizada actualizar la predicción del siguiente ciclo, realizando una estimación del estado actual del sistema.

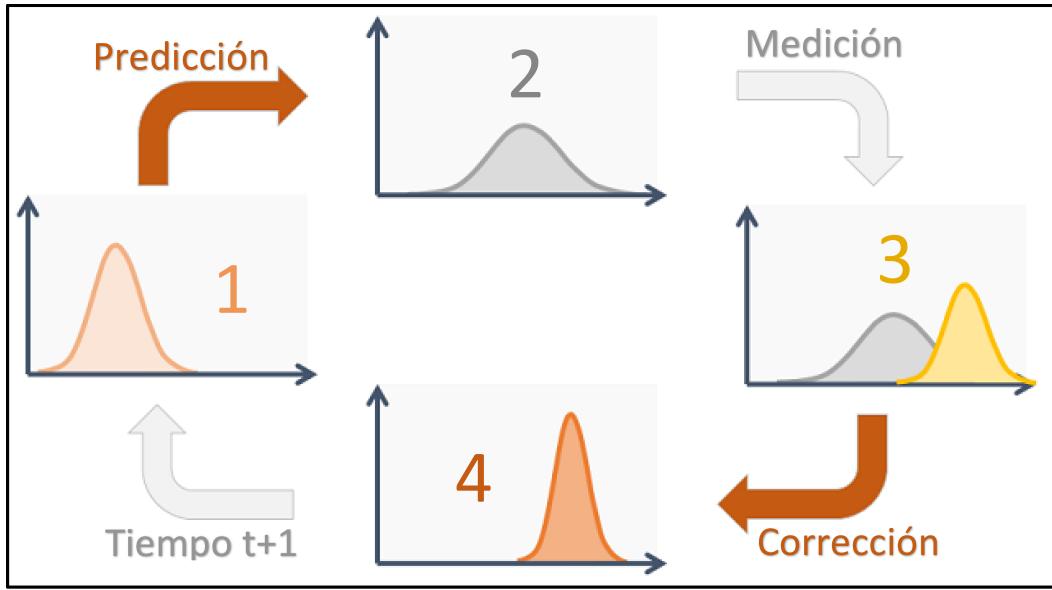


Figura 3.2: Ciclo del filtro lineal de Kalman [22].

## 3.2 Detección de Colisiones

Para realizar la detección de colisiones, los sensores miden constantemente las distancias en cada una de las seis direcciones. Si cualquier par de sensores mide una distancia inferior a la distancia de seguridad preestablecida ( $x_d, y_d, z_d$ ), el sistema detecta una colisión en dicha dirección y se asignan los valores de error ( $x_e, y_e, z_e$ ), para su utilización en la evasión de colisiones. Los valores de error se definen como:  $(x_e, y_e, z_e) = (x_d - x_r, y_d - y_r, z_d - z_r)$ .

Al funcionar en un espacio cerrado simple, se trabaja con hasta tres colisiones perpendiculares simultáneas, ya que más colisiones significa que el UAV se encuentra rodeado y no puede evadir los obstáculos, por ende esos casos no serán estudiados. Al tener seis direcciones independientes y hasta tres colisiones perpendiculares simultáneas, se obtienen distintos patrones de colisión. Estos patrones se invalidan en el caso de detectarse más de tres colisiones simultáneamente, o si se detectan colisiones en direcciones opuestas. Estos patrones son análogos a las caras, aristas y esquinas de un prisma rectangular en los casos de una, dos y tres dimensiones respectivamente. La combinatoria de estas dan un total de 26 patrones de colisión distintas.

### 3.3 Evasión de Colisiones

Ya detectada la colisión, el UAV debe modificar su trayectoria para evitar dicho obstáculo. Esto debe realizarse en tiempo real, con tal de cuidar la integridad del UAV y evitar interrupciones en su normal funcionamiento. Esto se realiza a través la interrupción de los comandos del piloto, enviando comandos calculados por el CAC en base a la diferencia de la distancia actual y la distancia mínima de seguridad predeterminada. Cuando el UAV se encuentre lejos de todo obstáculo, se le devuelve el control del UAV al piloto. Al tener seis direcciones independientes y hasta tres colisiones perpendiculares simultáneas, se generan patrones de evasión para los 26 casos mencionados anteriormente. Para evadir los obstáculos, se utiliza la técnica de campo de repulsión descrita en el capítulo 2. Para controlar los comandos de evasión enviados al UAV, se utiliza un controlador PID (Proporcional Integral Derivativo), con el cual se controla con cuanta intensidad debe alejarse el UAV de un obstáculo.

$$MV(t) = K_p e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt} \quad (3.3)$$

Siendo  $K_p, K_i, K_d$  constantes positivas correspondientes a la ganancia proporcional, ganancia integral y ganancia derivativa respectivamente. El error  $e(t)$  corresponde a la diferencia entre la distancia deseada y al distancia relativa al obstáculo.

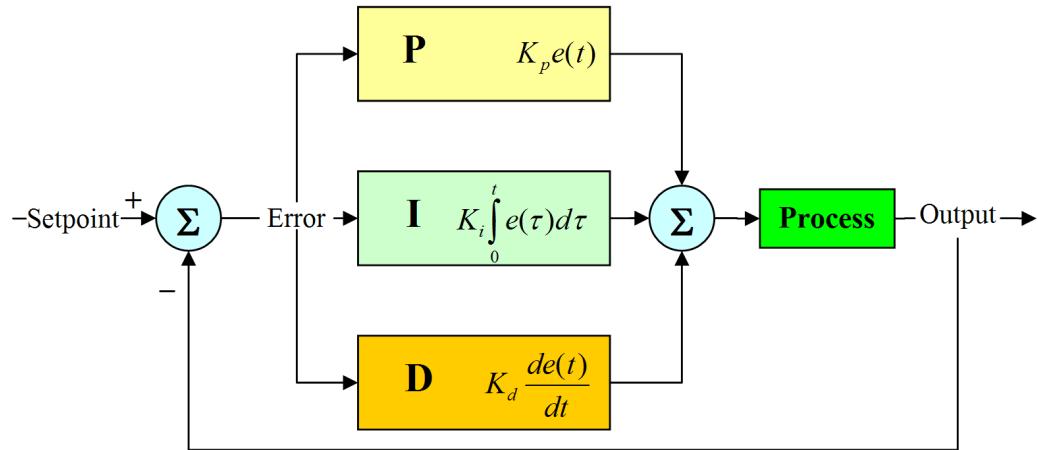


Figura 3.3: Diagrama de bloques del controlador PID [32].

Como se puede observar en la figura 3.3, el controlador PID se encuentra en un bucle de

retroalimentación. En este caso, el controlador recibe como input el error  $e$  de distancia al obstáculo, y como output, este entrega comandos de evasión en forma de señal PWM.

### 3.4 Arquitectura

Una vez que el UAV está en el aire, el controlador central ejecuta constantemente el algoritmo de detección de obstáculos, tomando en cuenta las distancias medidas por los sensores y las distancias de seguridad para cada dirección.

Los comandos del piloto pasan directamente al controlador de vuelo, siempre y cuando no se sobrepase el umbral de detección de colisiones.

Si alguno de los sensores inicia la detección de obstáculos, el controlador de posición inhibe el envío directo de comandos del piloto al controlador de vuelo, a su vez enviando comandos de evasión para evitar la colisión. Esto se explica gráficamente en la figura 3.4.

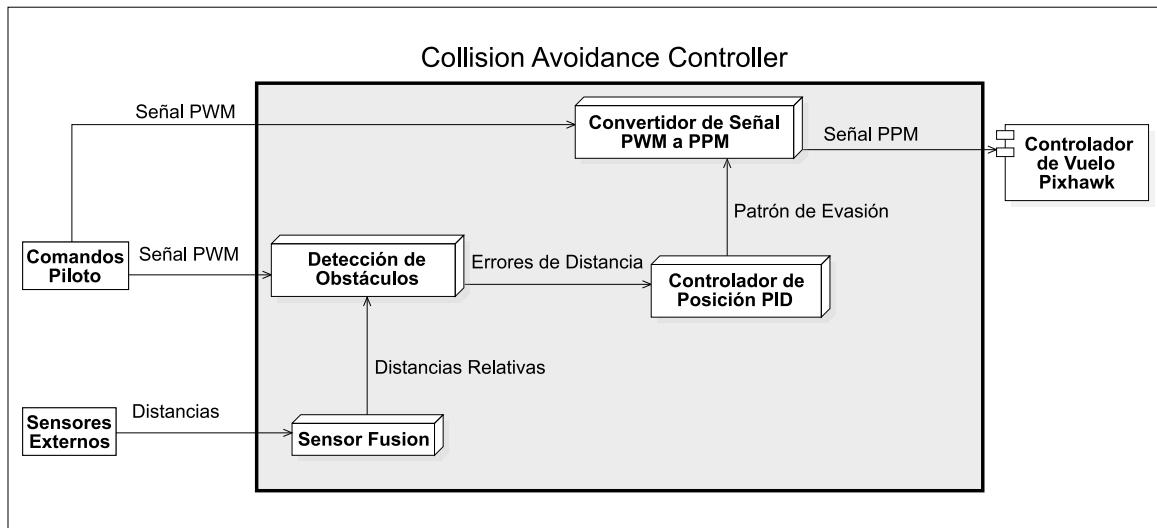


Figura 3.4: Sistema de prevención de colisiones.

En general, la arquitectura del sistema es sencilla: permitir el control del piloto hasta que se identifique un obstáculo.

## 3.5 Especificaciones Técnicas

Al ya tener definido el diseño del sistema de detección y evasión de colisiones, se pueden definir las especificaciones técnicas del UAV y del coprocesador sobre el cual se implementará el sistema de detección y evasión de colisiones.

### 3.5.1 UAV

Se construye un UAV utilizando un controlador de vuelo robusto y open source, además de un kit de UAV de alta calidad para resistir los rigores de las pruebas.

El Pixhawk de 3DR es un controlador de vuelo open source [5] utilizado en varios campos de investigación. Su firmware es totalmente modificable, posee gran cantidad de puertos adicionales para su expansión y puede ser utilizado en distintos tipos de vehículos, como aviones, helicópteros, UAVs y rovers.

Este controlador de vuelo tiene capacidades para enviar y recibir datos desde varias fuentes. Se utilizan los puertos de radiocontrolador y servo para controlar los motores en base a comandos PPM; y se utiliza el puerto de telemetría para monitorear en tiempo real el estado del controlador, desde el estado de la batería hasta la aceleración del UAV. Las señales de control se transmiten por radio a 2.4 GHz, y las señales de telemetría se transmiten mediante Bluetooth.

Para el hardware del UAV, se utiliza el kit Flamewheel F450 ARF, de la compañía DJI, el cual incluye el marco, motores, controlador de velocidad de motores, hélices y los cables necesarios para conectar la batería al UAV [18]. Utilizar un kit en vez de adquirir partes de forma independiente garantiza que estas son compatibles entre sí, evitando fallas eléctricas y de estabilidad del UAV [7].

La batería requerida para el funcionamiento del UAV debe contar con una capacidad y potencia capaces de garantizar suficiente tiempo de autonomía de vuelo y para asegurar un suministro estable de energía a los motores. Se utiliza para este trabajo la batería Turnigy

2.2, la cual es una batería de 3 celdas con capacidad de 2200 mAh, la cual da una autonomía de vuelo de entre 8 a 10 minutos, garantizando 11,1 volts de manera estable, lo cual es necesario para la alimentación tanto de los motores como el controlador de vuelo.

Para el manejo del UAV se necesita un control remoto y un receptor de control remoto. El control remoto utilizado es el Walkera Devo 7E, cuyo firmware open source permite modificaciones. Este control remoto transmite 7 canales, por lo cual se pueden enviar 7 comandos independientes al UAV simultáneamente. Por otra parte, el receptor Walkera RX701 recibe en los mismos 7 canales: estos envían señales en formato PWM, las cuales son transformadas por un codificador de señal a PPM, que es el formato de señal que controla el UAV.

### 3.5.2 Sensores y Microcontrolador

Se busca detectar colisiones de manera precisa y robusta, por lo cual se utilizan dos sensores distintos por cada una de las direcciones. Los detalles de la interpretación de ambos sensores para la medición de distancias se verán más adelante, en la sección [4.1](#).

Se utiliza el sensor ultrasónico Maxbotix MB1010 LV-MaxSonar-EZ1 para la medición de largas distancias: el sensor puede detectar paredes y obstáculos grandes entre 15 y 650 centímetros, y obstáculos más pequeños, como personas, entre 50 y 240 centímetros [29]. Las mediciones son realizadas emitiendo ondas de ultrasonido, las cuales son reflejadas al rebotar contra un obstáculo, que luego son detectadas por el sensor. Utilizando el tiempo entre la emisión y detección de la onda, más la constante de velocidad del sonido, se calcula la distancia entre el sensor y el obstáculo.

La medición de distancias más cortas se realiza con el sensor infrarrojo Sharp modelo GP2Y0A21YK0F; este tiene un rango de distancias de entre 10 y 80 centímetros [37]. Este emite un pulso infrarrojo y luego percibe la intensidad de onda infrarroja, con lo cual entrega un valor analógico que es luego transformado a distancia.

La interpretación de los sensores y modificación de señales de control se hace con un microcontrolador de uso general. Este requiere suficientes entradas para recibir los datos de los sensores, y la capacidad de computar y ejecutar los algoritmos de detección y evasión de

colisiones, modificando las señales PWM recibidas desde el transmisor de control remoto en tiempo real. El conjunto de algoritmos de detección y evasión de colisiones ejecutados por el microcontrolador se denomina controlador de evasión de colisiones, o CAC por sus siglas en inglés.

Se utiliza el Arduino Mega, el cual tiene el microcontrolador ATmega1280. Este microcontrolador cuenta con la capacidad de ajustar su frecuencia de funcionamiento, lo cual se requiere para poder interceptar y modificar correctamente las señales enviadas al controlador de vuelo. Cuenta con un gran número de compuertas analógicas, necesarias para recibir los datos de los sensores anteriormente mencionados. Finalmente, su capacidad de mantener 4 conexiones simultáneas de tipo serial [4] le permiten ser reprogramado, mantener comunicación con el UAV y conectarse con un computador externo, pudiendo enviar y recibir información en tiempo real.

Las conexiones tipo serial del microcontrolador pueden establecerse mediante cable o mediante otros sistemas; en particular, la conexión usada para el monitoreo del UAV se realiza de forma inalámbrica, mediante radiofrecuencia. En particular, se usa un kit XBee Series 1, el cual consiste de una antena que se conecta al microcontrolador, y de una antena recíproca que se conecta a un computador por USB. Estas antenas pueden transmitir a una distancia máxima de 30 metros [17].

## 3.6 Resumen

En este capítulo se detalló la forma en que se van a realizar las mediciones de distancia, fusionando los datos tomados por los sensores infrarrojos y ultrasónicos; se describe el filtro lineal de Kalman, usado para obtener mejores mediciones, y se explica cómo funcionará la detección y evasión de colisiones, una vez que la distancia al obstáculo es menor a la distancia de seguridad.

Los algoritmos descritos en esta sección requieren fijar varios parámetros de forma empírica; es necesario determinar cuál será la distancia de seguridad para cada dimensión y las ganancias del controlador PID que se usarán para realizar las maniobras de evasión.

Adicionalmente se describió, de entre las distintas opciones descritas en la sección [2.3](#), cuáles fueron las piezas utilizadas para armar el UAV: el hardware, el controlador de vuelo, los sensores, las antenas y el controlador de evasión de colisiones.

# **Capítulo 4**

## **Implementación**

Una vez que se define el diseño, integrando la extensión vertical al sistema de detección y evasión de colisiones, se procede a la programación del CAC. Para implementar este sistema, se empieza por implementar el sistema realizado por Rambabu et al. [36]. Luego, se implementa el sistema de detección y evasión de colisiones de manera vertical. La implementación de este trabajo contempla la medición de distancias en las seis direcciones, las cuales son luego utilizadas para la detección de colisiones, para finalmente evadir dichas colisiones. A continuación se describen los pasos de la implementación de este trabajo.

### **4.1 Medición de Distancias**

Para realizar la medición de distancias, se utilizan sensores posicionados en cuatro direcciones cubriendo el plano horizontal y otros dos para las direcciones superior e inferior. Se utilizan sensores ultrasónicos, los cuales envían un pulso de sonido de alta frecuencia, para luego medir cuánto tiempo se demora en escuchar el eco del sonido reflejado. Estas mediciones de distancia son entregadas en forma de señal analógica, las que el CAC convierte en valores digitales para luego transformar a centímetros. Luego se efectúa un proceso de filtrado digital, usando primero el algoritmo filtro de mediana y después el algoritmo filtro lineal de Kalman.

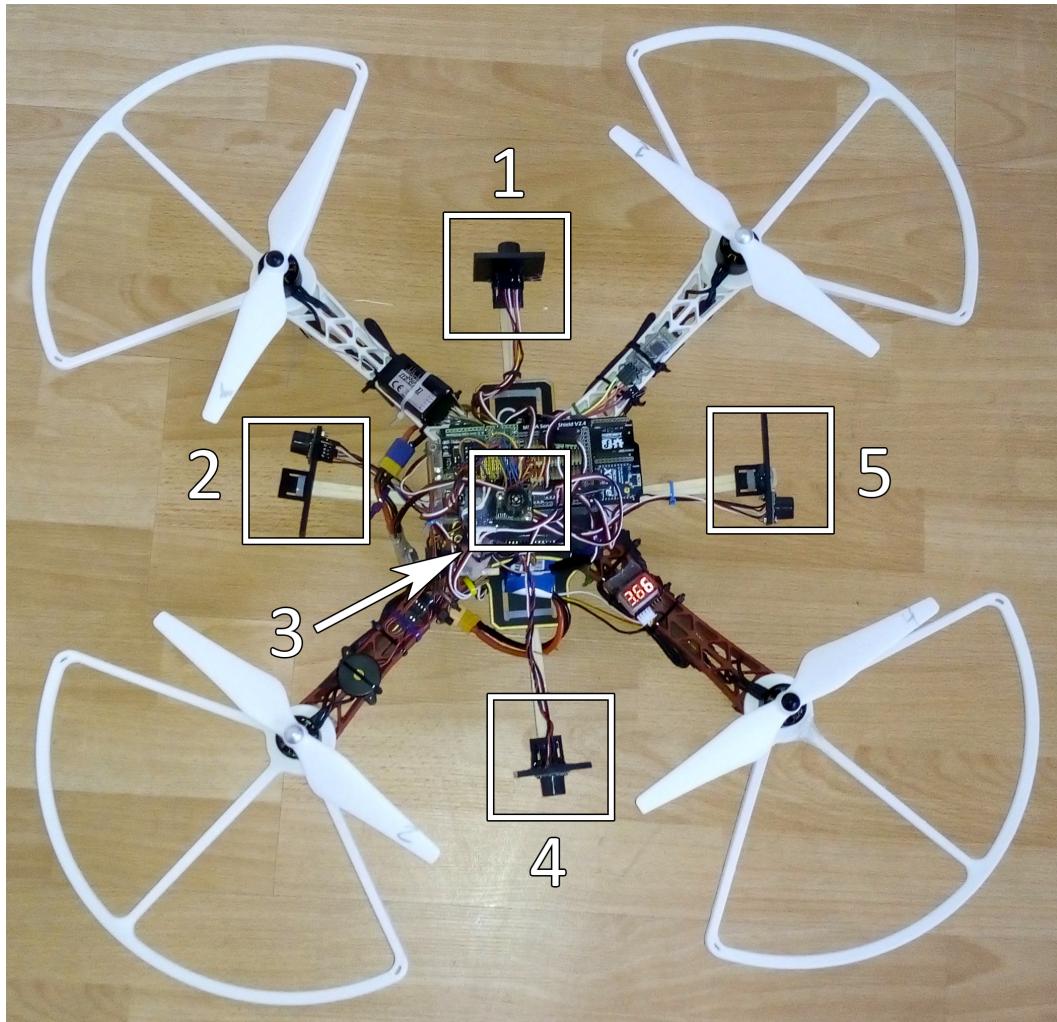


Figura 4.1: Vista superior UAV, mostrando los cuatro sensores horizontales y el sensor superior.

Al utilizarse multiples sensores ultrasónicos, estos deben ejecutarse en secuencia, con tal de que no se interfieran entre sí, evitando que un sensor reciba el eco de otro sensor en funcionamiento. Adicionalmente, estos sensores deben ser posicionados lo más lejos posible de los motores del UAV, con tal de evitar interferencia tanto de ruido de las hélices como del campo electromagnético producido por los mismos [30].

En la figura 4.1 se puede apreciar una vista superior del UAV, donde se puede apreciar el posicionamiento de los sensores horizontales y superior. Los sensores horizontales se encuentran posicionados en el punto medio entre cada par de motores, con tal de estar alejados

de los motores y al mismo tiempo realizar mediciones desde un punto representativo del tamaño del UAV.

### 4.1.1 Encadenamiento de Sensores

Por la forma de medir distancias de estos sensores, realizar mediciones simultáneamente genera interferencia en las mismas, por lo cual es necesario que las mediciones sean realizadas de manera secuencial. Los sensores utilizados tienen una entrada de pulso digital para iniciar su lectura y una salida digital para gatillar la lectura del siguiente sensor. De esta manera, los sensores se encadenan, un sensor gatillando al siguiente. El orden óptimo de gatillamiento de sensores se ha determinado empíricamente, siendo su orden el descrito en la figura 4.1. En este caso, el sensor delantero emite un pulso ultrasónico, detecta su eco y gatilla al sensor izquierdo, y así sucesivamente.

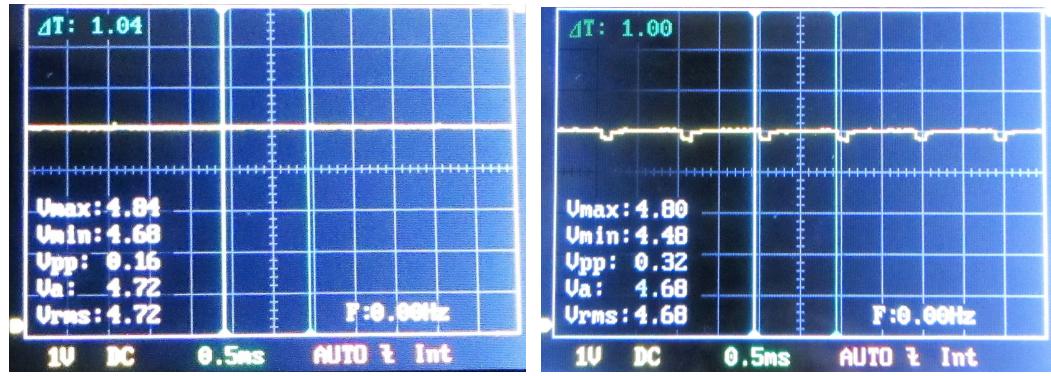
### 4.1.2 Sensores Infrarrojos

Iniciando las pruebas con los sensores infrarrojos se encontró un problema: al conectar los sensores infrarrojos, los valores entregados por los sensores ultrasónicos se alteraban a tal punto de resultar inutilizables. Un estudio del sistema encontró una baja de voltaje a nivel de todo el circuito cada vez que se encendía un sensor infrarrojo.

Como se puede observar en la figura 4.2b, cada milisegundo el sensor infrarrojo emite un pulso que consume mucha corriente, disminuyendo el voltaje del sistema y alterando las mediciones de los otros sensores. Como comparación, en la figura 4.2a se puede apreciar el voltaje antes de conectar un sensor infrarrojo.

La baja de voltaje con cada pulso es de 0,32 volts, lo cual resulta en mediciones impredecibles desde el resto de los sensores, haciendo infactible el uso de los sensores infrarrojos.

Luego de contactar a Rethnaraj Rambabu, co-autor de la investigación sobre el cual se está trabajando, él menciona que los sensores infrarrojos finalmente no fueron necesarios ya que los sensores ultrasónicos realizan buenas mediciones para las distancias relevantes en este



(a)

(b)

Figura 4.2: Voltaje sin sensor infrarrojo conectado (a) y voltaje luego de conectarlo un sensor infrarrojo (b).

trabajo, por lo cual los sensores infrarrojos son descartados de la implementación.

### 4.1.3 Limpieza de mediciones

Las mediciones realizadas por los sensores, aun siendo realizadas por encadenamiento, tienen ruido y errores, tanto por interferencia ambiental como por la turbulencia generada por el funcionamiento de los motores del UAV. Para obtener mediciones más precisas, éstas son procesadas por dos procesos de filtrado: filtro de mediana y filtro lineal de Kalman.

#### Filtro de Mediana

En estadística, la mediana es el dato que se encuentra en el lugar central de los datos en estudio, al estar éstos ordenados de menor a mayor. Por la forma en que se da el ruido en las mediciones, la mediana de las mediciones obtenidas es suficientemente cercana a la distancia real siendo medida. Se prueban distintas cantidades de valores a estudiar, con seis mediciones procesadas por filtro de mediana se obtiene mediciones suficientemente certeras. La cantidad de valores a tomar para el filtro de mediana debe balancear la exactitud de la medición final con el retraso introducido al tomar demasiadas muestras; se encontró que con seis mediciones procesadas se obtienen mediciones lo suficientemente certeras, sin agregar

demasiado retraso.

### Filtro Lineal de Kalman

Luego de procesar las mediciones con el filtro de mediana, éstas son procesadas por el filtro lineal de Kalman, con lo cual se reducen aún más los errores y el ruido de las mediciones. A través de ensayo y error se definen los siguientes parámetros para el filtro lineal de Kalman:

- $Q = 0,3$
- $R = 2$
- $P = 3$

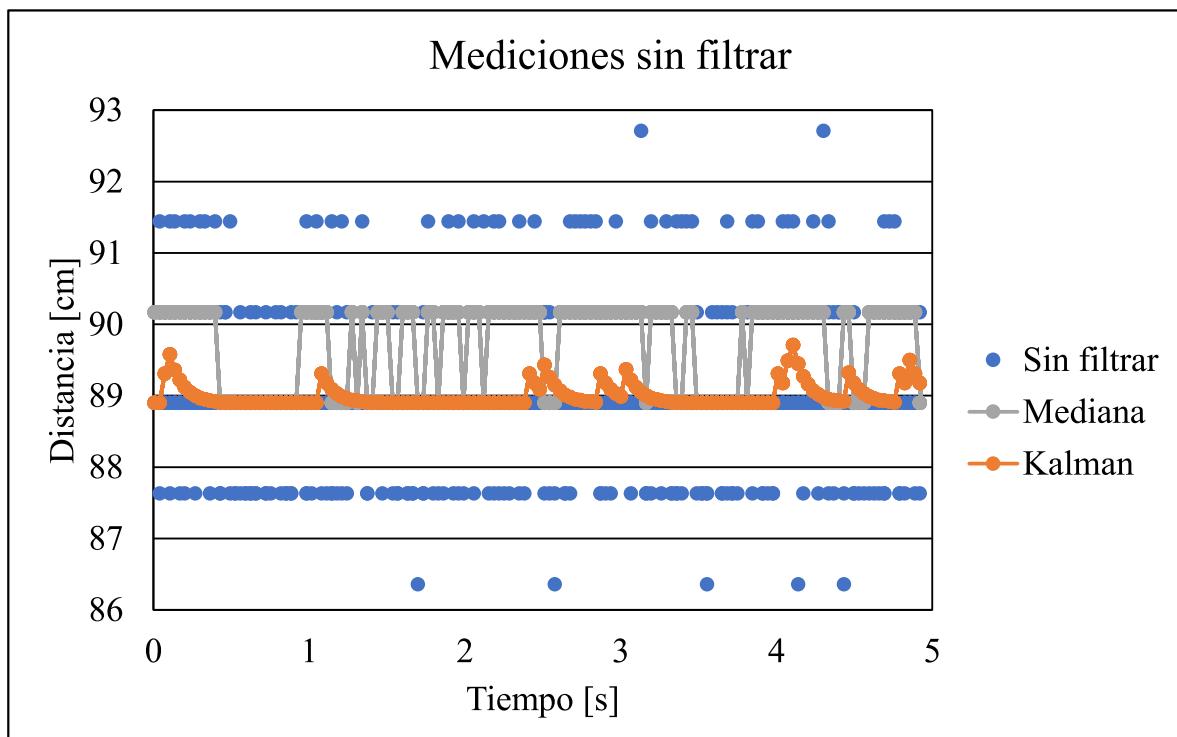


Figura 4.3: Con UAV en reposo, mediciones de distancia sin procesar y luego de ser procesadas por filtro de mediana y filtro lineal de Kalman.

La figura 4.3 muestra las mediciones realizadas por un sensor ultrasónico a una distancia de

89 centímetros, en color azul. De cada seis mediciones se toma la mediana (acá en color gris) y la serie es procesada por el filtro lineal de Kalman, en color naranjo.

Nótese que las mediciones del sensor ultrasónico sin filtrar oscilan entre los 87 y 92 centímetros. Luego de aplicar el filtro de mediana, la distancia varía drásticamente entre 89 y 90 centímetros; finalmente, mediante el filtro lineal de Kalman se determina una distancia definitiva de 89 centímetros, con variaciones menores.

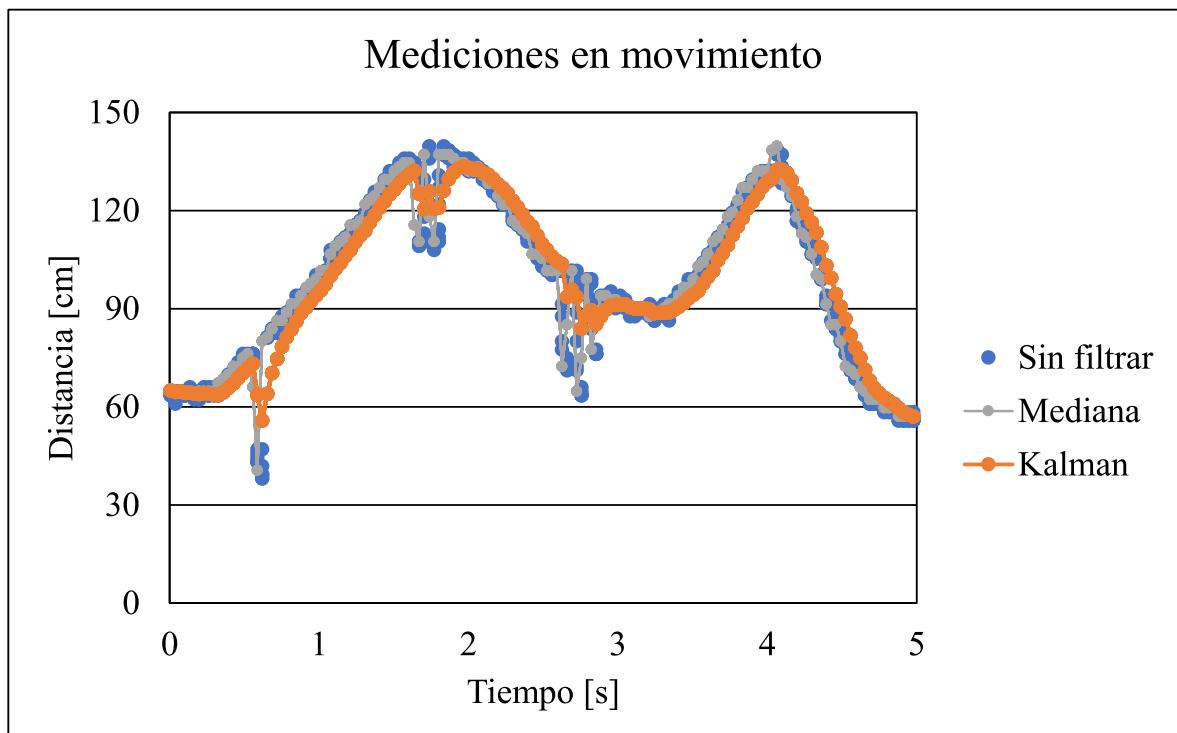


Figura 4.4: Con UAV en movimiento, mediciones de distancia sin procesar y luego de ser procesadas por filtro de mediana y filtro lineal de Kalman.

La figura 4.4, por otra parte, muestra las mediciones realizadas por el sensor mientras el UAV se encuentra en movimiento. Las mediciones se filtran de la misma forma: de cada 6 mediciones del sensor (en color azul) se toma la mediana (en color gris), y la serie se pasa por el filtro lineal de Kalman (en color naranjo).

La escala de la medición implica que las variaciones en las mediciones del sensor, vistas en el gráfico anterior, son casi irrelevantes; sin embargo, se puede ver que en ciertos puntos de la prueba se producen discontinuidades serias en las mediciones, las cuales se producen debido

a interferencias. Estas mediciones no son corregidas por el filtro de mediana; sin embargo, mediante el filtro lineal de Kalman, la medición final de distancia queda suavizada en gran medida.

Luego de ser procesadas las mediciones, éstas se encuentran en condiciones de ser utilizadas por el CAC para la detección de colisiones.

## 4.2 Detección de Colisiones

Para realizar la detección de colisiones, primero se define la distancia de seguridad ( $x_d, y_d, z_d$ ) para cada dirección del UAV, de tal forma que no se permitan obstáculos dentro de esa distancia. Luego de cada ciclo de mediciones, se comparan las distancias actuales luego del proceso de filtrado ( $x_r, y_r, z_r$ ) a las distancias de seguridad; en caso de que la distancia de seguridad sea menor a la medición, se considera que se ha detectado un obstáculo en dicha dirección. En este caso, se establecen los valores de error de distancia ( $x_e, y_e, z_e$ ) del UAV relativo al obstáculo. Luego de cada ciclo de mediciones, estas pasan por el proceso de filtrado y son comparadas a la distancia de seguridad.

Se pueden detectar múltiples colisiones simultáneamente, pero se ignora el caso en que se detecten obstáculos en direcciones opuestas, ya que en este caso el UAV se encontraría rodeado, por lo cual no se pueden ejecutar maniobras de evasión. En este caso, se desactiva el CAC hasta que se detecten colisiones que el sistema si va a ser capaz de evadir. Por lo tanto, se trabaja la detección de colisiones hasta tres colisiones perpendiculares simultáneas.

## 4.3 Evasión de Colisiones

Al detectar colisiones, y al encontrarse éstas dentro de los patrones de evasión, se ignoran los comandos del piloto del UAV, siendo estos reemplazados por comandos generados por el CAC, el cual utiliza el método de campo de repulsión, alejando el UAV del obstáculo. Los valores de pitch, roll y throttle ( $\phi_{CA}, \theta_{CA}, Z_{CA}$ ) son calculados utilizando los errores ( $x_e, y_e, z_e$ )

y utilizando el controlador de posición PID descrito en la sección 3.3.

$$\theta_{CA} = K_p x_e(t) + K_I \int_0^t x_e(\tau) d\tau + K_D \frac{dx_e(t)}{dt} \quad (4.1)$$

$$\phi_{CA} = K_p y_e(t) + K_I \int_0^t y_e(\tau) d\tau + K_D \frac{dy_e(t)}{dt} \quad (4.2)$$

$$Z_{CA} = K_p z_e(t) + K_I \int_0^t z_e(\tau) d\tau + K_D \frac{dz_e(t)}{dt} \quad (4.3)$$

Las constantes de proporcionalidad, integración y derivación son la ganancia con la cual se amplifica cada parte del controlador PID [24]. Se utilizan los siguientes valores para el sistema:

- $K_p = 1,2$
- $K_I = 1,75$
- $K_D = 0,6$

Estos valores son obtenidos del trabajo de Cheong et al. [15], el cual utiliza el mismo UAV que este trabajo.

El UAV realiza movimientos según las señales recibidas por el controlador de vuelo por los distintos canales, en este caso, 7 canales. Todo canal envía una señal numérica entre 1100 y 1900, la cual es interpretada por el controlador de vuelo. Los primeros cuatro canales de comunicación están relacionados a comandos de movimiento del UAV [8]:

- Canal 1: roll —movimiento lateral
- Canal 2: pitch —movimiento hacia adelante o atrás
- Canal 3: throttle —movimiento vertical
- Canal 4: yaw —rotación sobre su eje

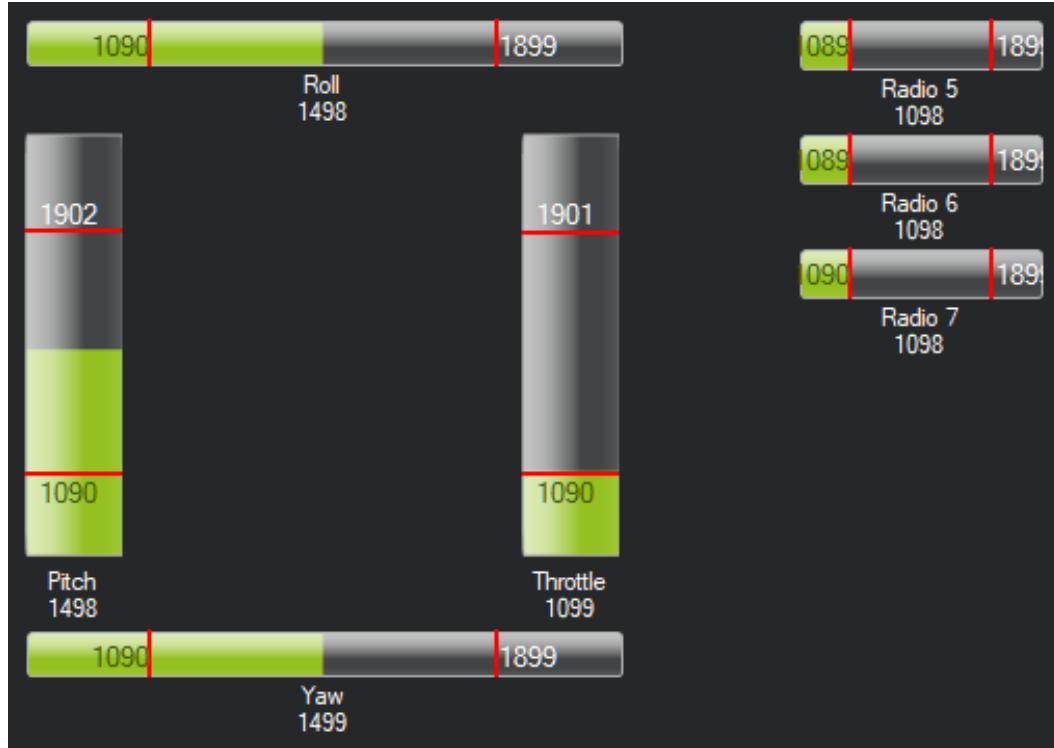


Figura 4.5: Rangos de valores para cada canal del UAV [6].

Para los canales 1, 2 y 4, el valor 1500 es neutro, lo cual significa que no realizan movimiento, mientras que valores superiores o inferiores significan movimiento en un sentido u otro. El canal 3 representa el movimiento vertical del UAV a través de la potencia de los motores, en este caso el valor mínimo significa que los motores se encuentran detenidos, mientras que el valor máximo significa que los motores se encuentran funcionando a su máxima potencia.

Los canales restantes se utilizan para realizar ajustes de comportamiento del UAV durante el vuelo. Para este sistema, se utiliza el canal 7 para activar o desactivar el CAC desde el control remoto.

En la figura 4.5 se pueden observar los distintos canales de comunicación con sus respectivos valores. Las barras verdes muestran el valor actual de cada canal, las líneas rojas muestran los valores superiores e inferiores que ha recibido el UAV.

Una vez calculados los valores para cada movimiento del UAV, éstos son convertidos a señales PWM para luego ser enviados al controlador de vuelo Pixhawk en forma de una

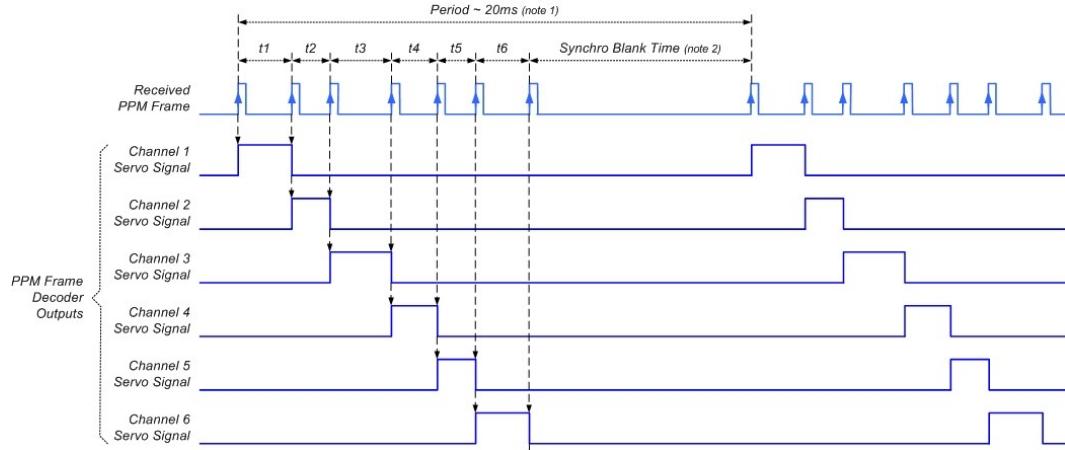


Figura 4.6: Transformación de señales entre PPM y PWM [27].

sola señal PPM, uniendo los valores PWM de los distintos canales. Como se puede observar en la figura 4.5, el controlador de vuelo recibe 7 señales PWM distintas simultáneamente, cada señal por un canal distinto, cuyo rango se encuentra entre 1100 y 1900. Los distintos canales son transformados desde PWM, que es lo que utiliza el CAC para la generación de los comandos de evasión, para luego transformarlos a señal PPM, la cual es utilizado por el controlador de vuelo, tal como se puede observar en la figura 4.6.

### 4.3.1 Patrones de Evasión

Para realizar las maniobras de evasión de colisiones, el CAC invalida los comandos enviados por control remoto, sobrescribiéndolos con comandos de pitch, roll y throttle acorde a la colisión detectada. Se definen patrones de evasión para una, dos o tres colisiones perpendiculares simultáneas. En esta implementación se trabaja con adelante, izquierda y arriba, con el objetivo de demostrar la factibilidad del sistema.

Los patrones implementados son los siguientes:

1. Una dimensión:

- (a) Adelante
- (b) Izquierda

(c) Arriba

2. Dos dimensiones:

- (a) Adelante - izquierda
- (b) Adelante - arriba

3. Tres dimensiones:

- (a) Adelante - izquierda - arriba

Para las colisiones en el plano horizontal, o sea adelante e izquierda, se generan comandos de evasión con valores acotados entre 1300 y 1520, siendo 1300 movimiento en la dirección opuesta al obstáculo, 1500 no movimiento, y 1520 movimiento ligeramente hacia el obstáculo. Se asigna un valor final mayor a la posición neutra para que, al terminar el patrón de evasión, se contrarreste ligeramente el momentum del UAV. Estos valores se obtienen a través de ensayo y error, ya que valores menores a 1300 generaban un movimiento evasivo muy brusco, generando accidentes, y los valores sobre 1500 generan un estado final de UAV más estable que la posición neutra. En el código 4.1 se muestra la implementación, utilizando la librería de Arduino PID creada por Brett Beauregard [11].

---

```
1 PID PID_front(&Input1, &Output1, &Setpoint, 1.2, 1.75, 0.6, P_ON_M,
2   ↪ REVERSE);
3 PID_front.SetOutputLimits(1300, 1520);
4 PID_front.SetMode(AUTOMATIC);
5 PID_front.SetSampleTime(25);
```

---

Listado 4.1: Implementación controlador PID evasión frontal

Esta implementación genera el comando de evasión cada 25 milisegundos. Luego se define dentro del ciclo de ejecución de Arduino el input como el valor de error, setpoint como la distancia de seguridad en la dirección frontal, y output es el comando de evasión generado. Se genera una salida en reversa (REVERSE), lo cual significa que mientras mayor sea el error, menor debe ser el valor generado por el PID. El parámetro P\_ON\_M indica que el valor de salida debe ser proporcional a la medición de entrada, ya que se trata de un proceso en el

cual la salida del PID controla la tasa de cambio de la entrada [12]. Este comando de evasión se genera constantemente, siendo enviado al controlador de vuelo desde el CAC sólo en caso de detectarse colisión en alguna dirección del plano horizontal, anulando los comandos del piloto.

El algoritmo de evasión en caso de detectar un obstáculo encima del UAV utiliza como valor de máxima evasión 1350, siendo 1425 el valor final de evasión. Se obtienen estos valores a través de ensayo y error, cuidando de evadir el obstáculo y que la maniobra de evasión no resulte en que el UAV descienda descontroladamente.

## 4.4 Monitoreo

Para monitorear en tiempo real el estado del UAV y su distancia a obstáculos, se usa una de las interfaces serial del CAC, mediante la cual se transmiten las mediciones de distancia calculadas por el mismo, a un computador externo. De este modo se pueden depurar las mediciones de distancia antes y después de ser procesadas, para diagnosticar posibles problemas en el sistema. Se utiliza el software SerialPlot para la lectura y visualización de esta información, y la transmisión de la interfaz serial se realiza usando la antena XBee descrita en la sección [3.5.1](#).

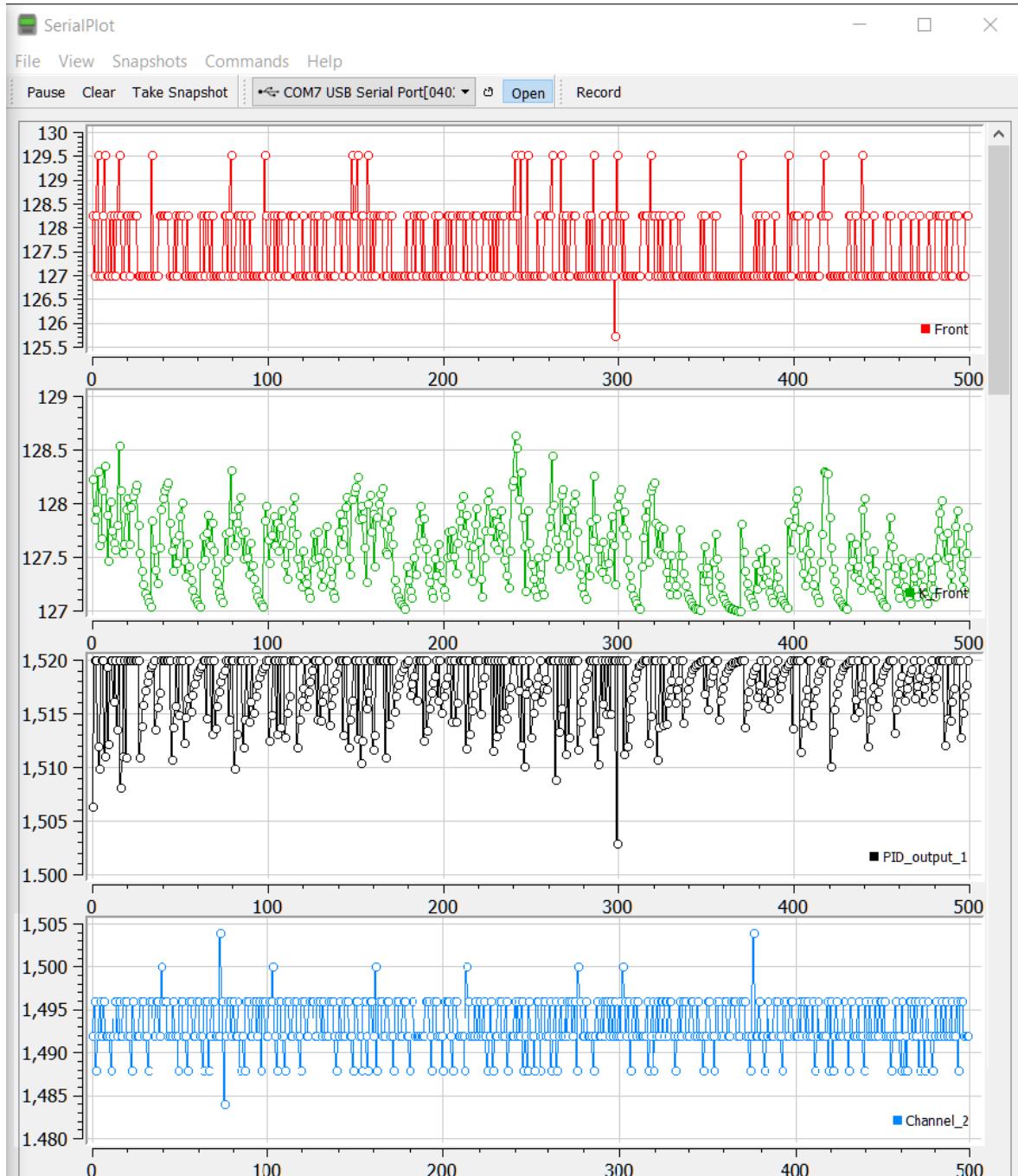


Figura 4.7: Ejemplo visualización mediciones enviadas desde Arduino a computador a través de antena XBee a un rango de 10 metros [39].

En la figura 4.7 se pueden apreciar las mediciones recibidas por el computador usando la interfaz serial. El primer gráfico, en rojo, muestra las mediciones de distancia hacia adelante;

el segundo gráfico, en verde, muestra las mismas mediciones luego de ser filtradas; el tercer gráfico, en negro, muestra los comandos de evasión de colisiones generados por el algoritmo de evasión de colisiones; mientras que el cuarto gráfico, en azul, muestra los comandos que recibe el UAV. En este ejemplo el UAV se encuentra estacionario, a una distancia de 127 centímetros de un obstáculo frontal.

Adicionalmente, usando el puerto de telemetría mencionado en la sección 3.5.1 y el software Mission Planner [6], se puede realizar monitoreo de las señales de radio recibidas por el controlador de vuelo, diagnosticar problemas y desactivar el UAV de manera remota en caso de incidentes. Cabe notar que no es recomendable utilizar el UAV sin que este se encuentre conectado remotamente a un computador: como el CAC interfiere con los comandos del piloto, cualquier problema con el sistema implicará no poder desactivar el UAV utilizando el control remoto, llevando a resultados catastróficos.

## 4.5 Discusión

En términos generales, el CAC funciona de la siguiente manera:

---

```
1 if CAC on and (distance < safety distance) then
2     Calculate distance errors (desired-current)
3     PID-position control
4     generate attitude
5     convert attitude to PWM
6     override command by writing PWM
7 end calculate
8 else
9     PWM in = PWM out
10 end if
```

---

Listado 4.2: Funcionamiento general del CAC.

El código 4.2 describe un ciclo completo de funcionamiento del CAC para el caso de detectar una colisión. Primero uno se debe asegurar que el sistema se encuentre activado, en este caso,

éste se activa mediante un switch en el control remoto. Si ese switch se encuentra activado y se detecta una colisión, se calcula el error  $r_e = r_r - r_d$ . Con esos errores, el controlador de posición PID genera *attitude* del UAV, para luego transformar el mismo a señales PWM y enviarlo al controlador de vuelo. Si no se detecta colisión o no se encuentra activado el CAC, los comandos del piloto se transmiten directamente al controlador de vuelo Pixhawk.

# Capítulo 5

## Evaluación

Luego de realizada la implementación del sistema de detección y evasión de colisiones, se procede a evaluar el sistema implementado. En primer lugar se hace una evaluación del sistema de evasión de colisiones en dos dimensiones, en comparación con el trabajo de Rambabu et al. [36]; para luego evaluar de forma independiente la extensión del sistema a la dirección vertical.

Se realizan pruebas en una, dos y tres dimensiones, en las cuales los obstáculos se encuentran estáticos. Se utilizarán las direcciones delantera, izquierda y superior para realizar las pruebas; ya que, al ser un sistema simétrico, se postula que el resultado de las pruebas descritas en el capítulo 3 tendrán resultados equivalentes para el resto de las direcciones.

Los gráficos obtenidos mediante las interfaces serial y de telemetría, descritas en el capítulo 4, permiten obtener datos concretos al ejecutar las pruebas, dejando registrados mediciones de distancias, limpia de distancias, comandos enviados desde el control remoto, comandos recibidos por el controlador de vuelo, e incluso instancias de choques.

## 5.1 Preparación de las pruebas

Los obstáculos utilizados en las evaluaciones son planchas de poliestireno expandido, mejor conocidos como plumavit. Sus dimensiones son de 200 centímetros de alto, 100 centímetros de ancho, y 5 centímetros de profundidad. Se utiliza este material debido a su bajo peso en relación a su tamaño, y su capacidad de absorber la fuerza de un impacto, lo cual ayuda a cuidar la integridad del UAV durante las pruebas. Al ser los obstáculos planos, se pueden realizar con claridad las mediciones relativas a los obstáculos, facilitando la obtención de datos para su posterior estudio.

### 5.1.1 Revisión del entorno de pruebas

La ejecución de las pruebas requiere de cierta preparación previa para asegurar un entorno consistente. En primer lugar, las pruebas deben realizarse en un espacio cerrado, sin corrientes de aire externas, por lo cual se debe cuidar de cerrar toda puerta y ventana con salida al exterior, con tal de minimizar el flujo de aire en el área. Adicionalmente, se deben iniciar las pruebas con el UAV alejado de las murallas, especialmente de las esquinas. Esto se debe a que el funcionamiento del mismo UAV genera corrientes de aire: si el UAV no tiene suficiente espacio en alguna dirección, el aire que desplazan las hélices se refleja en las paredes, desestabilizando todo el conjunto [23].

El hardware reemplazable del UAV, como las hélices y las baterías, se mantiene consistente entre pruebas, cuidando que la batería siempre quede posicionada cerca del centro de masa del UAV. Antes de encender el UAV, se prepara el computador con el que se realizará el monitoreo, conectando la antena XBee y el transmisor Bluetooth. Se posiciona el UAV sobre una superficie plana y nivelada para luego conectar la batería, lo cual enciende el UAV. Luego se debe encender el control remoto, cuidando de mantener las palancas en posición neutra. A continuación, se conecta el UAV al software Mission Planner [6] a través de Bluetooth, y se inicia la conexión entre el CAC y computador externo, utilizando las antenas XBee para su comunicación y el software SerialPlot [39] para su visualización en tiempo real, según lo determinado en el capítulo 4. Luego de revisar que los sensores se encuentren funcionando

correctamente y que el UAV se encuentre en condiciones para volar, se activa el CAC.

## 5.2 Pruebas

Para realizar la pruebas al sistema de detección y evasión de colisiones, lo primero es definir las distancias de seguridad ( $x_d, y_d, z_d$ ) en centímetros. Para estas pruebas, se definen dichos parametros como:

- $x_d = 60$  centímetros
- $y_d = 60$  centímetros
- $z_d = 80$  centímetros

Estos valores son definidos a través de ensayo y error durante la implementación del sistema. Se posiciona el UAV en el piso, estando este a distancias superiores a las de seguridad de los obstáculos. Se activa el UAV desde el control remoto, dando inicio a la prueba, luego de lo cual se encienden los motores desde el control remoto. El sistema se encuentra listo para realizar las pruebas, por lo cual se debe tener cuidado de mantenerse lejos del UAV, y en caso de perder el control, desactivar el CAC para luego apagar los motores del UAV desde el control remoto. En caso de no tener conexión entre control remoto y UAV, se desactiva el UAV desde el software Mission Planner enviando el comando a través de la conexión por Bluetooth.

Para la realización de pruebas específicas, se definen los parámetros del sistema acorde a la prueba a realizar, se establecen las direcciones de evasión activas, se compila el programa y se carga al Arduino Mega.

Para todas las pruebas, el resultado esperado es que el UAV no colisione con los obstáculos. El objetivo principal es probar que el sistema de detección y evasión de obstáculos en espacios cerrados funciona, y probar si la expansión del sistema propuesto por Rambabu et al. [36] a tres dimensiones previene más casos de colisión sin perjudicar el sistema de detección y evasión de colisiones en dos dimensiones.

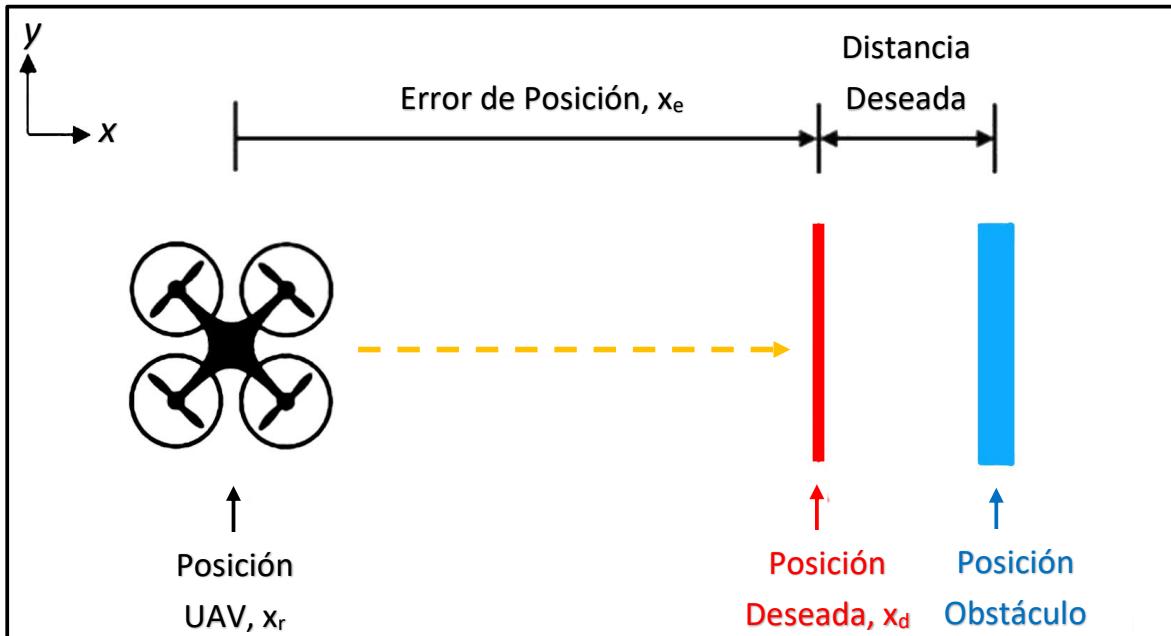


Figura 5.1: Evaluación frontal de detección y evasión de colisiones

### 5.2.1 Obstáculo estático

Para las pruebas de obstáculo estático en una dimensión, se activa una dirección para la detección y evasión de obstáculos. Se evalúan tres direcciones individualmente: adelante, izquierda y arriba. Se definen los parámetros del sistema para evasión en una dirección, especificando la dirección activada y la distancia de seguridad. Con el UAV en movimiento y el CAC activado, se mueve el UAV en la dirección del obstáculo, manteniendo el comando de movimiento hacia el obstáculo de forma constante. La figura 5.1 muestra visualmente la configuración de prueba.

Esta evaluación tiene como objetivo probar el procedimiento de anulación de los comandos del piloto en caso de detectarse colisión, y evaluar la detección y evasión de colisiones de manera individual. Al mantener el comando de movimiento hacia el obstáculo, cada vez con distinta intensidad, el UAV se acerca al obstáculo nuevamente, pero a distintas velocidades. Se realiza esto con el objetivo de calcular la velocidad máxima a la cual se puede acercar al obstáculo y no colisionar con el mismo.

Se evalúa análogamente el sistema de evasión y detección de colisiones hacia la izquierda



Figura 5.2: Campo de pruebas en espacio cerrado.

y hacia arriba de manera independiente. Luego se evalúa el sistema con obstáculos en dos direcciones simultáneamente, situando los obstáculos adelante y a la izquierda del UAV, para probar la evasión en dos dimensiones de manera horizontal, y situando obstáculos adelante y arriba para probar la evasión de manera horizontal y vertical simultáneamente. Para las pruebas en tres dimensiones, se sitúan obstáculos adelante, a la izquierda y arriba del UAV. Se evalúan colisiones lo más simultáneas posibles en las tres direcciones, con el objetivo evaluar si es posible realizar maniobras de evasión en tres dimensiones en tiempo real y manteniendo la integridad del UAV.

En la figura 5.2 se muestra como se ha organizado el campo de pruebas para las evaluaciones en interiores.

Para estas pruebas, se espera que el UAV no colisione en la dirección que se esté evaluando, y que al estar a una distancia mayor a la distancia de seguridad, que el piloto pueda controlar el UAV nuevamente.

# **Capítulo 6**

## **Resultados**

Una vez terminadas las implementaciones, se procedió a realizar las pruebas descritas en el capítulo [5](#). Estos resultados luego serán comparados a los del trabajo de Rambabu et al. [36].

### **6.1 Formato**

A través del sistema de monitoreo del CAC y UAV, se obtuvo cada 32 milisegundos las distancias medidas en las tres direcciones relevantes, las distancias filtradas, el valor de salida del controlador PID para los tres canales relevantes, y los comandos recibidos por el controlador de vuelo. Estos datos son visualizados con dos gráficos por cada dirección: uno de la distancia relativa del UAV a obstáculos en dicha dirección, y otro con los valores enviados al controlador de vuelo para controlar el movimiento en dicha dirección. Las secciones blancas de los gráficos significan que el piloto está en control del UAV, mientras que las secciones verdes representan momentos en que el CAC está en control del UAV, ejecutando maniobras de evasión. Los diamantes rojos representan instantes en que se activó momentáneamente el CAC por interferencia en las mediciones. La distancia de seguridad, bajo la cual se activa el CAC, se ve representada por una linea azul punteada a la altura de la distancia de seguridad correspondiente al caso.

Se define como zona de riesgo la distancia desde el obstáculo hasta la distancia de seguridad,

bajo la cual el UAV se encuentra en riesgo de colisión, por lo cual el CAC toma control del mismo para ejecutar comandos de evasión.

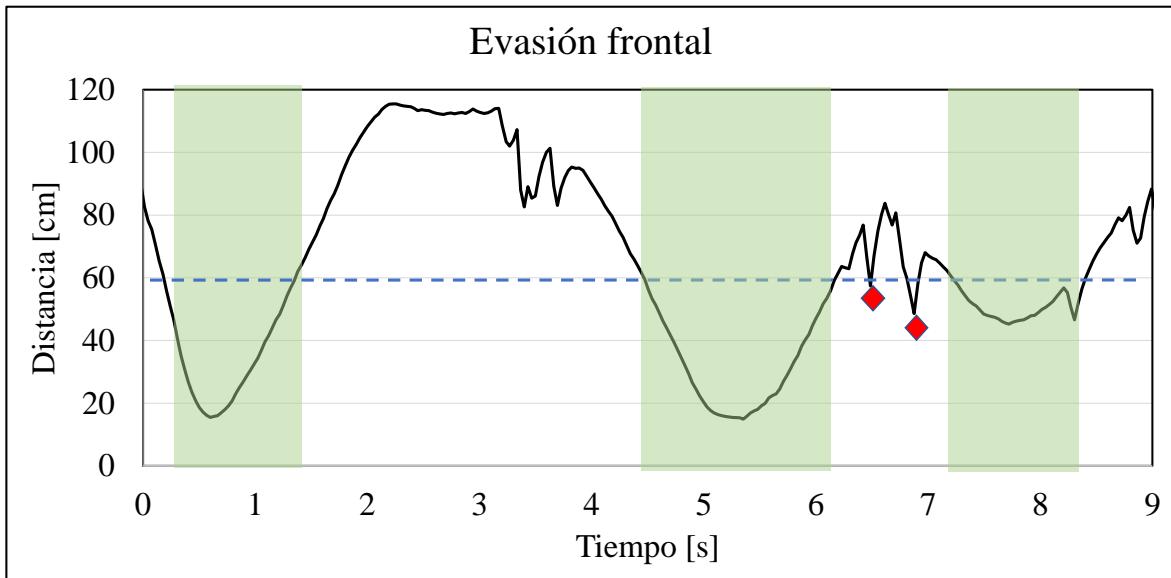
Se define el resultado de una prueba como exitosa cuando el obstáculo no es chocado, aunque el usuario esté ejecutando acciones que normalmente llevarían a tal choque; devolviendo finalmente el control al usuario, sin necesidad de interrumpir el vuelo. Por ejemplo, en una prueba exitosa, el usuario dirigiría el UAV hacia una pared; el CAC tomaría control del aparato al ingresar a una zona de riesgo, y devolvería el control al usuario una vez que el UAV se encuentre fuera de aquella zona; durante todo este proceso, el UAV no debe chocar ni perder el control, permitiendo que el usuario siga volando con normalidad.

## 6.2 Una Dimensión

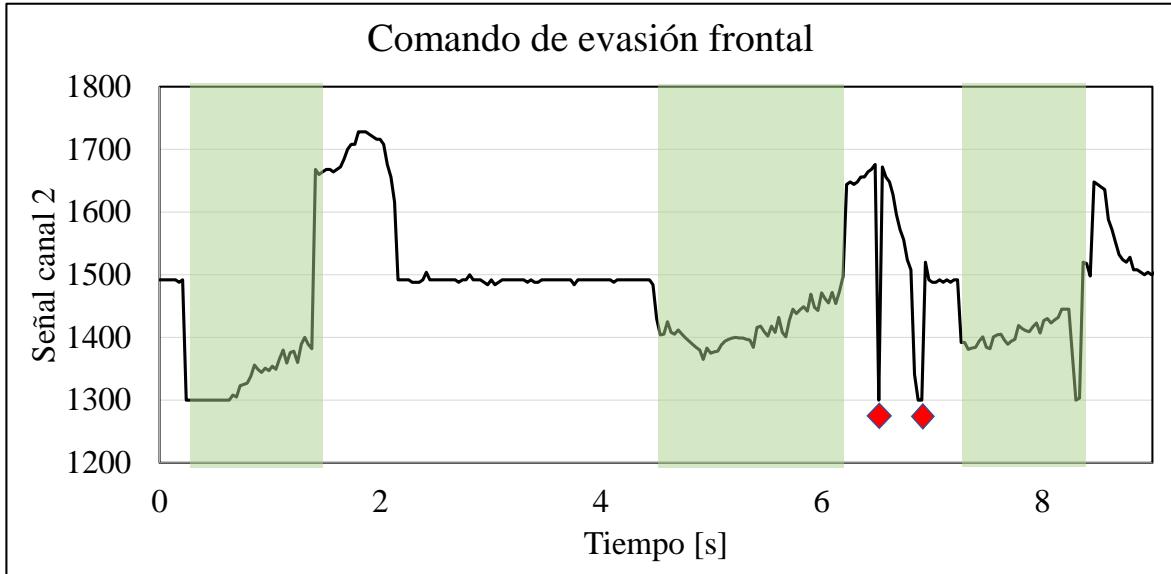
De acuerdo a lo mencionado en la sección 5.2, se define una distancia de seguridad para cada dirección, bajo la cual se activa el CAC. En el caso de direcciones horizontales, esta distancia es de 60 centímetros. Como se puede observar en la figura 6.1, apenas el UAV se encuentra a menos de 60 centímetros de un obstáculo frontal, se anulan los comandos del piloto y el CAC envía comandos en forma de señal PWM por en canal 2 de comunicación, alejando al UAV del obstáculo. Al encontrarse a una distancia mayor a 60 centímetros, el piloto recobra control del UAV. En este caso, el piloto envía comando de moverse hacia adelante, con tal de intentar colisionar nuevamente. El UAV logra evadir el obstáculo frontal exitosamente, acercándose a una velocidad máxima de 1,4 metros por segundo.

La evasión de obstáculos funciona de manera simétrica en el plano horizontal, es decir, funciona de la misma forma la evasión frontal y la evasión lateral, por lo cual se considera suficiente realizar una sola prueba para evasión frontal y lateral.

Para la prueba de evasión superior, se define una distancia de seguridad de 80 centímetros. Se acerca el UAV a un obstáculo posicionado arriba del UAV, con tal de que este entre a la zona de riesgo; luego de lo cual el CAC toma control del UAV, enviando comandos de evasión mediante el canal 3, reduciendo la potencia de los motores para realizar un descenso controlado. Al encontrarse el UAV a más de 80 centímetros del obstáculo, el piloto retoma



(a)



(b)

Figura 6.1: Distancia entre UAV y obstáculo, y comandos enviados al UAV: prueba en una dimensión, dirección frontal.

control del UAV, el cual envía comando de aumentar potencia de los motores, con tal de intentar colisionar nuevamente. El UAV logra evadir el obstáculo exitosamente, con una velocidad de acercamiento de 1,6 metros por segundo.

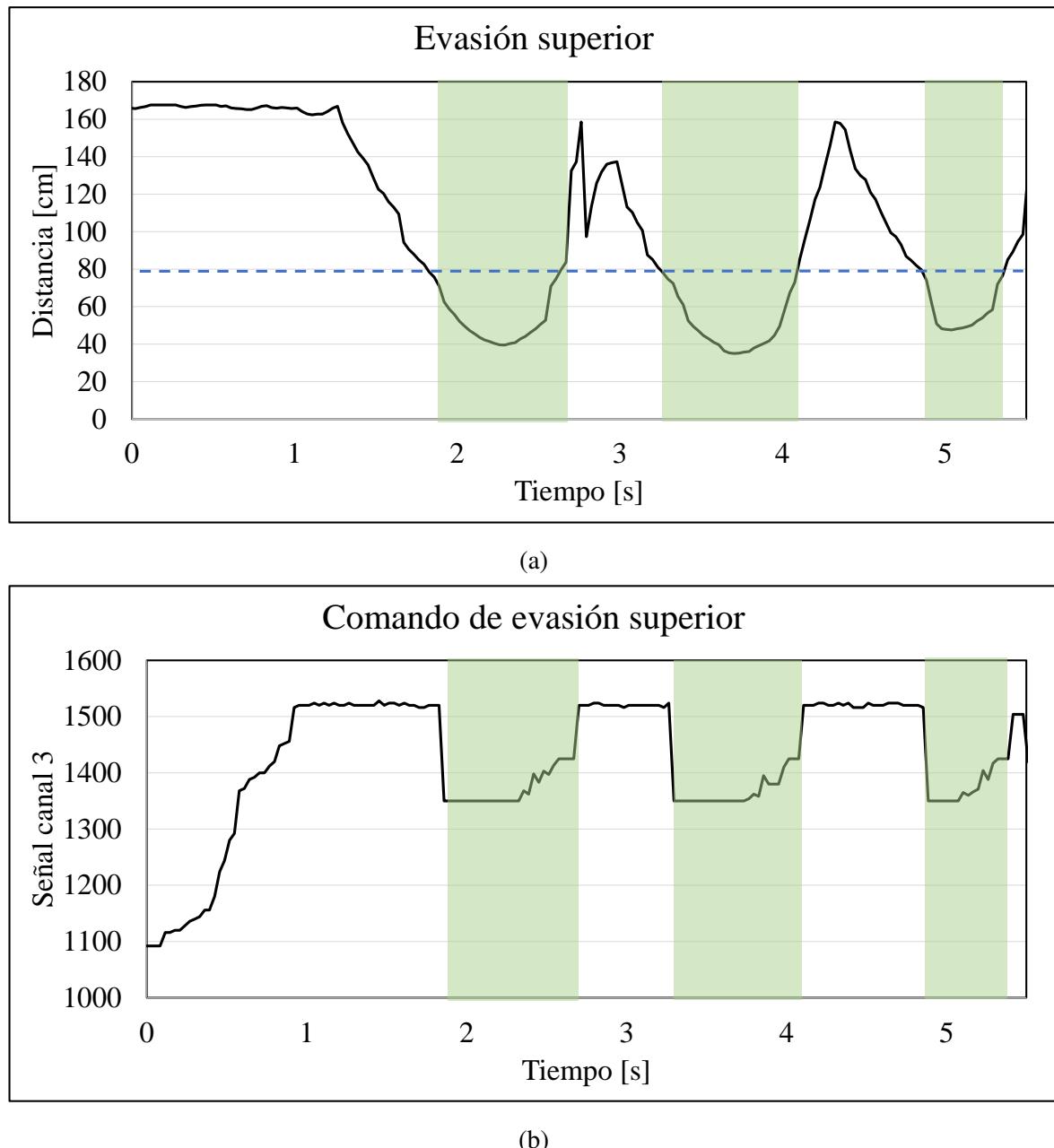


Figura 6.2: Distancia entre UAV y obstáculo, y comandos enviados al UAV: prueba en una dimensión, dirección superior.

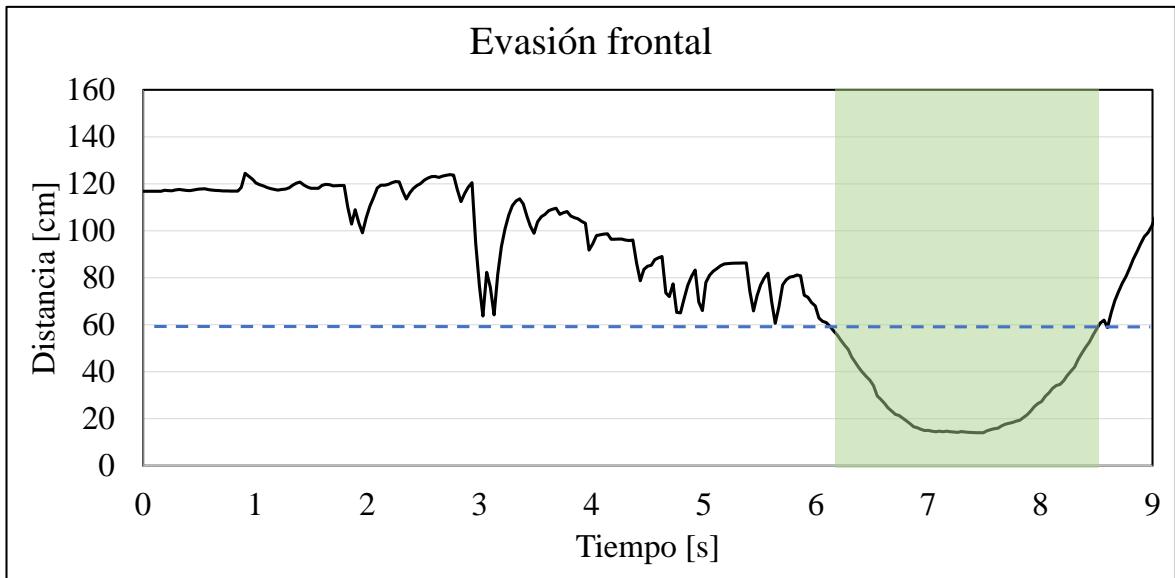
## 6.3 Dos Dimensiones

En esta prueba se busca probar evasión simultánea en dos dimensiones. Para lograr esto, se realiza la prueba a baja velocidad, intentando que se detecten ambas colisiones al mismo tiempo.

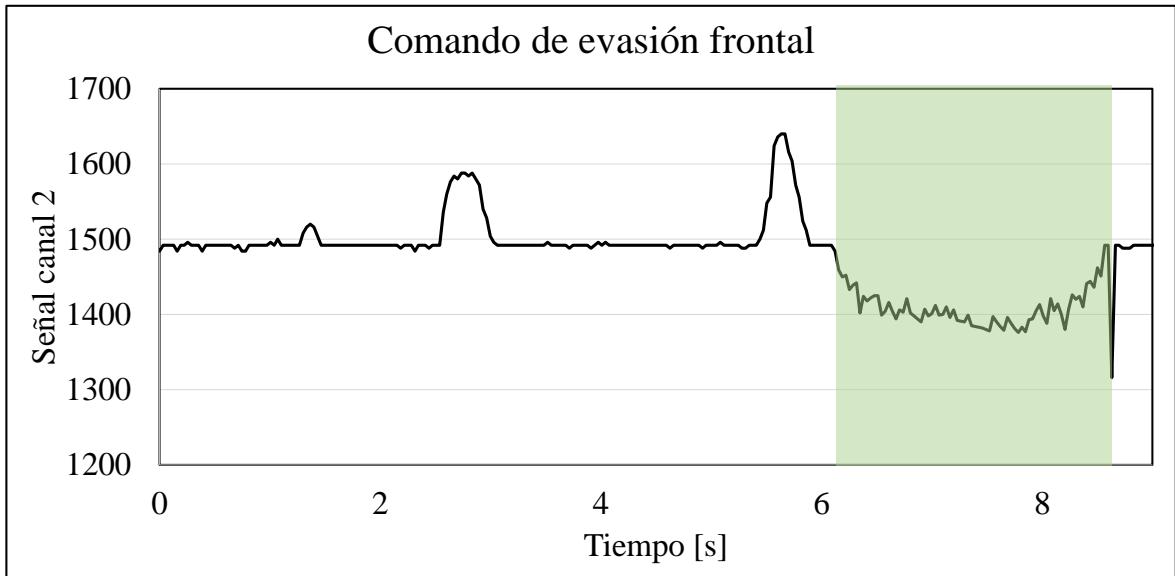
Al detectar ambas colisiones en el mismo ciclo de revisión, se envían los mismos patrones de evasión en una dimensión, pero a las dos direcciones a la vez: para la prueba realizada en el plano horizontal, el gráfico 6.3 muestra los parámetros frontales y el gráfico 6.4 muestra los parámetros laterales. Esto resulta en un patrón de evasión donde el UAV se aleja de los dos obstáculos siguiendo una trayectoria en diagonal; en esta prueba, el UAV sale de ambas zonas de peligro prácticamente al mismo tiempo. El piloto retoma el control del UAV mientras éste se desplaza por inercia, alejándose de la esquina con la que iba a chocar.

La siguiente prueba en dos dimensiones se hace en el plano vertical, es decir, se usa la dirección frontal y la dirección superior de forma simultánea.

Puesto que el patrón de evasión superior involucra quitar potencia de todos los motores, la velocidad con la que el UAV se aleja del obstáculo frontal es menor a las pruebas anteriores. Se pueden ver los parámetros frontales en la figura 6.5, y los parámetros superiores en la figura 6.6.

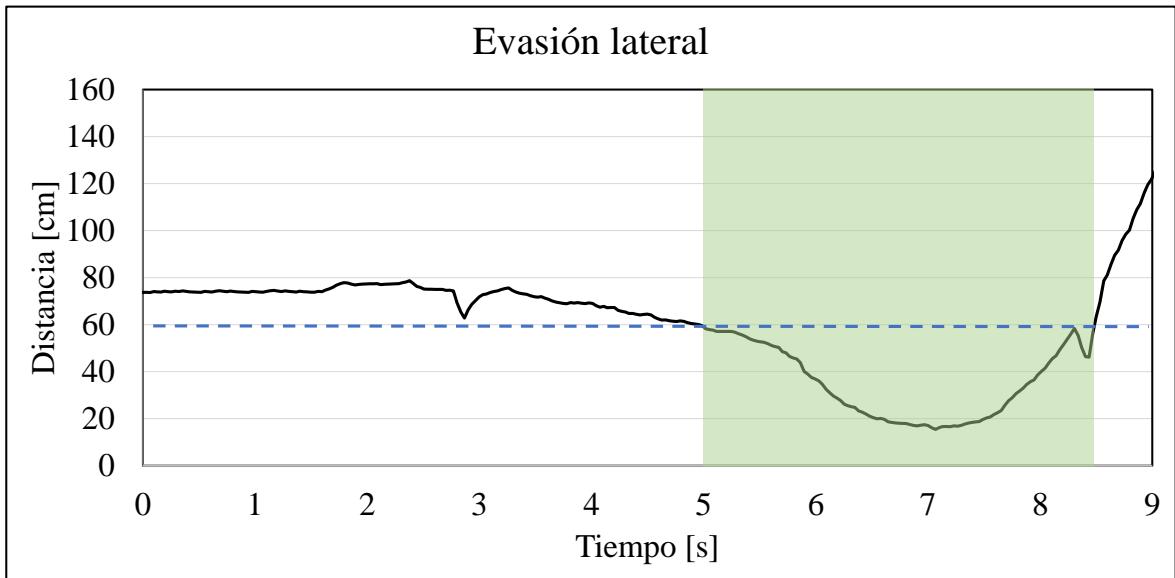


(a)

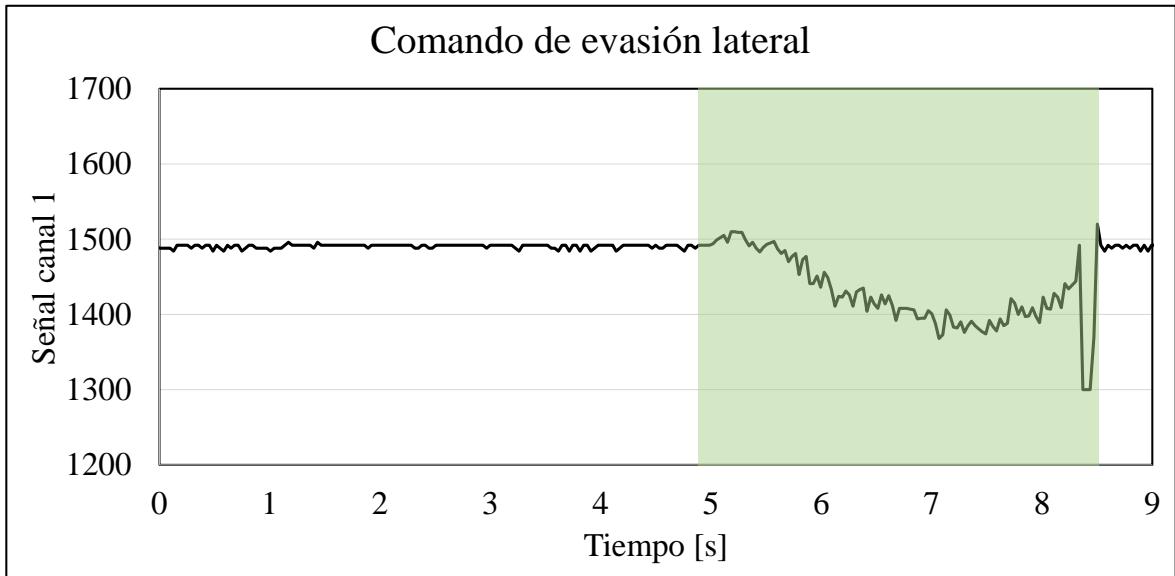


(b)

Figura 6.3: Distancia entre UAV y obstáculo, y comandos enviados al UAV: prueba en dos dimensiones, plano horizontal, dirección frontal.

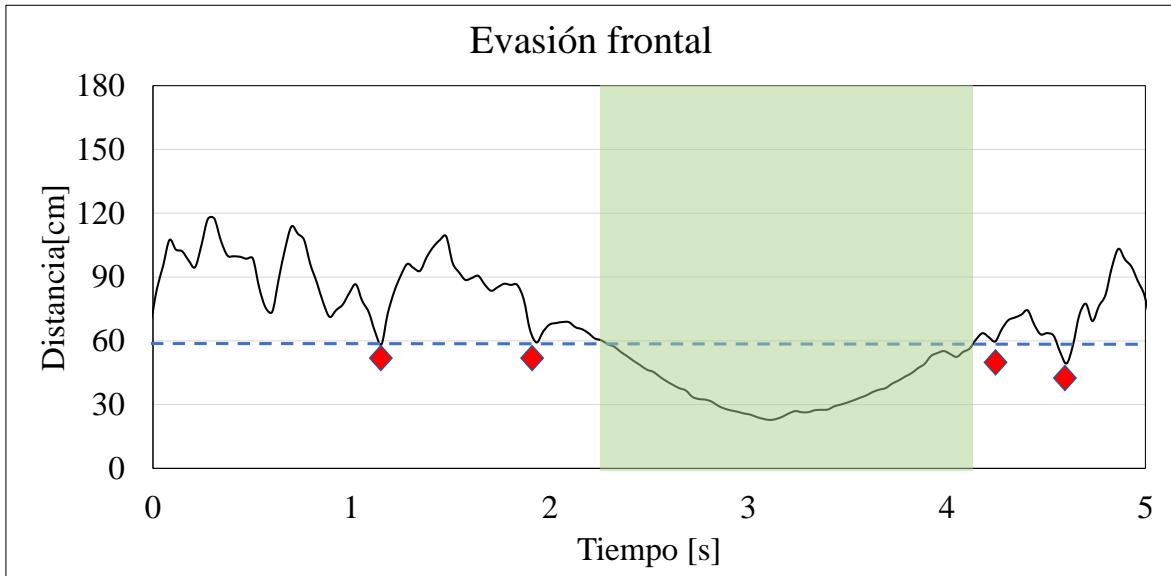


(a)

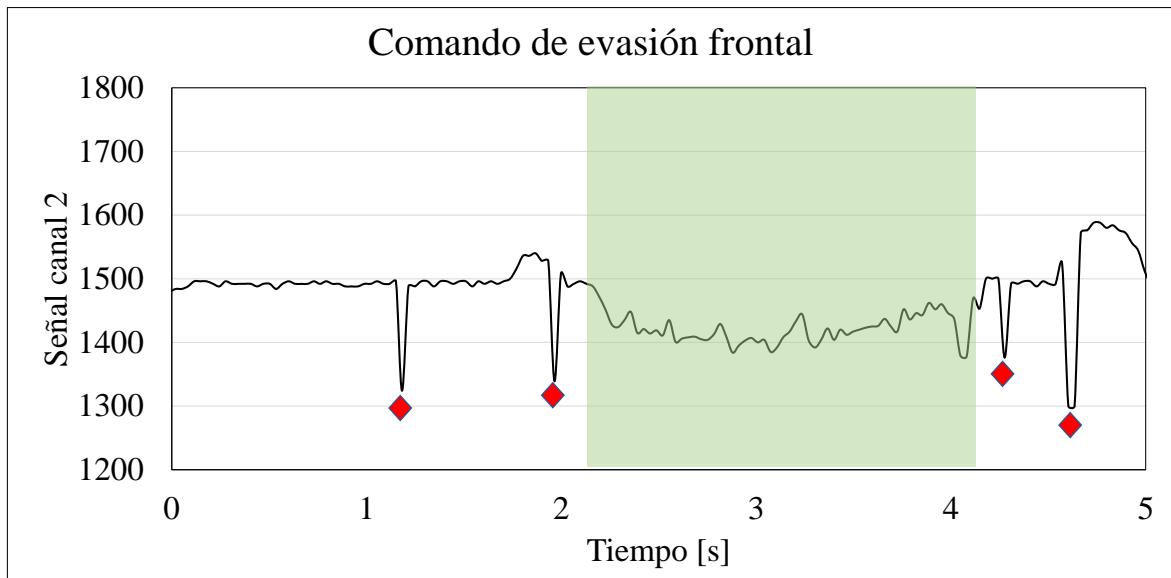


(b)

Figura 6.4: Distancia entre UAV y obstáculo, y comandos enviados al UAV: prueba en dos dimensiones, plano horizontal, dirección lateral.

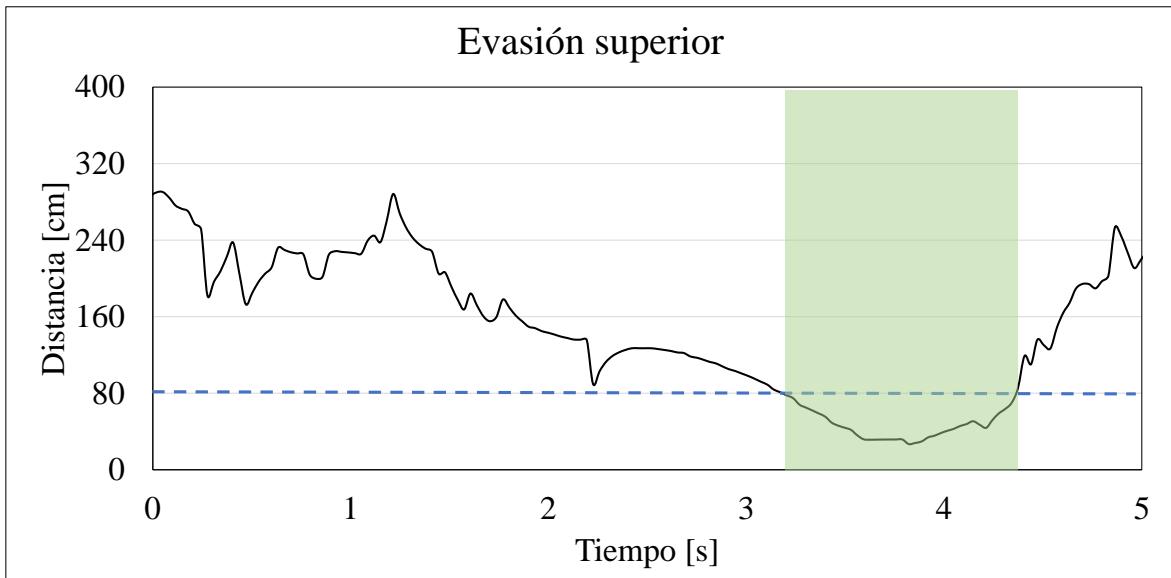


(a)

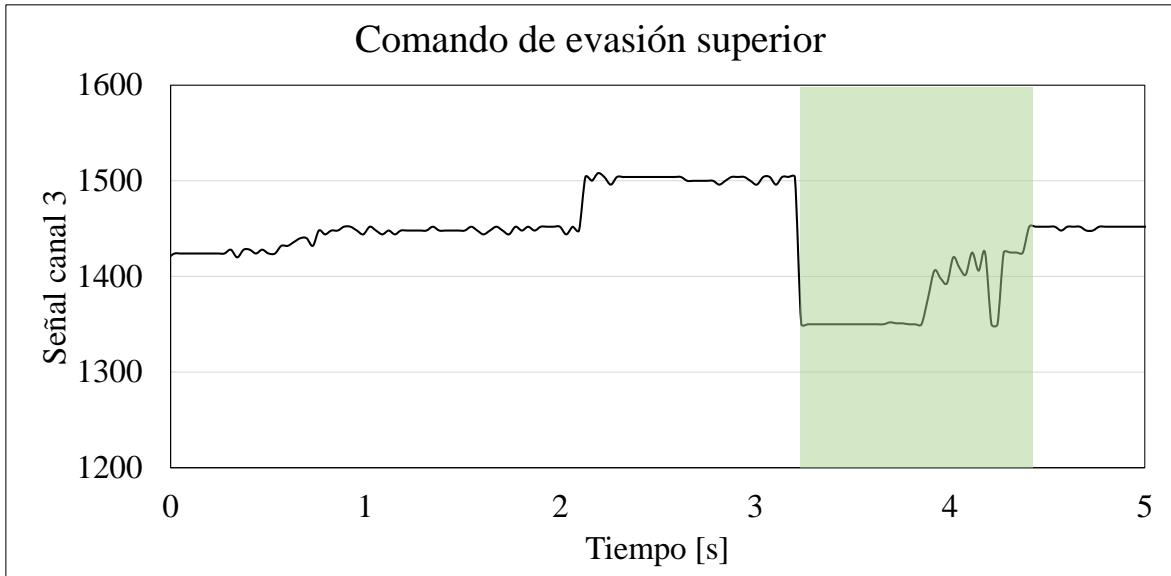


(b)

Figura 6.5: Distancia entre UAV y obstáculo, y comandos enviados al UAV: prueba en dos dimensiones, plano vertical, dirección frontal.



(a)



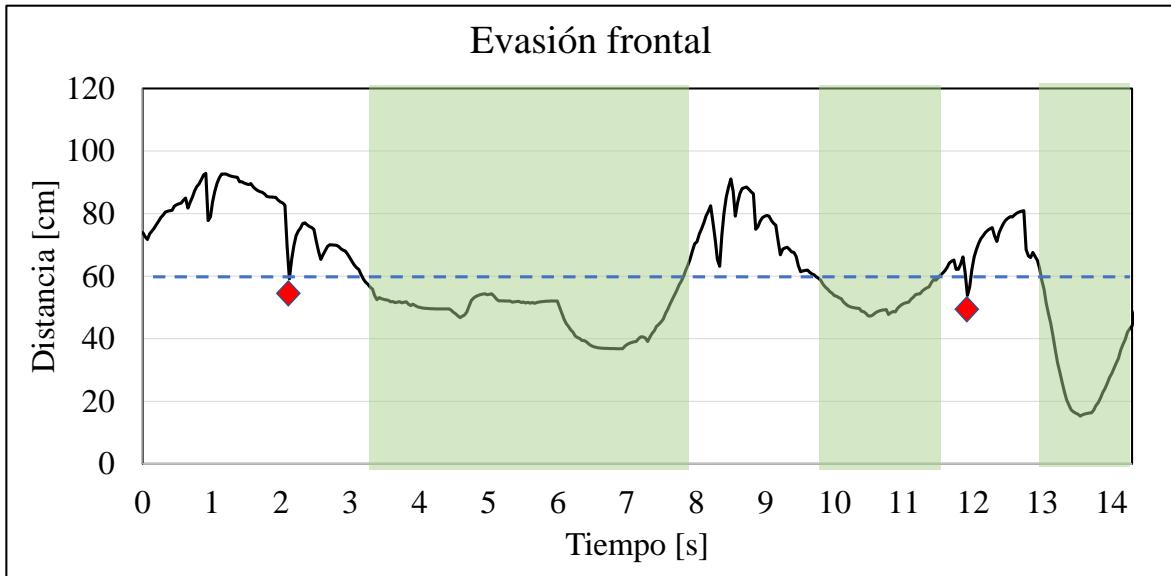
(b)

Figura 6.6: Distancia entre UAV y obstáculo, y comandos enviados al UAV: prueba en dos dimensiones, plano vertical, dirección superior.

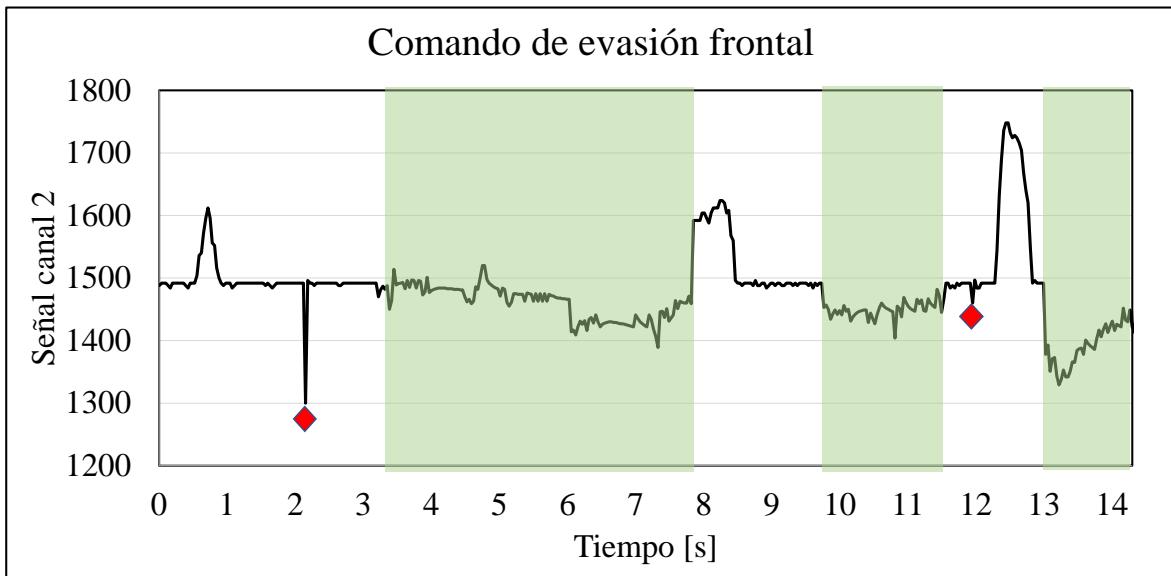
## **6.4 Tres Dimensiones**

Para la evasión en tres dimensiones, no se logra probar colisiones simultáneas en las tres direcciones, no por factibilidad técnica, sino porque al acercarse a los obstáculos se ejecutaban maniobras de evasión al detectar uno o dos obstáculos, no logrando activar los tres de manera simultánea. Por esta razón, se introduce momentáneamente un obstáculo superior para mostrar que el CAC se encuentra activo para las tres direcciones.

Al ingresar el obstáculo superior, el UAV descendió sin afectar su movimiento horizontal. luego de lo cual se acercó el UAV a obstáculos a su izquierda y adelante, los cuales fueron evadidos exitosamente.

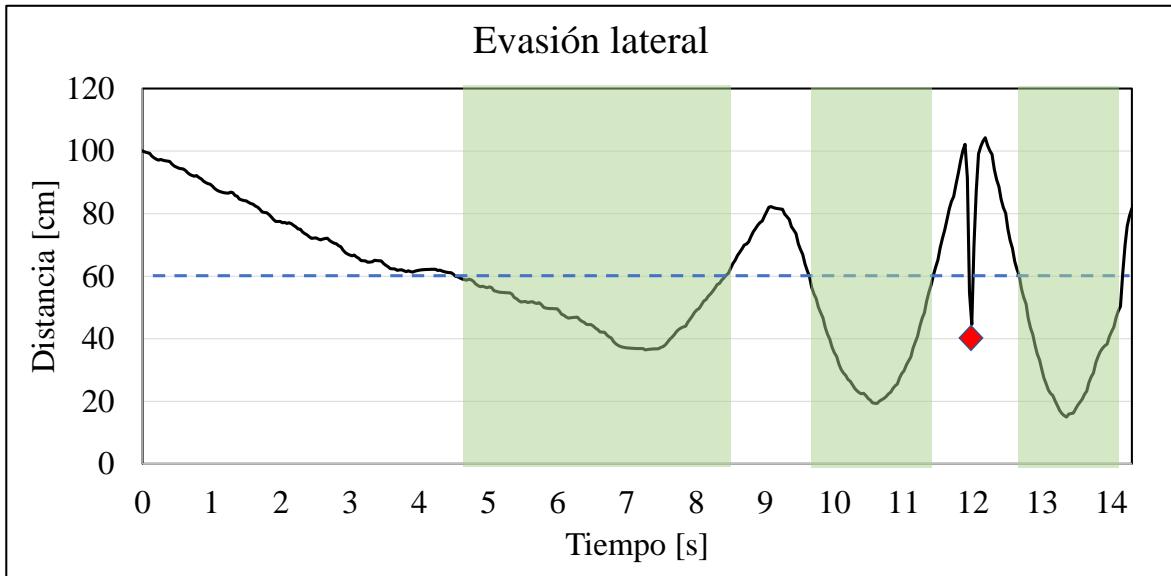


(a)

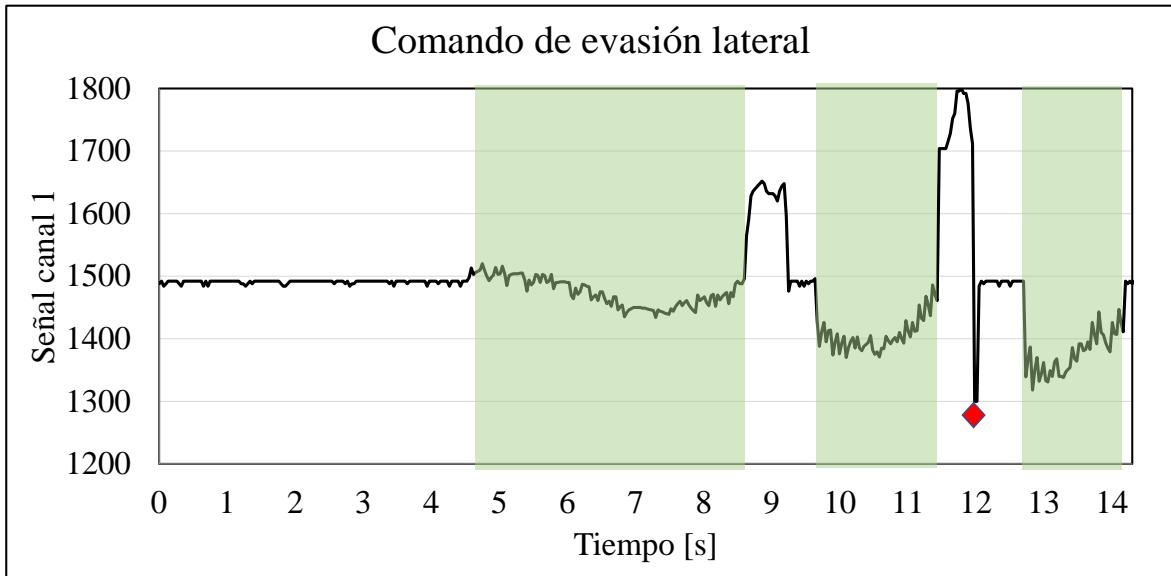


(b)

Figura 6.7: Distancia entre UAV y obstáculo, y comandos enviados al UAV: prueba en tres dimensiones, dirección frontal.

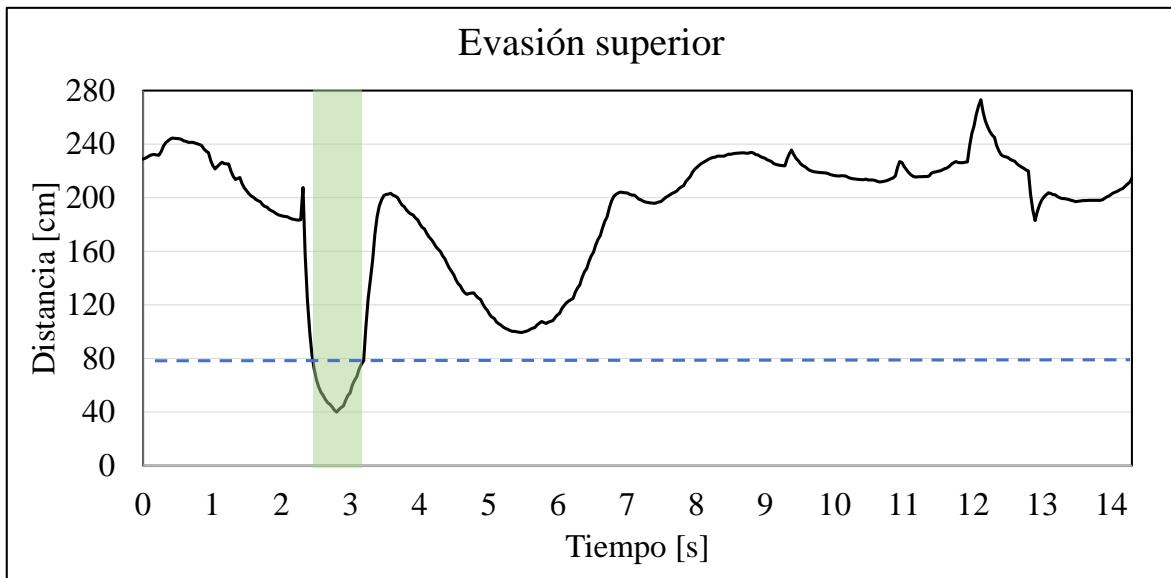


(a)

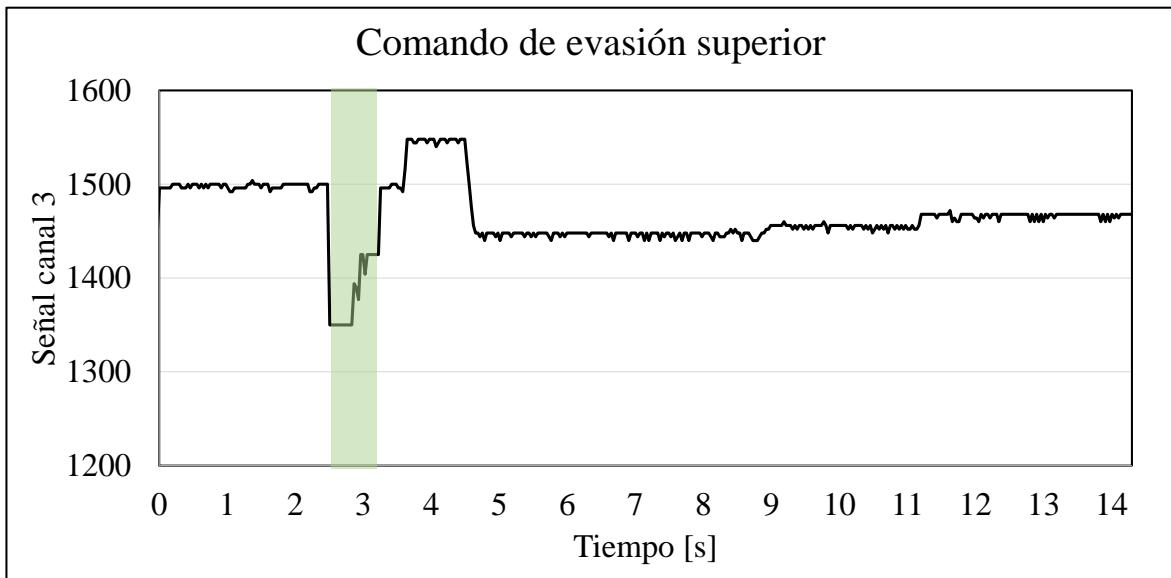


(b)

Figura 6.8: Distancia entre UAV y obstáculo, y comandos enviados al UAV: prueba en tres dimensiones, dirección lateral.



(a)



(b)

Figura 6.9: Distancia entre UAV y obstáculo, y comandos enviados al UAV: prueba en tres dimensiones, dirección superior.

## 6.5 Análisis de Resultados

### 6.5.1 Comparación con Algoritmo en 2 Dimensiones

El trabajo de Rambabu et al. [36] realizó pruebas tanto de obstáculo estático y obstáculo en movimiento. Este evalúa el CAC en una dirección horizontal, asumiendo que el sistema es simétrico, por lo cual debe exhibir un comportamiento equivalente en otra dirección en el plano horizontal. En dicho trabajo, se utilizó como distancia de seguridad 50 centímetros, y se logró una velocidad máxima de acercamiento al obstáculo de 2,5 metros por segundo.

En comparación, en esta memoria se utilizó 60 centímetros como distancia de seguridad para el plano horizontal, y se logró una velocidad máxima de acercamiento al obstáculo de 1,4 metros por segundo. Esto se debe a que, al tener más sensores ultrasónicos en funcionamiento, aumenta el tiempo de ciclo de medición, ya que estos se ejecutan en secuencia para evitar interrumpirse entre ellos.

Se intercambia la velocidad máxima de acercamiento al obstáculo que logre evadir la colisión exitosamente, por tener más direcciones detección y evasión de colisiones. Se realiza un intercambio entre un mayor tiempo de respuesta del CAC por más patrones de evasión, y por ende, más seguridad. Al ser este sistema creado para manejo de UAVs en espacios cerrados, la velocidad de vuelo es más baja en comparación de volar en exteriores.

# Conclusiones

El trabajo realizado en esta memoria tiene varias características que lo distinguen de otros trabajos hechos con vehículos aéreos no tripulados: se enfoca en corregir comandos enviados por un piloto, más que en seguir una ruta predefinida; el sistema está completamente contenido a bordo del UAV, sin necesidad de un computador externo más que para monitoreo; y está pensado especialmente para su uso en interiores, a diferencia de otras aplicaciones donde se cuenta con espacios amplios, y no hay riesgo de chocar contra el techo.

En particular, puesto que este UAV está diseñado para funcionar en espacios cerrados, no resulta preocupante fijar un límite de velocidad de 1,4 metros por segundo, cuando este límite asegura una correcta evasión de colisiones a 60 centímetros de distancia de las paredes, y 80 centímetros de distancia del techo.

Basado en el trabajo de Rambabu et al. [36], en esta memoria se realizó el armado de un vehículo aéreo de cuatro rotORES, con un controlador de vuelo Pixhawk. La pieza central de este trabajo es el microcontrolador Arduino montado en el UAV: a este se conecta el control remoto del piloto, y sensores de distancia que detectan obstáculos. Ambas entradas se combinan de forma de permitir el control manual del UAV hasta que se detecta una colisión; el microcontrolador envía comandos de evasión al controlador de vuelo y devuelve el control al piloto, sin necesidad de dejar de volar, y sin perder el control del aparato.

ConeCTar el controlador de vuelo, el microcontrolador Arduino, los sensores y el control remoto requirió varios meses, probando distintas iteraciones hasta encontrar una distribución óptima. Aun después de conectar todo, se encontraron varias dificultades, entre problemas de conexión al sistema de monitoreo e interferencias producidas por el mismo sistema, que

incluso requirieron descartar una combinación de sensores infrarrojo y ultrasónico, debido a las bajas de voltaje que producía encender los sensores infrarrojos. Detectar estos defectos tomó un tiempo considerable y experiencia en electrónica, agregando un conocimiento interdisciplinario a este trabajo.

La implementación de los filtros de mediana y filtro lineal de Kalman permitieron limpiar las mediciones obtenidas por los sensores ultrasónicos, las cuales sufren de interferencias por parte de varias fuentes diferentes. Usando estas mediciones, se define una distancia segura a obstáculos cercanos, que el UAV intentará mantener.

Los patrones de evasión implementados, que siguen la técnica de ‘campo de repulsión’, evitan los obstáculos con una velocidad que depende de la cercanía del obstáculo al UAV, permitiendo un manejo estable, siempre y cuando la velocidad sea razonable. Esta velocidad depende de la tasa de refresco de los sensores: a mayor tasa de refresco, mayor velocidad puede tener el UAV, aún pudiendo evadir obstáculos que se le presenten. Por otra parte, esta tasa de refresco dependerá de cuántos sensores es necesario revisar para generar un patrón de evasión; consultar cuatro sensores (dos dimensiones) es más rápido que consultar seis (tres dimensiones).

En las pruebas realizadas, después de ajustar los parámetros necesarios a través de ensayo y error, se pudieron evitar colisiones con obstáculos provenientes de hasta tres direcciones simultáneas; el UAV fue capaz de alejarse del obstáculo, sin necesidad de aterrizar, y devolviendo sin dificultades el control al piloto.

Una vez realizadas estas pruebas, y dados los resultados obtenidos, se puede concluir que se ha cumplido el objetivo principal de la memoria, como se mencionó en la sección 1.2: evitar que un UAV colisione en espacios cerrados, por medio de la detección de obstáculos y la toma de control del UAV para evadirlos.

Parte de los objetivos fue que el sistema fuese capaz de detectar obstáculos y tomar acción de forma suficientemente rápida y precisa. El ciclo de tomar mediciones de tres sensores de forma secuencial, filtrar dichas mediciones y determinar un patrón de evasión permite al UAV circular a una velocidad máxima de 1,4 metros por segundo, asegurando que el UAV podrá evadir obstáculos que se le presenten en cualquier dirección. Dada esta velocidad máxima de

circulación, la cual se determina de forma empírica, se puede concluir que el sistema realiza los cálculos a una velocidad lo suficientemente alta para resultar aplicable a un contexto real, de circulación en espacios cerrados.

Esta velocidad de circulación es reducida en comparación al trabajo base, de 2,5 metros por segundo; como aquel algoritmo de detección revisa menos sensores, es posible revisarlos más rápidamente y circular más rápido. Se puede considerar la velocidad reducida del sistema implementado como compensación por una capacidad más robusta de detección y evasión de colisiones.

En general, este trabajo complementa un área de la robótica que aún no está bien investigada: el sistema implementado permitirá usar UAVs en espacios cerrados, minimizando el riesgo de daños millonarios causados por colisiones; y si bien los algoritmos necesarios ya existen, es sólo recientemente que la miniaturización de los UAV ha permitido la implementación de los mismos en sistemas autónomos, que no requieran de procesamiento externo. En ese sentido, se espera que el presente trabajo habilite nuevas e interesantes aplicaciones para los vehículos aéreos, en lugares que antes no era posible acceder.

## Trabajo futuro

### Patrones de Detección de Colisiones

En el presente trabajo se probaron seis patrones de detección de colisiones, usando tres sensores. Si bien quedó demostrado que los patrones probados pueden extrapolarse por simetría, es necesario implementar completamente los 26 patrones en total que se pueden derivar de los seis sensores de distancia funcionando de forma simultánea.

Programar tales patrones incluiría situaciones que este trabajo no cubre. El ejemplo más notorio es el comportamiento que debe tener el UAV al detectarse colisiones en direcciones opuestas: el comportamiento intuitivo a implementar sería mantener la misma distancia entre ambos obstáculos, sin embargo, la implementación de ese comportamiento no necesariamente va a ser una extensión de los patrones ya programados, y no necesariamente va a poder

aprovechar la simetría de los sensores.

## **Refinamiento de parámetros**

Se puede seguir iterando los parametros tanto del filtro lineal de Kalman como los del controlador de posición PID, para mejorar la fluidez y control de las maniobras de evasión. Encontrar los parámetros que producirían el mejor resultado no es una ciencia exacta, y requiere un proceso exhaustivo de ensayo y error.

Adicionalmente, se podría realizar la misma calibración de parámetros para el controlador de vuelo. El Pixhawk incluye un controlador PID, el cual también es completamente configurable. Al ajustar los parámetros del Pixhawk, se lograría una mejora en el control del UAV, por parte del piloto y en la ejecución de las maniobras evasivas.

## **Detección de Colisiones utilizando distintos sensores**

Dado el aumento en la capacidad de procesamiento de los microcontroladores, y el surgimiento de computadores de propósito general en formato tarjeta, no es descabellado pensar que en un futuro cercano se podrán aplicar técnicas de detección de colisiones que en este trabajo se descartaron por su alto coste computacional.

Un ejemplo de tales técnicas sería ocupar cámaras tridimensionales para realizar la detección de colisiones. El uso de cámaras implica procesamiento de imágenes, técnica que a día de hoy es usada casi exclusivamente con computadores externos; con el hardware adecuado, estas técnicas podrán empezar a implementarse directamente en los controladores a bordo del UAV.

Otro sensor que requiere altas cantidades de procesamiento es el LIDAR, que ocupa sensores y emisores láser para crear mapas tridimensionales de alta resolución. Un sensor de esas características permitiría realizar mediciones mucho más exactas para detectar colisiones.

# Bibliografía

- [1] 3DR: *Community*. <https://3dr.com/about/community/>.
- [2] 3DR Store: *3dr pixhawk*. <https://pixhawk.org/modules/pixhawk>.
- [3] Alexopoulos, A., A. Kandil, P. Orzechowski y E. Badreddin: *A comparative study of collision avoidance techniques for unmanned aerial vehicles*. En *2013 IEEE International Conference on Systems, Man, and Cybernetics*, páginas 1969–1974, Oct 2013.
- [4] Arduino: *Arduino mega*. <https://www.arduino.cc/en/Main/arduinoBoardMega>, 2017.
- [5] ArduPilot: *Pixhawk overview*. <http://ardupilot.org/copter/docs/common-pixhawk-overview.html>, 2016.
- [6] ArduPilot Dev Team: *Mission planner overview*. <http://ardupilot.org/planner/docs/mission-planner-overview.html>, 2016.
- [7] ArduPilot Dev Team: *Dji f330 flamewheel quadcopter assembly*. <http://ardupilot.org/copter/docs/dji-f330-flamewheel.html>, 2017. Assemby of F330 is identical to assembly of F450.
- [8] ArduPilot Dev Team: *Rc input channel mapping (rcmap)*. <http://ardupilot.org/copter/docs/common-rcmap.html>, 2017.
- [9] Ascending Technologies: *Droneservice – uas indoor surveillance*. <http://www.asctec.de/en/droneservice-uas-indoor-surveillance/>, Diciembre 2015.
- [10] Baumer: *Operation and design of ultrasonic sensors*. <http://www.baumer.com/be-en/services/user-knowledge/ultrasonic-sensors/functionality/>.
- [11] Beauregard, Brett: *Arduino pid library*. <http://playground.arduino.cc/Code/PIDLibrary>, 2017.
- [12] Beauregard, Brett: *Introducing proportional on measurement*. <http://brettbeauregard.com/blog/2017/06/introducing-proportional-on-measurement/>, 2017.

- [13] Bills, C., J. Chen y A. Saxena: *Autonomous mav flight in indoor environments using single image perspective cues*. En *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, páginas 5776–5783, May 2011.
- [14] Chee, K.Y. y Z.W. Zhong: *Control, navigation and collision avoidance for an unmanned aerial vehicle*. Sensors and Actuators A: Physical, 190(Supplement C):66 – 76, 2013, ISSN 0924-4247. <http://www.sciencedirect.com/science/article/pii/S092442471200698X>.
- [15] Cheong, M K, M R Bahiki y S Azrad: *Development of collision avoidance system for useful uav applications using image sensors with laser transmitter*. IOP Conference Series: Materials Science and Engineering, 152(1):012026, 2016. <http://stacks.iop.org/1757-899X/152/i=1/a=012026>.
- [16] Chowdhary, Girish, John Ottander, Erwan Salaun y Eric N. Johnson: *Low cost guidance, navigation, and control solutions for a miniature air vehicle in gps denied environments*. American Institute of Aeronautics and Astronautics, jan 2009.
- [17] Digi: *XBee Multipoint RF Modules*. Digi International, 2011.
- [18] DJI: *FlameWheel F450*. DJI, 2.2 edición, Mayo 2015. [http://dl.djicdn.com/downloads/flamewheel/en/F450\\_User\\_Manual\\_v2.2\\_en.pdf](http://dl.djicdn.com/downloads/flamewheel/en/F450_User_Manual_v2.2_en.pdf).
- [19] Faragher, Ramsey: *Understanding the basis of the kalman filter via a simple and intuitive derivation*. IEEE Signal Processing Magazine, 2012.
- [20] Hart, P. E., N. J. Nilsson y B. Raphael: *A formal basis for the heuristic determination of minimum cost paths*. IEEE Transactions on Systems Science and Cybernetics, 4(2):100–107, July 1968, ISSN 0536-1567.
- [21] Hrabar, S.: *3d path planning and stereo-based obstacle avoidance for rotorcraft uavs*. En *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, páginas 807–814, Sept 2008.
- [22] Jurić, Darko: *Object tracking: Kalman filter with ease*. <https://www.codeproject.com/Articles/865935/Object-Tracking-Kalman-Filter-with-Ease>, Enero 2015.
- [23] Kadammatt, V.: *Flying a drone indoors: Considerations & tips*. <http://www.droneybee.com/flying-a-drone-indoors/>, 2017.
- [24] Kaya, Y. y S. Yamamura: *A self-adaptive system with a variable-parameter pid controller*. Transactions of the American Institute of Electrical Engineers, Part II: Applications and Industry, 80(6):378–386, Jan 1962, ISSN 0097-2185.
- [25] Khatib, O.: *Real-time obstacle avoidance for manipulators and mobile robots*. En *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, volumen 2, páginas 500–505, Mar 1985.

- [26] Krajník, Tomáš, Vojtěch Vonásek, Daniel Fišer y Jan Faigl: *AR-Drone as a Platform for Robotic Research and Education*. En *Research and Education in Robotics: EUROBOT 2011*, Heidelberg, 2011. Springer.
- [27] Lopez, Alberto: *How to read rc radio signals with arduino using ppm signal*. <http://micsircuitos.com/how-to-read-rc-radio-signals-with-arduino-using-ppm-signal-tutorials/>, 2011.
- [28] Lumb, David: *Mit researchers use drone fleets to track warehouse inventory*. <https://www.engadget.com/2017/08/25/mit-drone-fleets-track-warehouse-inventory/>, Agosto 2017.
- [29] Maxbotix Inc.: *LV-MaxSonar-EZ Series*. Maxbotix Inc., 2015. [https://www.maxbotix.com/documents/LV-MaxSonar-EZ\\_Datasheet.pdf](https://www.maxbotix.com/documents/LV-MaxSonar-EZ_Datasheet.pdf).
- [30] Maxbotix Inc.: *067 – maxsonar operation on a multi-copter*. <https://www.maxbotix.com/tutorials7/067-maxsonar-operation-on-a-multi-copter.htm>, 2017.
- [31] Muller, J., A. V. Ruiz y I. Wieser: *Safe & sound: A robust collision avoidance layer for aerial robots based on acoustic sensors*. En *2014 IEEE/ION Position, Location and Navigation Symposium - PLANS 2014*, páginas 1197–1202, May 2014.
- [32] Pai, Radhesh: *Pid controller simplified*. <https://radhesh.wordpress.com/2008/05/11/pid-controller-simplified/>, Mayo 2008.
- [33] Park, J. W., H. D. Oh y M. J. Tahk: *Uav collision avoidance based on geometric approach*. En *SICE Annual Conference, 2008*, páginas 2122–2126, Aug 2008.
- [34] Parrot SA.: *Products*. <http://developer.parrot.com/products.html>, 2016.
- [35] Rethnaraj Rambabu, Muhammad Rijaluddin Bahiki, Syaril Azrad: *Multi-sensor fusion based uav collision avoidance system*. Jurnal Teknologi (Sciences & Engineering), Mayo 2015.
- [36] Rethnaraj Rambabu, Muhammad Rijaluddin Bahiki, Syaril Azrad: *Relative position-based collision avoidance system for swarming uavs using multi-sensor fusion*. Journal of Engineering and Applied Sciences 10(21), nov 2015.
- [37] SHARP Corporation: *SHARP GP2Y0A21YK0F*. SHARP Corporation, Diciembre 2006. [http://www.mouser.com/ds/2/321/gp2y0a21yk\\_e-3493.pdf](http://www.mouser.com/ds/2/321/gp2y0a21yk_e-3493.pdf).
- [38] Szczyz, Mike: *Adding node.js based sensors to the parrot ar drone*. <http://hackaday.com/2012/11/28/adding-node-js-based-sensors-to-the-parrot-ar-drone/>.
- [39] Özderya, Hasan Yavuz: *Serialplot - realtime plotting software*. <https://hackaday.io/project/5334-serialplot-realtime-plotting-software>, 2017.

# Apéndice A

## Diagramas

Aquí se incluyen todos los diagramas de cableo y ensamblaje del UAV para futura referencia. Se incluye en los diagramas los sensores infrarrojos de tal forma que a futuro alguien tenga la opción de corregirlo e implementarlo.

En la figura A.1 se muestran las conexiones físicas e inalámbricas de forma simplificada.

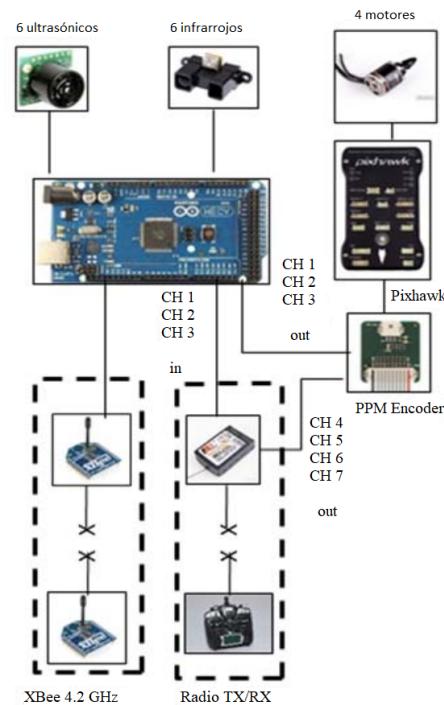


Figura A.1: Configuración del hardware a alto nivel.

Los sensores ultrasónicos deben ejecutarse en secuencia para evitar interferencia entre ellos mismos. En la figura A.2 se muestra la implementación para dos sensores ultrasónicos, donde un sensor gatilla al siguiente sensor. Este diagrama se puede extender a cualquier cantidad de sensores ultrasónicos que sean necesarios.

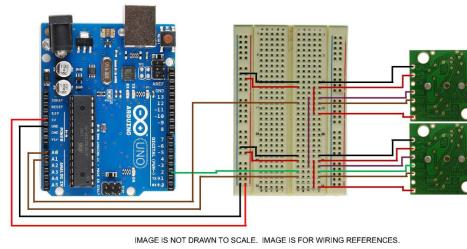


Figura A.2: Encadenamiento de sensores ultrasónicos.

Los sensores infrarrojos generan pulsos infrarrojos cada milisegundo, lo cual genera una baja de voltaje. La implementación probada para este trabajo es la que se puede observar en la figura A.3. El problema con esta implememtación es que, al bajar el voltaje en la linea, esto afecta las mediciones de los otros sensores. Una solución sería armar un sistema en que los sensores se alimenten de una fuente aparte a los cables de señal analógica o digital.

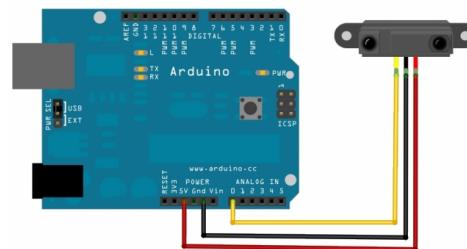


Figura A.3: Conexión analógica de sensor infrarrojo a Arduino.