



INAOE

Vuelo autónomo basado en estimadores estocásticos para mantener a un objetivo dentro del campo de visión

por

Ing. Beatriz Hernández Hernández

Tesis sometida como requisito parcial para obtener el
grado de

**MAESTRO EN CIENCIAS EN LA ESPECIALIDAD
DE ELECTRÓNICA**

en el

Instituto Nacional de Astrofísica, Óptica y Electrónica

Agosto 2017

Tonantzintla, Puebla

Supervisada por:

Dr. José de Jesús Rangel Magdaleno

Dr. José Martínez Carranza

©INAOE 2017

El autor otorga al INAOE el permiso de reproducir y distribuir copias en su
totalidad o en partes de esta tesis



**Vuelo autónomo basado en estimadores
estocásticos para mantener a un objetivo dentro
del campo de visión**

Tesis de Maestría

POR:

Ing. Beatriz Hernández Hernández

ASESORES:

Dr. José de Jesús Rangel Magdaleno

Dr. José Martínez Carranza

Instituto Nacional de Astrofísica, Óptica y Electrónica
Coordinación de Electrónica

*Este trabajo está dedicado a mis padres,
ya que con su amor, esfuerzo y dedicación
me mostraron el camino de la superación
y perseverancia para poder cumplir mis
objetivos.*

Agradecimientos

Antes que nada, a mis padres y a mis hermanos por su amor incondicional y darme la motivación que necesitaba para salir adelante.

A mis asesores de Tesis, el Dr. José de Jesús Rangel Magdaleno y el Dr. José Martínez Carranza ya que sin sus ideas no hubiera sido posible desarrollar este trabajo.

A Leticia Oyuki Rojas Perez, Aldrich Alfredo Cabrera Ponce y Roberto Munguia Silva por el apoyo prestado para la realización de esta tesis.

Al Almirante Octavio Trejo Hermida, por la confianza brindada para la realización de esta tesis en el INAOE.

A la Secretaría de Marina - Armada de México por la flexibilidad brindada para realizar este trabajo.

Al INAOE por las facilidades y los servicios otorgados para la realización de este proyecto.

Al Consejo Nacional de Ciencia y Tecnología por la beca otorgada para realizar mis estudios y poder llevar a cabo esta investigación.

Finalmente, agradezco a mis sinodales por el tiempo y la dedicación prestadas para la revisión de esta tesis.

Resumen

El uso de los drones en áreas como el cine, deportes, eventos sociales e incluso en áreas militares ha incrementado debido a su flexibilidad para capturar video en escenarios donde hay un gran interés por mantener a un objetivo dentro del campo de visión de la cámara a bordo del dron. Con el fin de eliminar la dependencia del piloto que controla al dron, en este trabajo se presenta un sistema para el control del vuelo autónomo del dron con el propósito de mantener a un objetivo dentro del campo de visión de la cámara que tiene a bordo. Las imágenes fueron capturadas por la cámara a bordo del dron y combinadas con un estimador estocástico de los estados del objetivo, basado en el Filtro Kalman Unscented, para generar comandos de control que permitan al dron realizar la grabación de forma autónoma. Se proponen dos enfoques, el primero con el objetivo visto desde el frente del dron y el segundo con el objetivo visto desde arriba por el dron. Ambos enfoques fueron validados con pruebas en tiempo real que involucran a un objetivo que se mueve en diferentes trayectorias y son comparadas con pilotos humanos. Para el primer enfoque, nuestro sistema mantiene al objetivo dentro del campo de visión con un 96.60% de éxito y para el segundo enfoque con un 92.05%, mientras que el éxito obtenido con pilotos humanos es del 83.30% y del 89.13% respectivamente. Lo anterior, indica que nuestro enfoque tiene el potencial para ser usado en aplicaciones en las que los drones autónomos pueden ser utilizados para grabación aérea de video, con un especial interés por mantener a un objetivo dentro del campo de visión de la cámara a bordo del dron.

Abstract

The use of drones in areas such as cinema, sports, social events and even military areas has been increasing due to their flexibility to capture video in scenarios where there is an interest to keep a target within the field of view of the drones onboard camera. In order to remove the dependence of the pilot that controls the drone, in this work we present a system for an autonomous flight control of the drone with the goal of keeping a target within the field of view of its onboard camera. For the latter, the images were captured by a camera onboard the drone, whose output was combined with a stochastic estimator of the target states, based on the Unscented Kalman Filter, to generate control commands so that the drone performs such recording autonomously. Two approaches are proposed, the first with the target seen from the front of the drone and the second with the target seen from above by the drone. Both approaches of the system were validated with real-time tests involving different targets moving with different trajectories and compared against human pilots. For the first approach, our system kept the target within the field of view with a 96.60 % of success and the second approach with a 92.05 %, compared to 83.30 % and 89.13 % of success obtained by human pilots, respectively. The latter indicates that our approach has the potential to be used in applications where autonomous drones could be used for aerial video recording, with a special interest in keeping a target within the field of view of the drone's camera.

Tabla de Contenido

Agradecimientos	I
Resumen	III
Abstract	V
Índice de Figuras	XI
Índice de Tablas	XV
1. Introducción	1
1.1. Descripción del problema	2
1.2. Justificación	3
1.3. Objetivos	3
1.3.1. Objetivo general	3
1.3.2. Objetivos específicos	3
1.4. Estructura de la tesis	4
2. Trabajo relacionado	5
3. Marco teórico	13
3.1. Quadrotor	13
3.2. ROS	14
3.2.1. Conceptos básicos	15
3.3. Procesamiento de imagen	16
3.3.1. Modelo de color	17
3.3.2. Imagen RGB	18

3.3.3.	Binarización	18
3.3.4.	Operaciones morfológicas	19
3.3.5.	Centroide	21
3.4.	Estimadores de estados estocásticos	22
3.4.1.	Filtro Kalman (KF)	22
3.4.2.	Filtro Kalman Extendido (EKF)	23
3.4.3.	Filtro Kalman Unscented (UKF)	24
4.	Metodología	27
4.1.	Adquisición y procesamiento de imagen	27
4.1.1.	Adquisición de la imagen	28
4.1.2.	Procesamiento de la imagen	29
4.2.	Implementación de estimadores estocásticos	31
4.2.1.	Implementación del KF	31
4.2.2.	Implementación del EKF y el UKF	32
4.2.3.	Selección de estimador estocástico	33
4.3.	Control Reactivo	33
4.3.1.	Primer enfoque: Objetivo visto de frente por el dron	34
4.3.2.	Segundo enfoque: Objetivo visto desde arriba por el dron	37
5.	Pruebas experimentales y resultados	39
5.1.	Resultados de la implementación del KF, EKF y UKF	39
5.1.1.	Resultados de la selección de estimador estocástico	41
5.2.	Resultados para objetivo visto de frente	44
5.2.1.	Comparación entre el vuelo autónomo implementado y el vuelo a control remoto realizado por un usuario con el objetivo visto de frente	45
5.3.	Resultados para objetivo visto desde arriba	46
5.3.1.	Comparación entre el vuelo autónomo implementado y el vuelo a control remoto realizado por un usuario con el objetivo visto desde arriba	48
5.4.	Sumario	49
6.	Conclusiones	51
6.1.	Limitaciones	52

6.2. Trabajo a futuro	52
Apéndice	
Productos obtenidos	53
Bibliografía	61

Índice de Figuras

1.1. Aplicaciones de los VANT's.	1
2.1. Segmentación de imagen basada en color	5
2.2. Resultados experimentales del seguimiento de un objetivo.	6
2.3. Diagrama de flujo del Controlador para el seguimiento de objetivos.	7
2.4. Simulación de respuesta del KF y el EKF.	8
2.5. Comparación de las trayectorias en XY del UKF y EKF.	9
2.6. Trayectorias para evaluación de desempeño de los filtros.	10
2.7. Tiempo promedio de ejecución del EKF y UKF.	11
2.8. Error RMS promedio de las posiciones	11
3.1. Movimientos básicos del quadrotor	14
3.2. Conceptos básicos de ROS	15
3.3. Representación del espacio de color RGB.	17
3.4. Componentes de una imagen de color RGB.	18
3.5. Binarización de una imagen. Izquierda:imagen original, derecha:imagen binarizada	19
3.6. Ejemplo de elemento estructurante.	19
3.7. Dilatación de una imagen.	20
3.8. Erosión de una imagen.	21
3.9. Representación del centroide de una imagen.	21
4.1. Quadrotor Parrot Bebop 2 utilizado para la implementación del trabajo propuesto.	27
4.2. Red ROS para la adquisición de imagen.	28
4.3. Imágenes recibidas en el espacio RGB.	29

4.4. Ejemplo de resta de canales de una imagen.	30
4.5. Resta de componentes de la imagen adquirida por el dron.	30
4.6. Centroides del objetivo.	30
4.7. Control de lazo abierto (control reactivo) para movimientos en roll del dron.	34
4.8. Límites del campo de visión de la cámara a bordo del dron para que el objetivo sea visto de frente	34
4.9. Diagrama de flujo del sistema para mantener a un objetivo dentro del campo de visión de la cámara a bordo del dron visto desde el frente.	35
4.10. Posición futura estimada del objetivo.	36
4.11. Control de lazo abierto (control reactivo) para movimientos en roll y pitch del dron.	37
4.12. Límites del campo de visión de la cámara a bordo del dron para que el objetivo sea visto desde arriba	37
4.13. Diagrama de flujo del sistema para mantener a un objetivo dentro del campo de visión de la cámara a bordo del dron visto desde arriba.	38
5.1. Predicción y Corrección del KF comparadas con la medida real de la posición del objetivo.	40
5.2. Predicción y Corrección del EKF comparadas con la medida real de la posición del objetivo.	40
5.3. Predicción y Corrección del UKF comparadas con la medida real de la posición del objetivo.	41
5.4. Comparación entre el KF, EKF y UKF con primer recorrido de prueba del objetivo.	42
5.5. Comparación entre el KF, EKF y UKF con segundo recorrido de prueba del objetivo	42
5.6. Comparación entre el KF, EKF y UKF con tercer recorrido de prueba del objetivo	43
5.7. Respuesta del primer enfoque propuesto observada desde el dron cuando el objetivo está en diferentes posiciones.	44
5.8. Respuesta del dron al primer enfoque propuesto observada desde un punto externo cuando el objetivo está en diferentes posiciones.	44
5.9. Trayectorias de referencia para evaluar el primer enfoque propuesto.	45

5.10. Respuesta del segundo enfoque propuesto observada desde el dron cuando el objetivo está en diferentes posiciones.	47
5.11. Trayectorias de referencia para evaluar el segundo enfoque propuesto.	48

Índice de Tablas

2.1. Comparación de los errores RMSE en los filtros EKF y UKF.	10
5.1. RMSE de la respuesta del KF.	40
5.2. RMSE de la respuesta del EKF.	40
5.3. RMSE de la respuesta del UKF.	41
5.4. RMSE para el KF, EKF y UKF con primer recorrido del objetivo. . .	42
5.5. RMSE para el KF, EKF y UKF con segundo recorrido del objetivo. . .	42
5.6. RMSE para el KF, EKF y UKF con tercer recorrido del objetivo. . .	43
5.7. RMSE promedio de los tres recorridos del objetivo para el KF, EKF y UKF.	43
5.8. Comparación entre el sistema de Vuelo Autónomo (VA) del primer enfoque propuesto y el Vuelo a Control Remoto (VCR) ejecutado por un usuario	46
5.9. Comparación entre el sistema de Vuelo Autónomo (VA) del segundo enfoque propuesto y el Vuelo a Control Remoto (VCR) ejecutado por un usuario	49

Capítulo 1

Introducción

En la actualidad, los Vehículos Aéreos No Tripulados (VANT's), también conocidos como drones, han tomado gran importancia debido a su bajo costo, tamaño, y especialmente por la aplicación que tienen en diferentes áreas de la industria y en la vida cotidiana (Figura 1.1). Dentro de las principales tareas que están tomando relevancia hoy en día se encuentra el uso de drones en la captura de video y toma de fotografías aéreas en áreas como el cine, deportes y eventos sociales, para aplicaciones de seguimiento, búsqueda y vigilancia de objetivos. Sin embargo, se requiere de personas con la habilidad suficiente para llevar a cabo las tareas encomendadas en tiempo y forma con el fin de evitar riesgos que lleven a una pérdida innecesaria de elementos. Es por ello, que el vuelo autónomo de un dron facilitaría el desarrollo de estas tareas.



Figura 1.1: Aplicaciones de los VANT's.

Imágenes tomadas de [1] [2] [3] [4].

Motivados por lo anterior, en este trabajo se investiga cómo aprovechar las aplicaciones que involucran a los drones cuyo objetivo es capturar video o fotografías de un objetivo de forma autónoma. Es por ello, que se busca que el dron realice dos

tareas principales: 1) permanecer en modo “hovering” si el objetivo se encuentra dentro de los límites del campo de visión de la cámara a bordo del dron; 2) realizar el vuelo de forma autónoma si el objetivo se sale del campo de visión con el fin de mantenerlo dentro del mismo.

Para lograr las tareas mencionadas, se propone un sistema que permite obtener una estimación de los estados del objetivo a partir de la información capturada por la cámara a bordo del dron. Estos estados incluyen la posición del objetivo en la imagen, su velocidad y su dirección y a partir de ellos se construyen reglas de control de velocidad (proporcional a la velocidad del objetivo) para que el dron pueda reaccionar a las situaciones mencionadas anteriormente, las cuales dependen de que el objetivo se encuentre o no dentro del campo de visión.

Para evaluar el sistema autónomo propuesto, se realiza una comparación de su desempeño con el de pilotos humanos con diferente nivel de habilidad de vuelo mientras que el objetivo se mueve en diferentes trayectorias con cierto grado de complejidad.

1.1. Descripción del problema

En la literatura existen trabajos enfocados en el vuelo autónomo de drones [5] y en el seguimiento puntual de objetivos empleando técnicas de visión [6] y herramientas como los estimadores estocásticos [7]. Sin embargo, a veces no es necesario llevar a cabo este seguimiento puntual ya que, de acuerdo a su aplicación, sólo se requiere mantener al objetivo dentro del campo de visión.

Para mantener a un objetivo dentro del campo de visión de un dron primero es necesario establecer la comunicación adecuada entre el dron y la estación de control en tierra que permita adquirir las imágenes captadas por el dron para su procesamiento y detección del objetivo. Posteriormente, se requiere encontrar una forma de estimar los estados del objetivo, tanto observables como no observables, con el fin de poder definir una política de navegación autónoma y eficiente. Finalmente, se deben realizar las pruebas necesarias que validen el funcionamiento del sistema para mantener a un objetivo dentro del campo de visión.

1.2. Justificación

En la vida diaria se presentan situaciones como la toma de video y fotografía para seguimiento o vigilancia de un objetivo en un área de peligro, que conllevan cierto grado de dificultad y riesgos para los elementos humanos. Sin embargo, estos riesgos pueden disminuir si se cuenta con elementos, como los drones, que lleven a cabo las tareas encomendadas. Aún así, el vuelo de drones requiere de elementos humanos con ciertas habilidades y condición que les permitan realizar las tareas de forma remota.

Por esta razón y con el fin de eliminar la dependencia del piloto que controla al dron, en el presente trabajo se implementa un sistema autónomo para mantener a un objetivo dentro del campo de visión de la cámara a bordo de un dron, lo que permite disminuir el riesgo de errores, la reducción del costo económico y la fátiga innecesaria del operador humano.

1.3. Objetivos

1.3.1. Objetivo general

Desarrollar un sistema que permita controlar de forma autónoma a un dron con el propósito de mantener a un objetivo dentro del campo de visión de la cámara que tiene a bordo.

1.3.2. Objetivos específicos

- Implementar el sistema de comunicación entre la PC y el dron..
- Realizar el procesamiento de la imagen adquirida con la cámara a bordo del dron para poder detectar al objetivo a mantener dentro del campo de visión.
- Seleccionar un algoritmo basado en un estimador estocástico que permita predecir estados observables y no observables del objetivo.
- A partir de la predicción de estados, diseñar políticas de navegación que permitan mantener al objetivo dentro del campo de visión de forma autónoma.

- Diseñar un marco experimental para realizar las pruebas necesarias del sistema diseñado.

1.4. Estructura de la tesis

Para presentar el sistema propuesto, esta tesis está organizada como a continuación se indica:

En el Capítulo 2 se hace una revisión del trabajo relacionado con seguimiento de objetivos utilizando técnicas como la segmentación de imagen, así mismo se presenta el uso de estimadores estocásticos como el KF, EKF y UKF para el seguimiento de trayectorias lineales y no lineales en aplicaciones de robótica.

En el Capítulo 3 se presenta el marco teórico que hace referencia a los conceptos básicos de las herramientas utilizadas para la implementación del sistema propuesto relacionados con el VANT, el sistema operativo ROS, el procesamiento de imágenes y los estimadores estocásticos KF, EKF y UKF.

En el Capítulo 4 se describe la metodología propuesta para la implementación del sistema a desarrollar, desde la adquisición y el procesamiento de la imagen y la selección del estimador estocástico hasta la implementación del control reactivo que permite mantener a un objetivo dentro del campo de visión.

En el Capítulo 5 se muestran los resultados de los experimentos realizados que permiten seleccionar al estimador estocástico adecuado. También, se muestran los resultados obtenidos con el sistema propuesto aplicado a un objetivo visto desde el frente del dron y a un objetivo visto desde arriba por el dron.

Finalmente, en el Capítulo 6 se presentan las conclusiones del trabajo realizado además de las limitaciones y el trabajo a futuro

Trabajo relacionado

El uso de VANT's en diferentes aplicaciones de la vida cotidiana ha ido incrementando con el paso del tiempo con el fin de facilitar las labores humanas y disminuir el riesgo en áreas donde se realizan actividades que implican cierto grado de peligro. Una de las aplicaciones que ha cobrado importancia con el uso de los VANT's es el seguimiento, vigilancia y toma de video aéreo de objetivos, es por ello, que en la literatura se pueden encontrar trabajos relacionados con diferentes técnicas utilizadas para llevar a cabo esta misión.

Dang et al. [8] presentaron un enfoque para el problema de seguimiento de objetos en tierra con un quadrotor a través de su sistema de visión integrado. En este enfoque ellos utilizan el método de la segmentación de una imagen basado en color para identificar y estimar la posición del objetivo sobre la imagen capturada con ayuda del sistema de visión del dron (Figura 2.1).

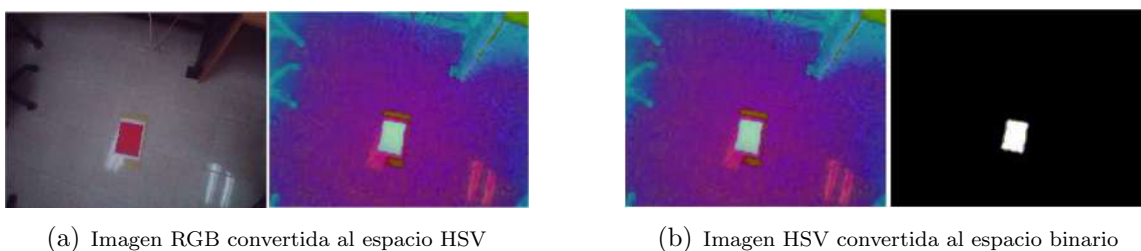
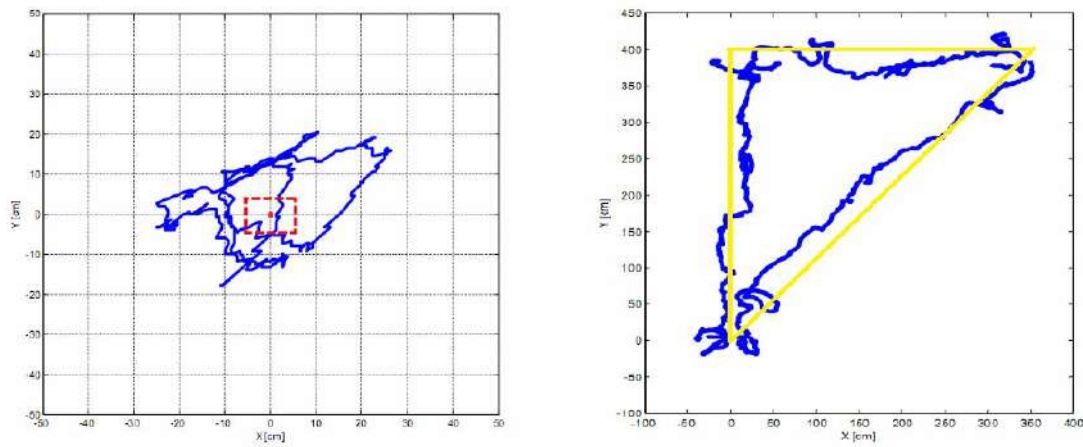


Figura 2.1: Segmentación de imagen basada en color
Imagen tomada de [8].

La imagen adquirida por la cámara se encuentra en el espacio de color RGB y para su segmentación basada en color es convertida al espacio HSV para posteriormente obtener una imagen binaria a partir del umbral establecido (Figura 2.1). Cuando

la imagen se encuentra en el espacio binario, las coordenadas del objetivo dentro de la imagen se estiman mediante la distribución de probabilidad óptima con un filtro Gaussiano. Las pruebas se realizaron con un objetivo estacionario en tierra y con un objetivo en movimiento en una trayectoria triangular. Para que el dron se estabilice cuando engancha al objetivo en tierra utilizaron un controlador PD que permite mantener al objetivo que está en tierra en el centro de la imagen adquirida. Sin embargo, en el intento por mantener de forma puntual al objetivo en el centro de la imagen adquirida, el dron realiza movimientos indeseados como se puede ver en la Figura 2.2.



(a) Seguimiento de un objetivo estacionario. Rojo: Objeto, Azul: Drone

(b) Seguimiento de un objetivo en movimiento. Amarillo: Trayectoria del objetivo, Azul: Trayectoria del dron

Figura 2.2: Resultados experimentales del seguimiento de un objetivo.

Imagen tomada de [8].

Vinayagam et al. [9] presentaron un sistema a bordo de un dron para el seguimiento de objetivos en tiempo real y que fué implementado en una tarjeta Raspberry Pi para el control de vuelo y el procesamiento de imagen. Para el seguimiento de objetivos estáticos o en movimiento, implementaron dos controladores: el Controlador para el Seguimiento de Objetivos y el Controlador PID para la estabilidad del sistema.

Después de analizar los principios teóricos y los mecanismos de implementación de un algoritmo de seguimiento de objetivos, presentaron una mejora del algoritmo TLD, el cual es un algoritmo que permite realizar el seguimiento de un objetivo por un largo

tiempo [10] y que se compone de seguimiento (Tracking), aprendizaje (Learning) y detección (Detection). Este algoritmo envía la información al Controlador PID para que el dron pueda ajustar su posición hasta que su objetivo esté centrado en el campo de visión de la cámara a bordo del mismo. El diseño que implementaron se puede ver en la Figura 2.3 y permite que el seguimiento de un objetivo sea mas estable y en tiempo real. El seguimiento realizado en este trabajo es puntual, esto quiere decir, que los movimientos que realiza el objetivo también son realizados por el dron, lo que en muchas ocasiones no es necesario, ya que para aplicaciones de vigilancia solo es necesario mantenerlo dentro del campo de visión del dron.

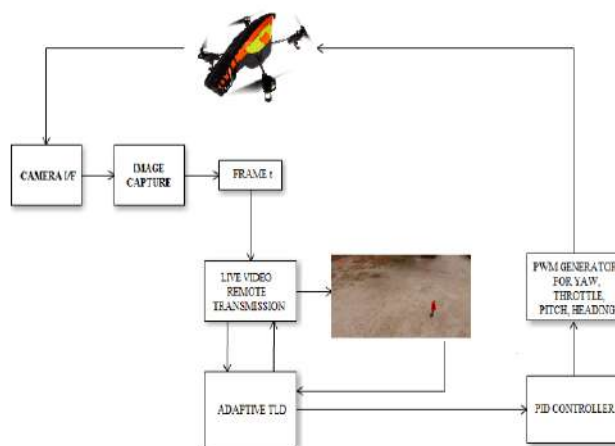


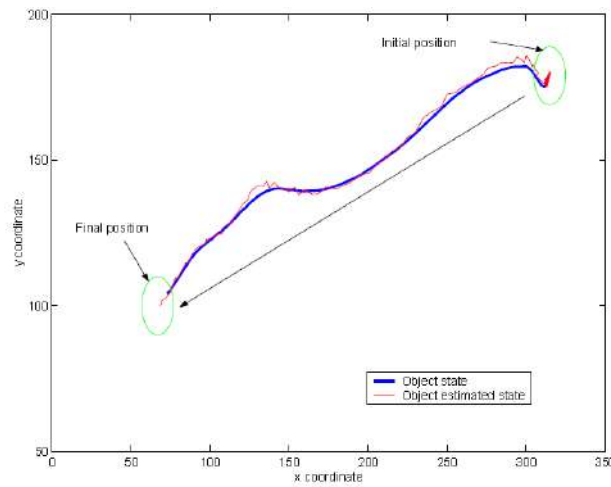
Figura 2.3: Diagrama de flujo del Controlador para el seguimiento de objetivos.

Imagen tomada de [9].

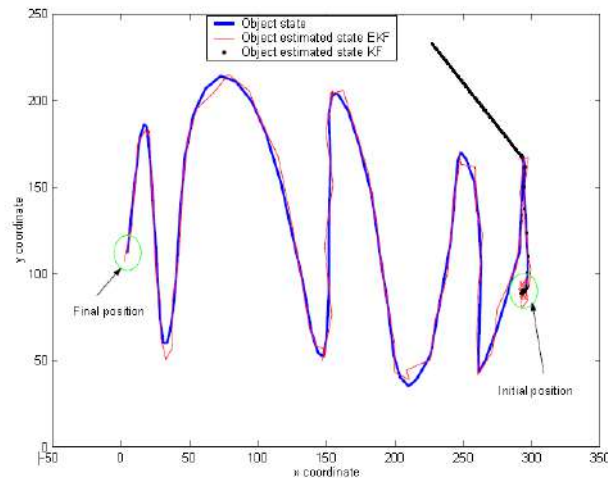
En los trabajos anteriores el seguimiento de objetivos está basado en sistemas de visión y procesamiento de imagen, sin embargo, el uso de estimadores estocásticos como el KF, EKF y el UKF en áreas como la robótica para el seguimiento de objetivos están siendo de gran importancia, mostrando resultados favorables.

Cuevas et al. [7] presentaron el uso de del KF para el seguimiento de una pelota de futbol soccer con un movimiento lineal uniforme. El movimiento de la pelota fué extraído de una secuencia de imágenes adquiridas por una cámara, las cuales fueron procesadas y segmentadas para obtener la posición de la pelota en cada instante de tiempo dentro de la imagen. En la Figura 2.4a se puede ver el resultado de la implementación del KF, donde la línea roja representa la estimación calculada con este.

Debido a que el KF sólo es para sistemas lineales, en este trabajo también implementaron el EKF, el cual es un estimador estocástico para sistemas no lineales. En la Figura 2.4b se muestra el resultado de la estimación con el EKF para una trayectoria compleja del objetivo, así mismo, se puede observar que el KF tiene una respuesta deficiente para este tipo de trayectorias.



(a) Seguimiento de una trayectoria lineal de un objetivo usando el KF



(b) Seguimiento de una trayectoria compleja de un objetivo usando el KF y el EKF.

Figura 2.4: Simulación de respuesta del KF y el EKF.

Imagen tomada de [7].

Nivedita y Pooja [11] también implementaron el EKF para el seguimiento de un objetivo que realiza un movimiento no lineal en la plataforma de Simulink. Cabe mencionar que los trabajos presentados del EKF fueron realizados en simulaciones.

En 1997, Julier y Uhlmann [12] desarrollaron y demostraron un nuevo estimador estocástico bajo el principio de que un conjunto de puntos de muestra permiten obtener la media y la covarianza de un sistema. Este nuevo estimador está basado en el KF pero enfocado a sistemas no lineales sin los pasos de linealización requeridos por el EKF. Posteriormente, Leyton [13] presentó un trabajo de tesis que se enfoca en el desarrollo e implementación en Matlab/Simulink del UKF para estimar variables de estado de interés de un mini-helicóptero robot, a partir de variables de estado observables que se obtienen de los sensores de a bordo del mini-helicóptero.

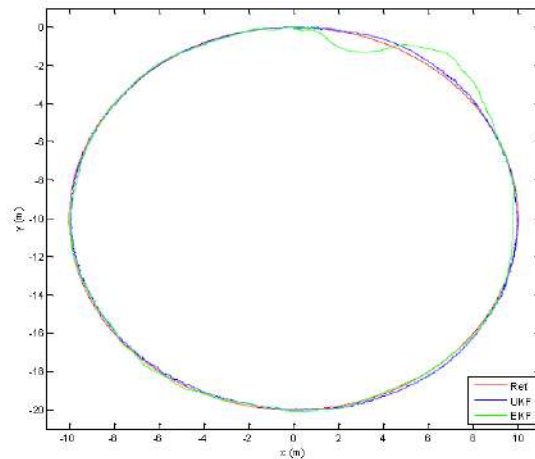


Figura 2.5: Comparación de las trayectorias en XY del UKF y EKF.

Imagen tomada de [13].

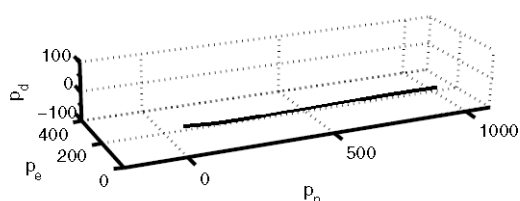
El comportamiento del sistema implementado fue analizado con una simulación en la estimación de estados del mini-helicóptero robot en conjunto con un controlador PID para el seguimiento de una trayectoria circular (2.5). El objetivo de las pruebas realizadas es determinar si el comportamiento del vector de estado con el UKF es mejor que el EKF mientras el mini-helicóptero sigue una trayectoria circular y esto se puede comprobar con los resultados que se muestran en la Tabla 2.1.

Algoritmo	Promedio de error RMSE								
	Posición			Velocidad			Ángulos de Euler		
	x	y	z	v_x	v_y	v_z	$roll$	$pitch$	yaw
UKF	0.1313	0.0536	0.0181	0.1315	0.0604	0.1874	0.0023	0.0023	0.0868
EKF	0.1340	0.0726	0.0479	0.1694	0.1257	0.2382	0.0052	0.0052	0.0046

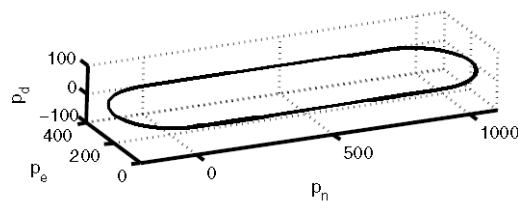
Tabla 2.1: Comparación de los errores RMSE en los filtros EKF y UKF.

Tabla tomada de [13].

Natássya et al. [14] realizaron una evaluación para comparar el desempeño del EKF y del UKF en diferentes situaciones. Los factores que evaluaron fueron el tipo de filtro (EKF o UKF), el tamaño del vector de estado (10 o 16 estados) y la trayectoria (parcial o completa). En la Figura 2.6 se muestran las trayectorias empleadas para la evaluación del desempeño de los filtros.



(a) Trayectoria parcial con un nivel constante de vuelo



(b) Trayectoria completa con un nivel constante de vuelo con maniobras

Figura 2.6: Trayectorias para evaluación de desempeño de los filtros.

Imagen tomada de [14].

Los experimentos son realizados con simulaciones de Matlab y los datos son generados con ruido blanco añadido y sesgo de acuerdo a las covarianzas típicas de los sensores utilizados en los VANT's.

Los resultados de la evaluación del desempeño se pueden ver en la Figura 2.7, que muestra el tiempo de ejecución de cada uno de los filtros para cada una de las trayectorias y en la Figura 2.8 que muestra el error RMS (Root Mean Square) de las posiciones.

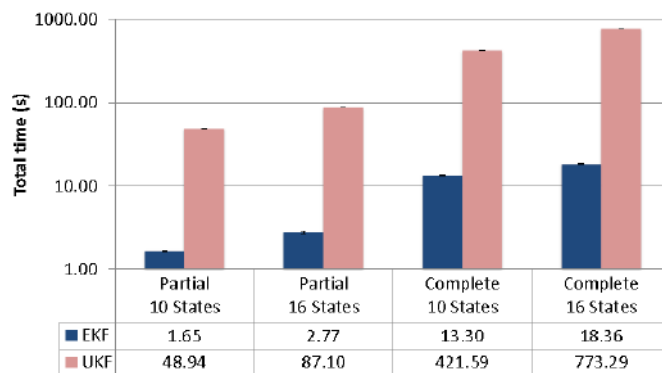
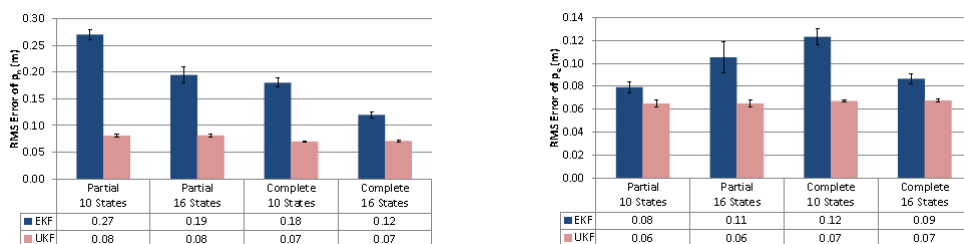


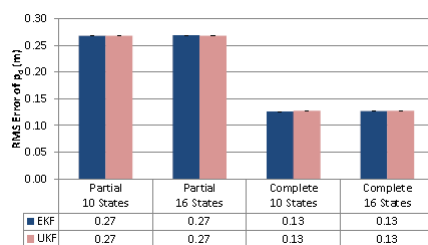
Figura 2.7: Tiempo promedio de ejecución del EKF y UKF.

Imagen tomada de [14].



(a) Error RMS promedio de p_n

(b) Error RMS promedio de p_e



(c) Error RMS promedio de p_d

Figura 2.8: Error RMS promedio de las posiciones

Imagen tomada de [14].

En este trabajo llegaron a la conclusión de que el EKF tiene un costo computacional menor que el UKF debido a que la propagación de los sigma-points es más costosa que la propagación de un solo vector de estado. Sin embargo, el UKF presenta un menor error en la estimación de la posición.

Podemos darnos cuenta que los trabajos relacionados con el seguimiento de trayectorias utilizando el UKF son simulaciones y no están implementados en tiempo real.

Como se mencionó al principio de este capítulo, el uso de VANT's para seguimiento y vigilancia de objetivos ha ido incrementando, sin embargo, en muchas ocasiones para operaciones de vigilancia de un objetivo o simplemente para toma de video aéreo no es necesario realizar un seguimiento puntual y lo que se debe buscar es mantener al objetivo dentro del campo de visión del VANT utilizando los sensores y herramientas necesarias para ello. Es por ello que en este trabajo presentamos un sistema que permite mantener a un objetivo dentro del campo de visión de la cámara a bordo del dron utilizando estimadores estocásticos.

Capítulo 3

Marco teórico

En este capítulo se presentan los conceptos teóricos de las herramientas necesarias para el desarrollo del trabajo presentado.

Dentro de estos conceptos se encuentra el del quadrotor (tipo de dron utilizado en este trabajo), ROS que es el sistema operativo utilizado para entablar comunicación con el dron, los conceptos necesarios para el procesamiento de imagen y la descripción teórica de los estimadores estocásticos implementados. Finalmente, se describen los conceptos del tipo de control utilizado para poder cumplir el objetivo propuesto.

3.1. Quadrotor

De acuerdo al Departamento de Defensa (DOD, por sus siglas en inglés) de los Estados Unidos Americanos, en su Diccionario de Términos Militares y Asociados, una aeronave no tripulada es una aeronave que no lleva un operador humano y es capaz de volar con o sin control remoto humano [15]. A lo largo de la historia se han utilizado diferentes nombres para describir a una aeronave no tripulada, siendo los más utilizados Vehículo Aéreo No Tripulado (VANT) y Unmanned Aerial System (UAS).

Un VANT (también conocido como dron) se refiere a una aeronave no tripulada, una máquina de vuelo sin piloto humano o pasajeros a bordo. Como tal “no tripulado” implica la ausencia total de un ser humano que dirija y opere la aeronave [16]. Estos vehículos han sido ampliamente utilizados para diferentes aplicaciones como búsqueda y rescate, vigilancia aérea, monitoreo, detección de fuego, grabaciones y toma de fotografías aéreas.

Uno de los drones más utilizados en la actualidad es el helicóptero de cuatro

rotores también conocido como quadrotor. Los rotores del quadrotor están montados simétricamente por lo que su movimiento es similar al de un helicóptero convencional. La diferencia es que su movimiento se genera cuando se varía la velocidad de cada uno de los rotores de acuerdo al movimiento deseado [17]. En la Figura 3.1 se muestran los movimientos de los rotores de acuerdo al movimiento deseado del quadrotor.

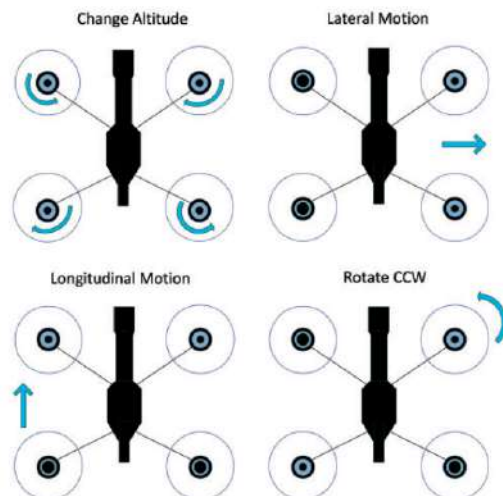


Figura 3.1: Movimientos básicos del quadrotor
Imagen tomada de [17].

Los movimientos del dron también se conocen por el ángulo en el que se realicen. El cambio de altitud se conoce como vertical rate, el movimiento lateral como movimiento en roll, el movimiento longitudinal como movimiento en pitch y la rotación en sentido de las manecillas del reloj o en contra como movimiento en yaw.

3.2. ROS

El Robotic Operating System (ROS) es un entorno de trabajo flexible para el desarrollo de software para robots que provee la funcionalidad de un sistema operativo en un conjunto de ordenadores unidos entre sí por una red de alta velocidad que permite establecer comunicación con el dron. ROS es una colección de herramientas, librerías y convenciones que tienen como objetivo simplificar la tarea de crear un comportamiento robótico complejo y robusto a través de una amplia variedad de plataformas robóticas [18].

3.2.1. Conceptos básicos

Una red ROS comprende diferentes partes de un sistema de robot que se comunican a través de ROS. La red se puede distribuir en varias máquinas y se compone de los siguientes elementos:

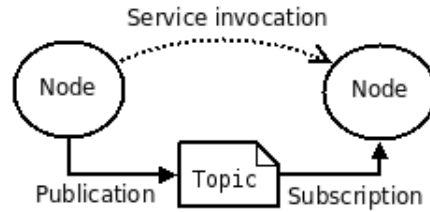


Figura 3.2: Conceptos básicos de ROS

Imagen tomada de [19].

- ROS master, se encarga de coordinar las diferentes partes de una red ROS. Es identificado por un Master URI (Uniform Resource Identifier) que especifica el hostname o la dirección IP de la máquina donde se está ejecutando el master.
- Un nodo ROS es una entidad que contiene una colección de capacidades (como *Publishers*, *suscribers* y *services*). Una red ROS puede tener muchos nodos ROS.
- *Publishers*, *suscribers* y *services* son diferentes tipos de entidades que procesan datos. Intercambian datos usando mensajes. Un *publisher* envía mensajes a un tópico específico y quienes se encuentran suscritos a este tópico reciben los mensajes. Puede haber múltiples *publishers* y *suscribers* asociados a un solo tópico.
- Un tópico es un bus que sirve para intercambiar mensajes entre nodos y cuyo nombre se usa para identificar el contenido de un mensaje. Un nodo publica mensajes en un tópico, y si existe un nodo interesado en algún mensaje, este nodo se suscribe al tópico adecuado.
- El servidor de parámetros es un diccionario compartido de múltiples variables. Los nodos utilizan este servidor para almacenar y recuperar parámetros en tiempo de ejecución. El nodo de registro *rosout*, es el mecanismo de informe de registro de la consola en ROS.

- El *roscore* es una colección de nodos y programas que son prerequisites de un sistema basado en ROS. Debe existir un *roscore* con ejecución para que los nodos ROS se comuniquen. Cuando se ejecuta *roscore* se inicializa el maestro ROS, el servidor de parámetros y el nodo de registro *rosout* [19].

3.3. Procesamiento de imagen

Una imagen es una representación visual de un objeto, una persona o una escena que se produce por un dispositivo óptico tal como un espejo, un lente o una cámara [20].

Una imagen puede ser definida como una función de dos dimensiones, $f(x, y)$, donde x y y son coordenadas espaciales, y la amplitud de f en cualquier par de coordenadas (x, y) es llamada la intensidad o nivel de gris de la imagen en ese punto. Cuando x , y , y la amplitud de f son finitos, cantidades discretas, la imagen se llama *imagen digital* [21].

Cuando se tiene una imagen digital se representa como una matriz de la siguiente forma:

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, N-1) \\ f(1,0) & f(1,1) & \dots & f(1, N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1, N-1) \end{bmatrix}$$

donde M es el número de filas, N el número de columnas y se dice que la imagen es de tamaño $M \times N$.

El procesamiento de una imagen puede ser definido como la ciencia que modifica imágenes digitales por medio de una computadora digital. Los cambios que se producen en las imágenes se realizan generalmente de forma automática y se basan en algoritmos diseñados cuidadosamente [20] y pueden ser de bajo nivel si se realizan operaciones primitivas, de medio nivel si se lleva a cabo la extracción de características y de alto nivel si se hace un análisis e interpretación del contenido de una escena.

3.3.1. Modelo de color

El color es percibido por los humanos como una combinación triestímulo de R (red), G (Green) y B (blue) los cuales son usualmente conocidos como los tres colores primarios. De la representación R,G,B se pueden derivar otro tipo de espacios de color usando una transformación lineal o no lineal [22].

La representación de colores en una imagen se obtiene usando una combinación de uno o más canales de color que son combinados para formar el color usado en la imagen. La representación que se utiliza para almacenar los colores, especificando el número y la naturaleza de los canales de color, se conoce generalmente como el espacio de color [23].

Un espacio de color o modelo de color es una especificación de un sistema de coordenadas y un subespacio dentro de ese sistema donde cada color está representado por un solo punto [20] conocido como pixel. Existen diferentes modelos de color que se utilizan para diferentes propósitos y uno de los más conocidos y usado para para la representación de imágenes digitales es el modelo de color RGB.

El modelo de color RGB se puede ver en la Figura 3.3a y está basado en un sistema Cartesiano de coordenadas cuyos ejes representan a los tres colores primarios de luz (R, G, y B), usualmente normalizados en el rango $[0, 1]$. Los ocho vértices del cubo resultante que corresponden a los tres colores primarios de luz, los tres colores secundarios, el color blanco y el color negro. En la Figura 3.3b se muestra el cubo de color RGB resultante.

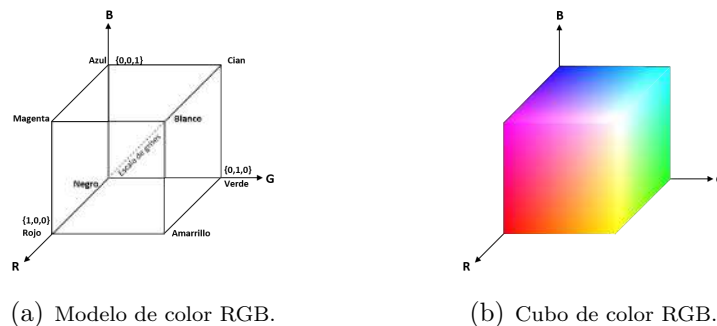


Figura 3.3: Representación del espacio de color RGB.

3.3.2. Imagen RGB

Una imagen de color RGB es un arreglo de $M \times N \times 3$ píxeles de color, donde cada píxel de color es un triplete que corresponde a las componentes roja, verde y azul de una imagen RGB en una ubicación espacial específica [21].

El número de filas M representa la altura de la imagen, el número de columnas N representa la anchura de la imagen y el 3 representa el número de componentes cromáticas. Cada arreglo individual de tamaño $M \times N$ se conoce como componente cromática de la imagen y en la Figura 3.4 se observan las tres componentes de la imagen RGB.

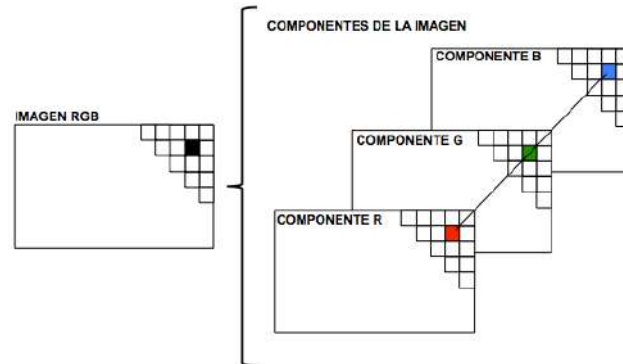


Figura 3.4: Componentes de una imagen de color RGB.

3.3.3. Binarización

La binarización de una imagen es un proceso de reducción de la información que contiene, y muestra la información resultante en sólo dos valores: verdadero o falso (binaria). Se basa en la segmentación usando un umbral de intensidad.

Se escoge un valor de umbral tal que los píxeles que poseen valores mayores que el umbral se asignan a una región mientras que aquellos que caen por debajo del umbral se asignan a otra región (adjunta) [23]. El umbral crea una imagen binaria $im_{bin}(x, y)$ de una imagen de intensidad $I(x, y)$ bajo el siguiente criterio:

$$im_{bin}(x, y) = \begin{cases} 1, & \text{si } I(x, y) > T \\ 0, & \text{si } I(x, y) \leq T \end{cases} \quad (3.3.1)$$

donde T es el valor del umbral (threshold).

En la Figura 3.5 se muestra el resultado de una binarización con un algoritmo implementado en Matlab.

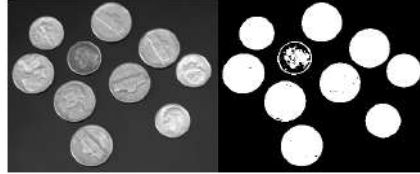


Figura 3.5: Binarización de una imagen. Izquierda:imagen original, derecha:imagen binarizada
Algoritmo para imagen tomado de [24].

3.3.4. Operaciones morfológicas

La morfología matemática es una rama del procesamiento de imagen que ha sido usada satisfactoriamente para proporcionar herramientas que representen, describan y analicen figuras en una imagen. El principio básico de la morfología matemática es la extracción de información geométrica y topológica de un conjunto desconocido (una imagen) a través de transformaciones usando otro elemento bien definido como elemento estructurante (SE, Structuring Element). En el procesamiento morfológico de una imagen, el diseño de los SEs, su forma y tamaño, es crucial para el éxito de las operaciones morfológicas que las utilizan [20].

Un SE es una región con valores binarios, 2D o multidimensional, en el que los pixeles verdaderos se incluyen en el cálculo morfológico y los pixeles falsos no. El pixel del centro del SE, llamado el origen, identifica el pixel de la imagen que será procesado [24]. Existen diferentes SEs y un ejemplo de ellos se muestra en la Figura 3.6. Dentro de las operaciones morfológicas más utilizadas que se realizan a las imágenes se encuentran la dilatación y la erosión.

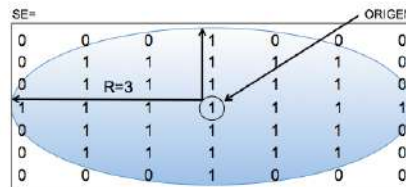


Figura 3.6: Ejemplo de elemento estructurante.

Imagen tomada de [24].

Dilatación

Es una operación morfológica que “crece” o “engrosa” una imagen binaria. La forma específica y el grosor que puede alcanzar esta operación es controlada por el SE [21]. Matemáticamente la dilatación de una imagen A por un SE donde \emptyset es un conjunto vacío, está definida por:

$$A \oplus SE = \left\{ z \mid (\hat{S}E_z) \cap A \neq \emptyset \right\} \quad (3.3.2)$$

La Figura 3.7 muestra cómo trabaja la dilatación. La dilatación de la imagen binaria A por el elemento estructurante SE ocurre cuando el centro de este recorre a la imagen A y la cubre con valores verdaderos correspondientes a la forma y grosor del SE cuando encuentra valores verdaderos sobre la imagen.

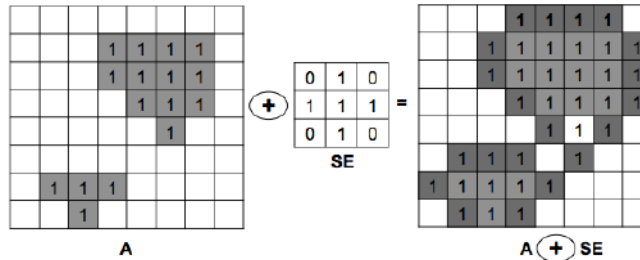


Figura 3.7: Dilatación de una imagen.

Erosión

Es una operación morfológica que “encoge” o “adelgaza” los objetos en una imagen binaria. Como en la dilatación, la forma específica y el grosor que puede alcanzar esta operación es controlada por el SE [21]. Matemáticamente la erosión de una imagen A por un SE está definida por:

$$A \ominus SE = \left\{ z \mid (\hat{S}E_z) \cap A^c \neq \emptyset \right\} \quad (3.3.3)$$

La Figura 3.8 muestra que la operación de erosión en una imagen ocurre cuando se considera que el centro del elemento estructurante SE se superpone a cada valor verdadero de A, si SE está contenido completamente en A el valor del pixel verdadero de A se mantiene, de lo contrario es eliminado.

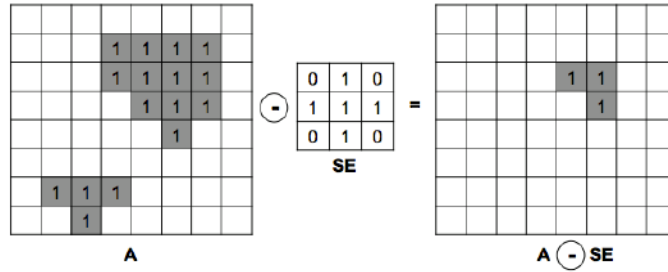


Figura 3.8: Erosión de una imagen.

3.3.5. Centroide

El centroide de una imagen, también conocido como centro de áreas, representa el lugar geométrico donde se concentran las áreas que conforman a los pixeles de la misma intensidad. Como se observa en la Figura 3.9, en el caso de una imagen binarizada el centroide representa la posición promedio de todos los pixeles de color blanco ya que los pixeles de color negro se consideran vacíos.

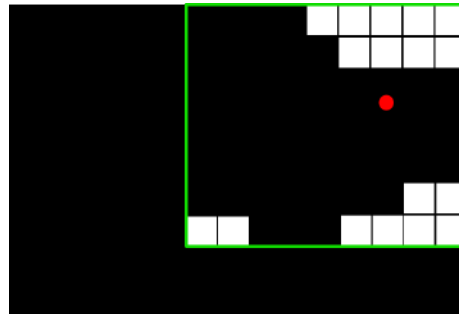


Figura 3.9: Representación del centroide de una imagen.

Las coordenadas del centroide de un objeto O_i denotadas como (\bar{x}_i, \bar{y}_i) están dadas por:

$$\bar{x}_i = \frac{1}{A_i} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} x O_i(x, y) \quad (3.3.4)$$

$$\bar{y}_i = \frac{1}{A_i} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} y O_i(x, y) \quad (3.3.5)$$

donde A_i es el área del objeto O_i y M y N son el ancho y alto de la imagen, respectivamente.

3.4. Estimadores de estados estocásticos

Un estimador de estados es un sistema que está compuesto por variables de estado que representan al sistema en equilibrio y que tiene la capacidad de predecir el estado siguiente por las acciones predecibles del proceso y por elementos aleatorios. Se dice que es estocástico cuando el desarrollo temporal del mismo se puede analizar en términos de probabilidad.

Dentro de la literatura, uno de los estimadores estocásticos ampliamente utilizado es el Filtro Kalman, sin embargo, debido a que en la mayoría de los sistemas los procesos son no lineales, se ha incrementado el uso del Filtro Kalman Extendido y el Filtro Kalman Unscented presentando diferencias que permiten decidir su implementación de acuerdo a los requerimientos.

3.4.1. Filtro Kalman (KF)

El Filtro Kalman (KF, por sus siglas en inglés) es un conjunto de ecuaciones matemáticas que proporciona un medio recursivo eficiente para estimar el estado de un proceso de tal forma que minimice el error cuadrático medio [25].

Para describir el algoritmo del KF se considera un sistema dinámico lineal representado por las siguientes ecuaciones conocidas como ecuación de proceso (3.4.1) y ecuación de medida (3.4.2), respectivamente:

$$x_k = \mathbf{F}x_{k-1} + w \quad (3.4.1)$$

$$y_{k-1} = \mathbf{H}x_{k-1} + v \quad (3.4.2)$$

donde \mathbf{F} es la matriz de transición del estado x_k , de $k - 1$ a k , w es el ruido de proceso que está normalmente distribuido con media cero y matriz de covarianza \mathbf{Q} , y_{k-1} es el estado observable en $k - 1$, \mathbf{H} es la matriz de medida y v es el ruido de medida que está normalmente distribuido con media cero y matriz de covarianza \mathbf{R} .

El KF se compone de dos etapas: la predicción y la corrección. En la etapa de predicción se propaga un estado y un error de covarianza estimados y en la etapa

de corrección se incorpora una medición realizada que mejora la predicción. En el Algoritmo 3.1 se muestran las ecuaciones del KF con sus respectivas etapas.

Inicializa con:	
$\hat{x}_0 = E[x_0]$ $\mathbf{P}_0 = E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T]$	
Para $k \in \{1, \dots, \infty\}$	
Predicción:	Corrección:
Estado	Innovación de covarianza
$\hat{x}_k^- = \mathbf{F}[\hat{x}_{k-1}]$	$\mathbf{S}_k = \mathbf{H}\mathbf{P}_k^-\mathbf{H}^T + \mathbf{R}$
Covarianza	Ganancia del Filtro
$\mathbf{P}_k^- = \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^T + \mathbf{Q}$	$K = \frac{\mathbf{P}_k^-\mathbf{H}^T}{\mathbf{S}_k}$
Medida	Estado corregido
$\hat{y}_k^- = \mathbf{H}[\hat{x}_k^-]$	$\hat{x}_k = \hat{x}_k^- + K(y_k - \hat{y}_k^-)$
	Covarianza corregida
	$\hat{\mathbf{P}}_k = \mathbf{P}_k^- - K\mathbf{S}_kK^T$
donde y_k es la medida actual observada.	

Algoritmo 3.1: Ecuaciones del Filtro Kalman

3.4.2. Filtro Kalman Extendido (EKF)

Todos los sistemas cuentan con un grado de no linealidad y dependiendo del tipo de proceso y su región de operación pueden ser aproximados con un modelo lineal donde el KF puede ser utilizado como estimador de estados. Sin embargo, en algunos casos esta aproximación lineal no es lo suficientemente adecuada por lo que es necesario un estimador de estados que utilice un modelo no lineal del sistema y el más común es el Filtro Kalman Extendido (EKF, por sus siglas en inglés).

El EKF resuelve el problema de la estimación del estado generado por un sistema no lineal, utilizando la expansión de la serie de Taylor que aproxima las ecuaciones no lineales de estado y de observación, sobre el valor actual estimado del estado con una precisión de primer orden; igualmente, provee un estimado de la varianza mínima del estado basada en la información estadística del modelo [26].

Para describir el algoritmo del EKF se considera un sistema dinámico no lineal representado por las siguientes ecuaciones, conocidas como ecuación de proceso (3.4.3)

y ecuación de medida (3.4.4), respectivamente:

$$x_k = \mathbf{f}[x_{k-1}, w] \quad (3.4.3)$$

$$y_{k-1} = \mathbf{h}[x_{k-1}, v] \quad (3.4.4)$$

donde \mathbf{f} es el modelo dinámico no lineal del sistema de $k - 1$ a k , w es el ruido de proceso que está normalmente distribuido con media cero y matriz de covarianza \mathbf{Q} , y_{k-1} es el estado observable en $k - 1$, \mathbf{h} es la función de medida y v es el ruido de medida que está normalmente distribuido con media cero y matriz de covarianza \mathbf{R} .

El EKF, al igual que el KF se compone de una etapa de predicción y una de corrección como se muestra en el Algoritmo 3.2.

Inicializa con:	
$\hat{x}_0 = E[x_0]$	
$\mathbf{P}_0 = E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T]$	
Para $k \in \{1, \dots, \infty\}$	
Predicción:	Corrección:
Jacobianos del modelo de proceso:	Jacobianos del modelo de medida:
$\mathbf{F} = \left. \frac{d\mathbf{f}}{dx} \right _{x=\hat{x}_{k-1}}$, $\mathbf{W} = \left. \frac{d\mathbf{f}}{dw} \right _{w=\bar{w}}$	$\mathbf{H} = \left. \frac{d\mathbf{h}}{dx} \right _{x=\hat{x}_{k-1}}$, $\mathbf{V} = \left. \frac{d\mathbf{h}}{dv} \right _{v=\bar{v}}$
Estado	Innovación de la covarianza
$\hat{x}_k^- = \mathbf{f}[\hat{x}_{k-1}]$	$\mathbf{S}_k = \mathbf{H}\mathbf{P}_k^-\mathbf{H}^T + \mathbf{V}\mathbf{R}\mathbf{V}^T$
Covarianza	Ganancia del Filtro
$\mathbf{P}_k^- = \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^T + \mathbf{W}\mathbf{Q}\mathbf{W}^T$	$K = \frac{\mathbf{P}_k^-\mathbf{H}^T}{\mathbf{S}_k}$
Medida	Estado corregido
$\hat{y}_k^- = \mathbf{h}[\hat{x}_k^-]$	$\hat{x}_k = \hat{x}_k^- + K(y_k - \hat{y}_k^-)$
	Covarianza corregida
	$\hat{\mathbf{P}}_k = \mathbf{P}_k^- - K\mathbf{S}_kK^T$
donde y_k es la medida actual observada.	

Algoritmo 3.2: Ecuaciones del Filtro Kalman Extendido

3.4.3. Filtro Kalman Unscented (UKF)

El filtro Kalman Unscented (UKF, por sus siglas en inglés) es un estimador de estados que tiene un comportamiento equivalente al KF, pero está enfocado a sistemas no lineales sin los pasos de linealización requeridos por el EKF.

La distribución de estados está representada por una variable aleatoria gaussiana (GRV) como en el EKF, pero ahora se especifica usando un conjunto mínimo de

puntos de muestreo escogidos cuidadosamente. Estos puntos de muestreo capturan completamente la media y covarianza verdaderas de la GRV, y cuando se propagan a través de un sistema no lineal capturan la media y covarianza posterior con una precisión de 3er orden, de acuerdo a la expansión de Taylor, para cualquier no linealidad [27].

Para entender el algoritmo del UKF, primero es necesario explicar la transformación unscented (UT), la cual es un método para calcular la estadística de una variable aleatoria que experimenta una transformación no lineal. Este método está fundamentado en la idea intuitiva que plantea que es más fácil aproximar una distribución gaussiana que aproximar una función no lineal arbitraria [12].

Si se tiene una variable aleatoria x de dimensión L , con media \bar{x} y covarianza \mathbf{P}_x y una segunda variable aleatoria y que se relaciona con x a través de una función no lineal

$$y = f(x),$$

para obtener la media de y , \bar{y} y su covarianza \mathbf{P}_y es necesario construir una matriz de puntos sigma con $2L + 1$ vectores sigma χ_i y sus respectivos pesos W_i de acuerdo a lo siguiente:

$$\begin{aligned} \chi_0 &= \bar{\mathbf{x}} \\ \chi_i &= \bar{\mathbf{x}} + \left(\sqrt{(L + \lambda) \mathbf{P}_x} \right)_i & i = 1, \dots, L \\ \chi_i &= \bar{\mathbf{x}} + \left(\sqrt{(L + \lambda) \mathbf{P}_x} \right)_{i-L} & i = L + 1, \dots, 2L \\ W_0^m &= \lambda / (L + \lambda) \\ W_0^c &= \lambda / (L + \lambda) + (1 - \alpha^2 + \beta) \\ W_i^m &= W_i^c = 1/2 \{(L + \lambda)\} & i = 1, \dots, 2L \end{aligned}$$

donde $\lambda = \alpha^2(L + k) - L$ es un parámetro de escalamiento. α determina la dispersión de los puntos alrededor de $\bar{\mathbf{x}}$ (generalmente un valor positivo pequeño, e.g. $1 \leq \alpha \leq 1e - 4$), k es un parámetro secundario de escalamiento (generalmente 0 o $3 - L$) y β es el parámetro que incorpora conocimiento a priori de la distribución \mathbf{x} (para distribuciones Gaussianas $\beta = 0$). $\left(\sqrt{(L + \lambda) \mathbf{P}_x} \right)_i$ es la i -ésima columna de la raíz de la matriz cuadrada [27]. Estos vectores sigma se propagan a través de la función no lineal y se obtiene una nube de puntos transformados,

$$y_i = f(\chi_i) \quad i = 0, \dots, 2L$$

a los que se les calcula su estadística y con sus respectivos pesos calculados se obtiene la media y la covarianza.

El UKF se puede considerar como el resultado de incorporar la UT al EKF para mejorar las aproximaciones que se hacen de los dos primeros momentos de una variable aleatoria que resulta de propagar otra variable aleatoria a través de una transformación no lineal [28].

En el Algoritmo 3.3, se presentan las ecuaciones del UKF que asume un modelo no lineal del sistema y es representado por la ecuación de proceso y de medida (3.4.3) y (3.4.4), respectivamente.

Inicializa con:	
$\hat{x}_0 = E[x_0]$	
$\mathbf{P}_0 = E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T]$	
$\hat{x}_0^a = E[x^a] = [x_0^T \quad \mathbf{0} \quad \mathbf{0}]$	
$\mathbf{P}_0^a = E[(x_0^a - \hat{x}_0^a)(x_0^a - \hat{x}_0^a)^T] = \begin{bmatrix} \mathbf{P}_0 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R} \end{bmatrix}$	
Para $k \in \{1, \dots, \infty\}$	
Calcular los sigma-points:	
$\chi_{k-1}^a = [\hat{x}_{k-1}^a \quad \hat{x}_{k-1}^a \pm \sqrt{(L + \lambda) \mathbf{P}_{k-1}^a}]$	
Predicción:	Corrección:
Estado	Innovación de la covarianza
$\chi_{k k-1}^x = \mathbf{F} [\chi_{k-1}^x, \chi_{k-1}^w]$	$\mathbf{P}_{\hat{y}_k \hat{y}_k} = \sum_{i=0}^{2L} W_i^{(c)} [\mathbf{y}_{i,k k-1} - \hat{y}_k^-] [\mathbf{y}_{i,k k-1} - \hat{y}_k^-]^T$
$\hat{x}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \chi_{i,k k-1}^x$	$\mathbf{P}_{\mathbf{x}_k \mathbf{y}_k} = \sum_{i=0}^{2L} W_i^{(c)} [\chi_{i,k k-1}^x - \hat{x}_k^-] [\mathbf{y}_{i,k k-1} - \hat{y}_k^-]^T$
Covarianza	Ganancia del Filtro
$\mathbf{P}_k^- = \sum_{i=0}^{2L} W_i^{(c)} [\chi_{i,k k-1}^x - \hat{x}_k^-] [\chi_{i,k k-1}^x - \hat{x}_k^-]^T$	$K = \frac{\mathbf{P}_{\mathbf{x}_k \mathbf{y}_k}}{\mathbf{P}_{\hat{y}_k \hat{y}_k}}$
Medida	Estado corregido
$\mathbf{y}_{k k-1} = \mathbf{H} [\chi_{k k-1}^x, \chi_{k-1}^v]$	$\hat{x}_k = \hat{x}_k^- + K (y_k - \hat{y}_k^-)$
$\hat{y}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \mathbf{y}_{i,k k-1}$	Covarianza corregida
$\hat{\mathbf{P}}_k = \mathbf{P}_k^- - K \mathbf{P}_{\hat{y}_k \hat{y}_k} K^T$	
donde $x^a = [x^T \quad w^T \quad v^T]^T$, $\chi^a = [(\chi^x)^T \quad (\chi^w)^T \quad (\chi^v)^T]^T$, λ =parámetro de escalamiento, L =dimension de estado aumentado, \mathbf{Q} =cov. de ruido de proceso, \mathbf{R} =cov. de ruido de medida, W_i =pesos.	

Algoritmo 3.3: Ecuaciones del Filtro Kalman Unscented

Algoritmo tomado de [27]

Capítulo 4

Metodología

En este capítulo se presenta la metodología propuesta para el desarrollo de este trabajo y se compone de la adquisición y el procesamiento de la imagen, la implementación de los estimadores estocásticos para seleccionar al adecuado y el diseño del control reactivo propuesto para los dos enfoques presentados.

4.1. Adquisición y procesamiento de imagen

Para realizar el procesamiento de una imagen, de acuerdo a las características del objetivo con el que se va a trabajar, se lleva a cabo la adquisición de las imágenes de un video con el dron a través de C++ utilizando el sistema operativo ROS.

El dron utilizado para este trabajo es el quadrotor Parrot Bebop 2 (Figura 4.1), el cual es un dron de última generación utilizado para actividades recreativas y tiene la capacidad de volar, filmar y tomar fotos al mismo tiempo tanto en interiores como exteriores. Cuenta con una cámara de 14 megapíxeles con ángulo de captura de 180° horizontal y vertical y tiene un alcance de 300 m [29].



Figura 4.1: Quadrotor Parrot Bebop 2 utilizado para la implementación del trabajo propuesto.

Imagen tomada de [29].

4.1.1. Adquisición de la imagen

Para la adquisición de la imagen con C++ a través del sistema operativo ROS se crea una red ROS, la cual, se compone de diferentes nodos y un maestro. Primero se lanza el roscore que se encarga de inicializar el puerto 11311 de la PC como maestro y es necesario para que los nodos se comuniquen entre sí. Posteriormente se inicializan los nodos que conforman a la red, el que corresponde a la cámara del dron para la toma de la imagen y el de C++ que permite visualizar y almacenar la imagen.

El proceso de adquisición de las imágenes con la red ROS que se muestra en la Figura 4.2 funciona como se menciona a continuación:

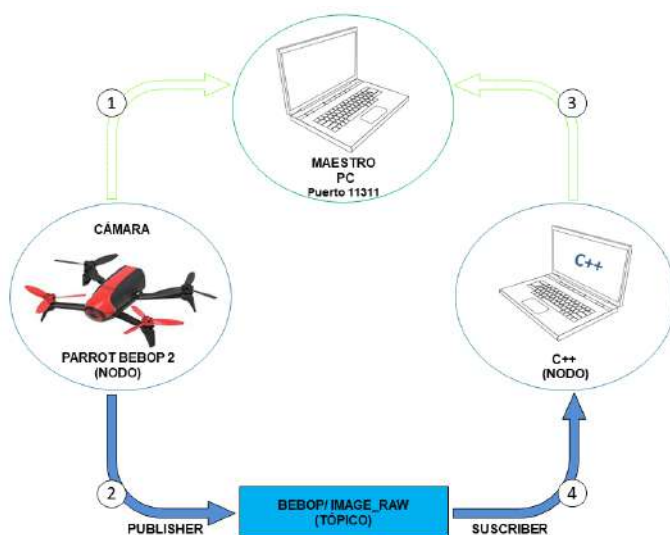


Figura 4.2: Red ROS para la adquisición de imagen.

La cámara del dron notifica al maestro que quiere publicar imágenes en el tópico bebop/image_raw ① y publica la imagen en el tópico requerido ②. Mientras ningún nodo requiera las imágenes almacenadas en el tópico, estas permanecen ahí. Posteriormente C++ notifica al maestro que se quiere suscribir al tópico bebop/image_raw para ver si hay imágenes publicadas ③ y se suscribe ④ para después recibirlas como un mensaje. Finalmente, las imágenes recibidas como mensajes en ROS se convierten con C++ en imágenes que se encuentran en el espacio de color RGB (Figura 4.3).



Figura 4.3: Imágenes recibidas en el espacio RGB.

4.1.2. Procesamiento de la imagen

El procesamiento de la imagen, que consiste en su segmentación, es una etapa muy importante para el desarrollo del presente trabajo, ya que a partir de su resultado se obtiene la información necesaria para detectar el objetivo. El tipo de segmentación utilizada es la segmentación de imagen basada en intensidad de color, cuya finalidad es aislar los pixeles que no correspondan al objetivo a detectar (objetivo de color rojo).

El primer paso es realizar la descomposición de la imagen en sus respectivas componentes RGB. A la componente R se le restan las componentes G y B con la finalidad de poder detectar el color rojo ya que, si el valor de los pixeles de la componente R son mayores a la suma de los pixeles de las componentes G y B, la imagen resultante muestra pixeles con el valor de la resta realizada. Si los pixeles de la componente R son menores a la suma de los pixeles de las componentes G y B, los pixeles de la imagen resultante toman el valor de cero. La operación descrita anteriormente se puede ver en la siguiente ecuación:

$$Pixel_resultante = \begin{cases} 0, & \text{si } Resta \leq 0 \\ Resta, & \text{si } Resta > 0 \end{cases} \quad (4.1.1)$$

En la Figura 4.4 se puede ver un ejemplo de la resta realizada entre las componentes de una imagen, donde se toman dos pixeles como muestra. El pixel 1 tiene un valor de 220 para la componente R, 54 para la componente G y 66 para la componente B. El pixel 2 tiene 120 para R, 139 para G y 103 para B. Se realiza la operación de resta (Resta1 para pixel 1 y Resta2 para pixel 2) y se obtiene lo siguiente:

$$\begin{aligned} Resta1 &= 220 - (54 + 66) = 100 \\ Resta2 &= 120 - (139 + 103) = -122 \end{aligned}$$

De acuerdo a la ecuación 4.1.1, como Resta1 es mayor que 0 el Pixel_resultante1 toma el valor de Resta1 y como Resta2 es menor que 0 el Pixel_resultante2 toma el valor de 0.

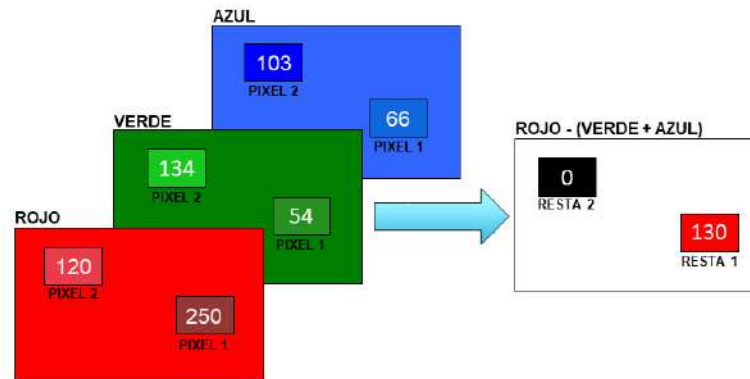
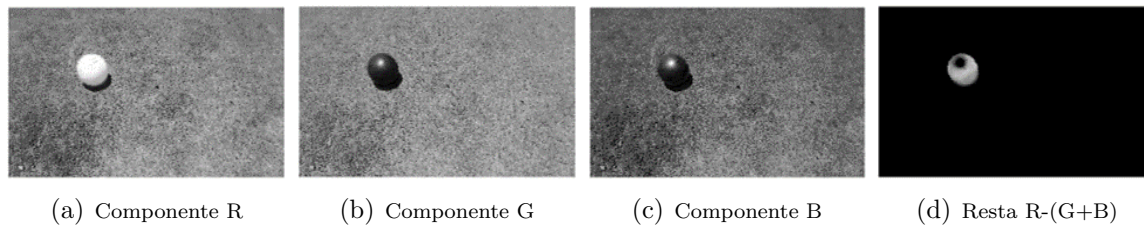


Figura 4.4: Ejemplo de resta de canales de una imagen.

El resultado de la operación de resta entre las componentes de la imagen adquirida por el dron se puede ver en la Figura 4.5.



(a) Componente R

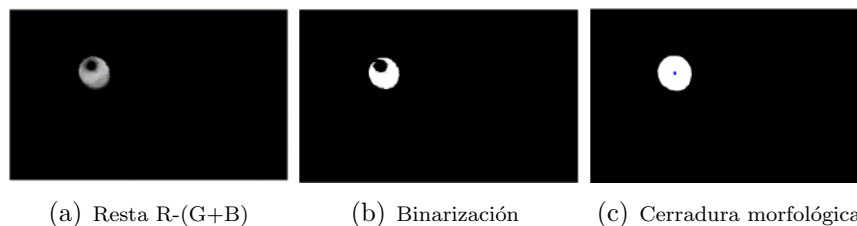
(b) Componente G

(c) Componente B

(d) Resta R-(G+B)

Figura 4.5: Resta de componentes de la imagen adquirida por el dron.

Después de la resta de las componentes, la imagen resultante (Figura 4.6a) se binariza (Figura 4.6b) y se hace pasar por las operaciones morfológicas de dilatación y erosión, lo que en procesamiento de imagen se conoce como cerradura morfológica. Finalmente, se obtiene el centroide de la imagen, el cual nos proporciona la posición del objetivo dentro de la imagen (Figura 4.6c).



(a) Resta R-(G+B)

(b) Binarización

(c) Cerradura morfológica

Figura 4.6: Centroides del objetivo.

4.2. Implementación de estimadores estocásticos

La estimación de estados es necesaria para poder obtener una predicción del estado siguiente de un sistema a partir del estado actual. De acuerdo al sistema con el que se trabaje, ya sea lineal o no lineal, se define el estimador de estados a utilizar.

En el presente trabajo se implementaron el filtro Kalman (para sistemas lineales), el filtro Kalman Extendido y el filtro Kalman Unscented (para sistemas no lineales) en Matlab y se realizaron pruebas con diferentes trayectorias del objetivo.

4.2.1. Implementación del KF

Para la implementación del Filtro Kalman (KF) se considera la predicción de estados de una trayectoria lineal uniforme. La ecuación de proceso está dada por:

$$\begin{bmatrix} x_k \\ y_k \\ \Delta x_k \\ \Delta y_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \Delta x_{k-1} \\ \Delta y_{k-1} \end{bmatrix} + w \quad (4.2.1)$$

donde k es la imagen, x, y es la posición del objetivo, $\Delta x, \Delta y$ es el incremento en la posición del objetivo, y w es el ruido de proceso que está normalmente distribuido con $N(0, Q)$. La ecuación de medida está dada por:

$$\begin{bmatrix} x_{mk-1} \\ y_{mk-1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \Delta x_{k-1} \\ \Delta y_{k-1} \end{bmatrix} + v \quad (4.2.2)$$

donde x_m, y_m es la medida de la posición del objetivo y v es el ruido de medida que está normalmente distribuido con $N(0, R)$. La medida de la posición del objetivo está dada por el centroide que se obtiene del procesamiento de la imagen recibida por el dron.

Las matrices iniciales de covarianza \mathbf{P}_0 , de covarianza del ruido de proceso \mathbf{Q}_0 y de covarianza de ruido de medida \mathbf{R}_0 que se consideran para el filtro Kalman son:

$$\mathbf{P}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{Q}_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{R}_0 = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}$$

4.2.2. Implementación del EKF y el UKF

En el Filtro Kalman Extendido (EKF) y el Filtro Kalman Unscented (UKF) se considera un movimiento no lineal del objetivo que se describe por la siguiente ecuación de proceso:

$$\begin{bmatrix} x_k \\ y_k \\ r_k \\ \theta_k \\ \dot{r}_k \\ \dot{\theta}_k \end{bmatrix} = \begin{bmatrix} x_{k-1} + [r_{k-1} + (\dot{r}_{k-1} + \alpha) \Delta t] \cos \left[\theta_{k-1} + \left(\dot{\theta}_{k-1} + \Omega \right) \Delta t \right] \\ y_{k-1} + [r_{k-1} + (\dot{r}_{k-1} + \alpha) \Delta t] \sin \left[\theta_{k-1} + \left(\dot{\theta}_{k-1} + \Omega \right) \Delta t \right] \\ r_{k-1} + (\dot{r}_{k-1} + \alpha) \Delta t \\ \theta_{k-1} + \left(\dot{\theta}_{k-1} + \Omega \right) \Delta t \\ \dot{r}_{k-1} + \alpha \\ \dot{\theta}_{k-1} + \Omega \end{bmatrix} \quad (4.2.3)$$

donde k , x , y representan lo mismo que en 4.2.1, r representa el desplazamiento total en x , y , θ el desplazamiento angular, \dot{r} la velocidad r , $\dot{\theta}$ la velocidad de θ , α el ruido de proceso en \dot{r} y Ω representa el ruido de proceso en $\dot{\theta}$. De acuerdo a lo anterior, el ruido de proceso está dado por:

$$w_k = \begin{bmatrix} \alpha \\ \Omega \end{bmatrix}$$

el cual está normalmente distribuido con $N(0, Q)$. La ecuación de medida está dada por:

$$\begin{bmatrix} x_{mk} \\ y_{mk} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ r_{k-1} \\ \theta_{k-1} \\ \dot{r}_{k-1} \\ \dot{\theta}_{k-1} \end{bmatrix} + v_k \quad (4.2.4)$$

donde x_m , y_m y v_k representan lo mismo que en el KF.

Las matrices iniciales de covarianza \mathbf{P}_0 , de covarianza del ruido de proceso \mathbf{Q}_0 y de covarianza de ruido de medida \mathbf{R}_0 que se consideran para el EKF y el UKF son:

$$\mathbf{P}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{Q}_0 = \begin{bmatrix} 8 & 0 \\ 0 & 4 \end{bmatrix}, \quad \mathbf{R}_0 = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}$$

4.2.3. Selección de estimador estocástico

La selección del estimador estocástico es uno de los puntos más importantes en este trabajo, ya que permite tener una mejor estimación del estado del objetivo lo que conlleva a una mejor capacidad de reacción del dron ante una situación presentada que le permita mantener al objetivo dentro del campo de visión.

Para seleccionar el estimador estocástico adecuado se consideran diferentes recorridos del objetivo que permiten comparar a los estimadores propuestos. Esta comparación se lleva a cabo utilizando el RMSE (Root Mean Square Error), el cual es una medida estadística de la diferencia entre la estimación y la medida actual observada [30]. El RMSE está definido por:

$$\mathbf{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (Xm_i - Xe_i)^2}$$

donde N es el número total de muestras, Xm_i es la medida observada y Xe_i es la medida estimada.

4.3. Control Reactivo

En el sistema propuesto para el desarrollo del presente trabajo se implementa un control reactivo también conocido como control de lazo abierto, con el fin de que el dron responda de forma directa a los valores de entrada asignados. Los valores de

referencia que pueden generar un movimiento en el dron son en roll, pitch, yaw y vertical rate y, de acuerdo a las características del quadrotor utilizado los valores de velocidad se encuentran entre los límites $[-1 \ 1]$.

4.3.1. Primer enfoque: Objetivo visto de frente por el dron

Para implementar el primer enfoque propuesto, que consiste en ver de frente al objetivo para mantenerlo dentro del campo de visión de la cámara a bordo del dron, se considera el movimiento en roll, el cual se genera mediante un control reactivo (Figura 4.7) y conlleva al vuelo autónomo del dron con valores de pitch, yaw y vertical rate constantes.



Figura 4.7: Control de lazo abierto (control reactivo) para movimientos en roll del dron.

El campo de visión del dron está limitado como se muestra en la Figura 4.8 y su movimiento sólo se lleva a cabo cuando el objetivo sale del límite 1 y se detiene cuando está dentro del límite 0. Cabe mencionar que el objetivo a mantener dentro del campo de visión del dron es una persona usando ropa de color rojo y que la pelota se utilizó para determinar el tipo de procesamiento de acuerdo al color y para seleccionar el estimador estocástico adecuado (Figura 4.8).

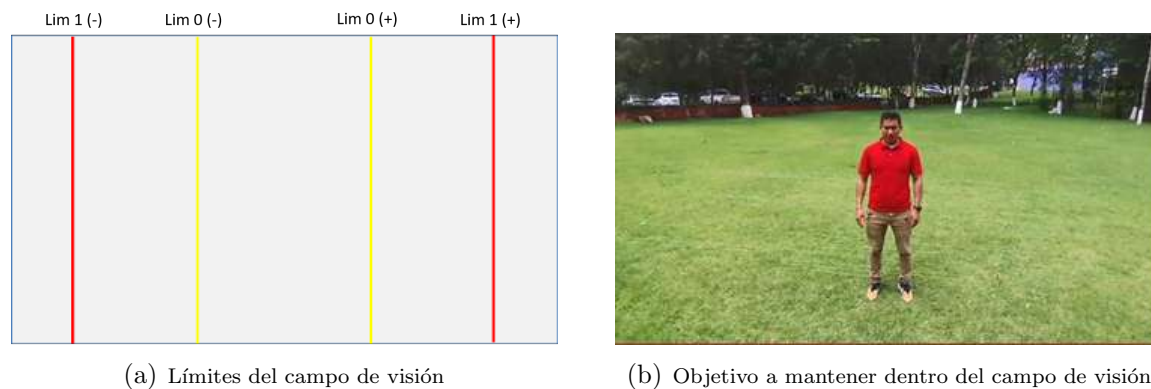


Figura 4.8: Límites del campo de visión de la cámara a bordo del dron para que el objetivo sea visto de frente

El comportamiento del sistema está basado en el diagrama de flujo que se muestra en la Figura 4.9 y depende tanto de la imagen capturada por el dron como de la respuesta del Filtro Kalman Unscented. El diagrama se describe a continuación:

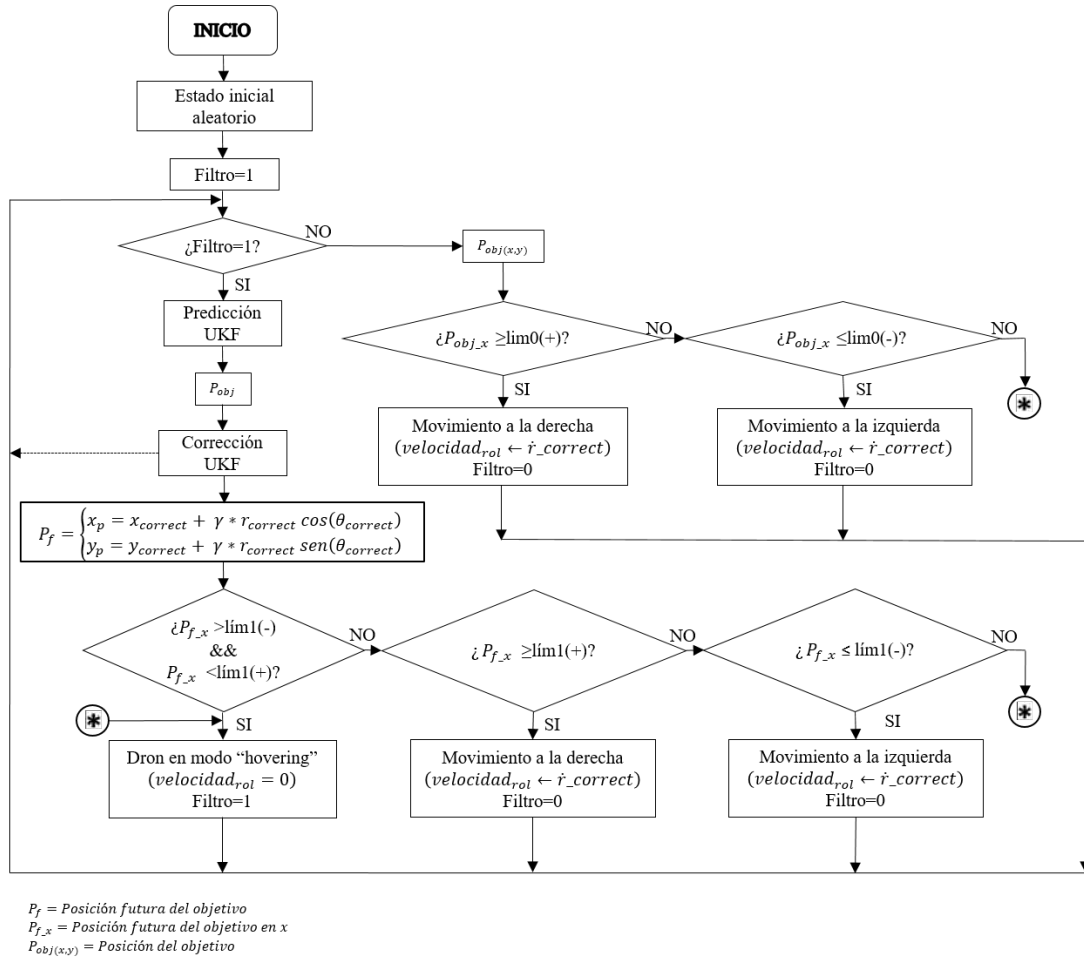


Figura 4.9: Diagrama de flujo del sistema para mantener a un objetivo dentro del campo de visión de la cámara a bordo del dron visto desde el frente.

El UKF tiene un vector de estados que está compuesto de estados observables como la posición en x (x) y la posición en y (y) y de estados no observables como la distancia del punto actual al siguiente (r), el desplazamiento angular (θ), la velocidad de incremento de la posición (\dot{r}) y la velocidad angular ($\dot{\theta}$). El algoritmo propuesto para el control reactivo se inicializa con un vector de estados aleatorio, por lo que asignamos el centro del tamaño de la imagen adquirida a la posición en x y y y al resto de las variables se les asignó cero como valor inicial. El UKF puede estar activado con filtro igual a uno o puede estar desactivado con filtro igual a cero. Si el filtro

está activado, el UKF predice el estado y la medición siguientes a partir del estado actual (para la primera predicción, el estado actual es el estado declarado como estado inicial). Posteriormente, se obtiene la medición de la posición del objetivo a partir del procesamiento de la imagen adquirida por el dron y el UKF corrige la predicción realizada. El estado corregido pasa a ser el estado actual.

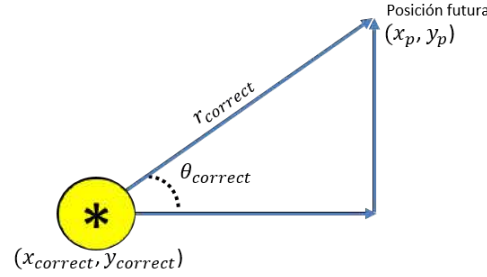


Figura 4.10: Posición futura estimada del objetivo.

Del vector de estado corregido se toman la posición $(x_{correct}, y_{correct})$, la distancia del punto actual al siguiente $(r_{correct})$ y el desplazamiento angular $(\theta_{correct})$, y se calcula la posición futura estimada del objetivo (Figure 4.10) como sigue:

$$Posición\ futura = \begin{cases} x_p = x_{correct} + \gamma * r_{correct} \cos(\theta_{correct}) \\ y_p = y_{correct} + \gamma * r_{correct} \sin(\theta_{correct}) \end{cases} \quad (4.3.1)$$

donde γ es un factor que representa que tan lejos se calcula la posición futura, en nuestro caso le asignamos valor igual a 10. Esta posición futura es la que permite al dron reaccionar de forma anticipada al objetivo y con ello poder mantenerlo dentro de su campo de visión.

Si la posición futura en x calculada se encuentra dentro del límite 1, no es necesario que el dron se mueva, por lo tanto permanece en modo “hovering” y el filtro sigue activo. Sin embargo, si la posición futura en x es menor al límite 1(-), el dron se debe mover a la izquierda con una velocidad proporcional a la velocidad de incremento de la posición corregida $\dot{r}_{correct}$ y se deja de mover hasta que la posición del objetivo sea mayor al límite 0(-). Si la posición futura en x es mayor al límite 1(+), el dron se debe mover a la derecha con una velocidad proporcional a la velocidad de incremento de la posición corregida $\dot{r}_{correct}$, y se deja de mover hasta que la posición del objetivo sea menor al límite 0(+). Mientras el dron se esté moviendo, el UKF se desactiva.

4.3.2. Segundo enfoque: Objetivo visto desde arriba por el dron

Para el desarrollo del segundo enfoque, que consiste en que el objetivo sea visto desde arriba para mantenerlo dentro del campo de visión de la cámara a bordo del dron, se considera el movimiento en roll y en pitch que se generan mediante un control reactivo (Figura 4.11) y que permiten el vuelo autónomo del dron.



Figura 4.11: Control de lazo abierto (control reactivo) para movimientos en roll y pitch del dron.

El campo de visión para que la cámara del dron pueda ver al objetivo desde arriba se encuentra limitado como lo muestra la Figura 4.12.

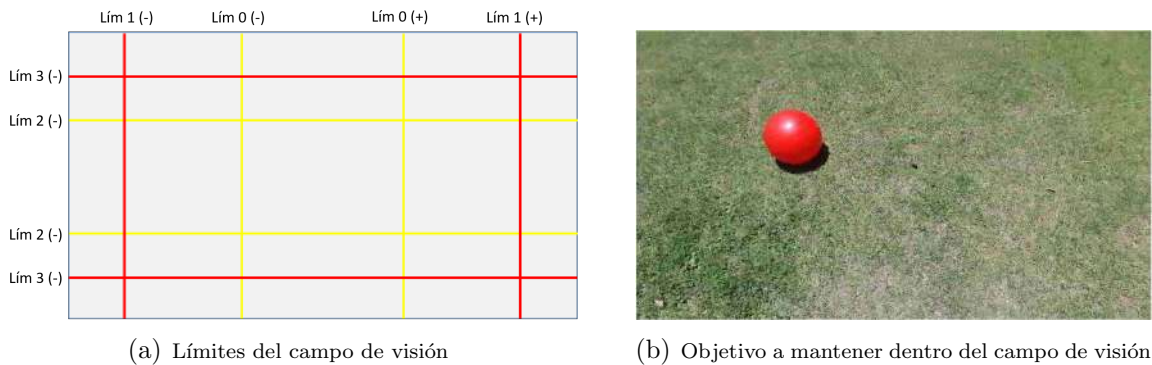


Figura 4.12: Límites del campo de visión de la cámara a bordo del dron para que el objetivo sea visto desde arriba

El objetivo a mantener dentro del campo de visión de la cámara a bordo del dron es la pelota roja utilizada para el procesamiento de imagen. Los movimientos del dron en pitch y roll se basan en el diagrama de flujo que se muestra en la Figura 4.13 y depende del procesamiento de la imagen capturada por la cámara a bordo del dron y de la respuesta del UKF.

De acuerdo a la respuesta de la etapa de corrección del UKF se calcula una posición futura según la ecuación 4.3.1 que permite al dron reaccionar de forma anticipada

al objetivo, tanto en pitch como en roll. Si la posición futura en x del objetivo se encuentra dentro del límite 1 o sale de él, el dron reacciona de acuerdo a lo descrito en la sección anterior. Si la posición futura en y se encuentra dentro del límite 3 el dron no se debe mover, permanece en modo “hovering” y el filtro sigue activo. Si la posición futura en y es menor al límite 3(-) el dron se debe mover en pitch hacia adelante con una velocidad proporcional a la velocidad de incremento de la posición corregida $\dot{r}_{correct}$ y se deja de mover hasta que la posición del objetivo sea mayor al límite 2(-). Si la posición futura en y es mayor al límite 3(+) el dron se debe mover en pitch hacia atrás con una velocidad proporcional a la velocidad de incremento de la posición corregida $\dot{r}_{correct}$ y se deja de mover hasta que la posición del objetivo sea menor al límite 2(+). Mientras el dron se esté moviendo, el UKF permanece desactivado.

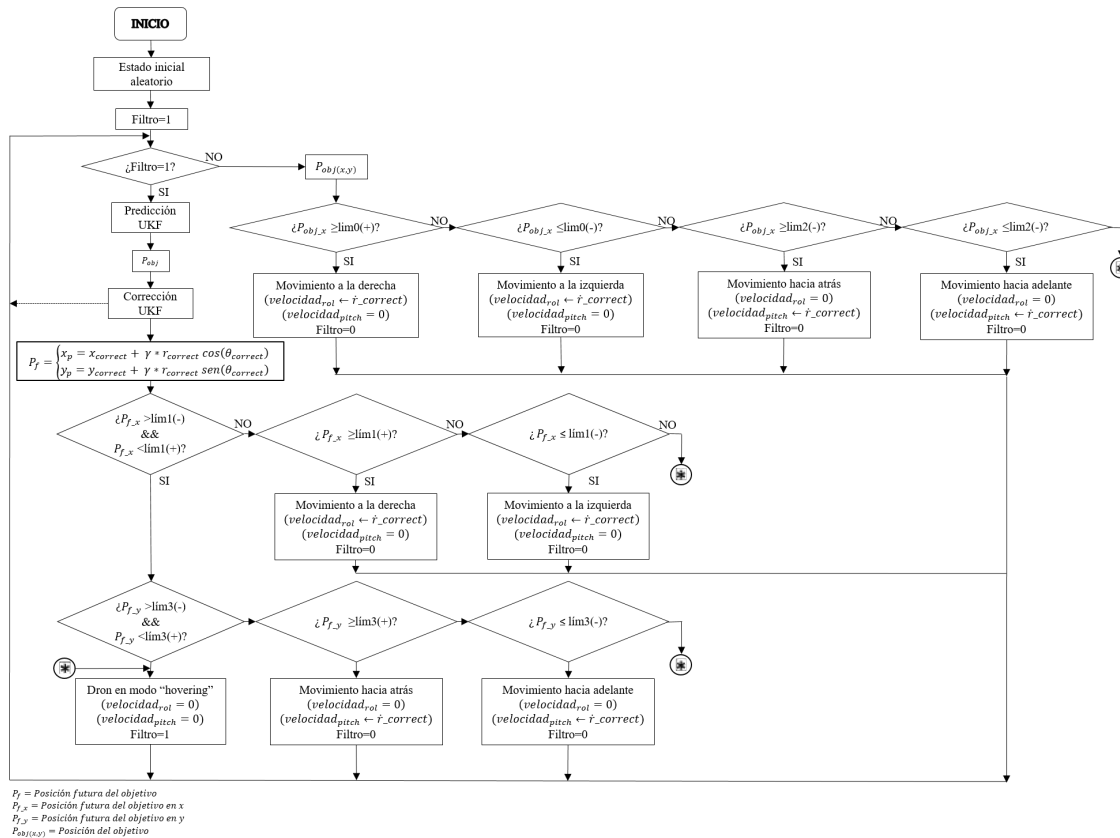


Figura 4.13: Diagrama de flujo del sistema para mantener a un objetivo dentro del campo de visión de la cámara a bordo del dron visto desde arriba.

Pruebas experimentales y resultados

En este capítulo se presentan los resultados de la implementación de cada uno de los estimadores estocásticos descritos en el Capítulo de Metodología así como la selección del más adecuado de acuerdo a los recorridos propuestos. Además se presentan los resultados de los dos enfoques propuestos para el vuelo autónomo de un dron que permiten mantener a un objetivo dentro del campo de visión de la cámara que tiene a bordo.

5.1. Resultados de la implementación del KF, EKF y UKF

La selección del estimador estocástico adecuado es un punto muy importante para el desarrollo de este trabajo, ya que de ello depende la precisión de la respuesta del dron ante las situaciones que le permitan mantener a un objetivo dentro del campo de visión de la cámara que tiene a bordo. Para poder realizar esta selección primero se implementaron los filtros (KF, EKF y UKF) de forma independiente.

La Figura 5.1 muestra la respuesta del KF implementado en Matlab, tanto la predicción como la corrección, así como la medida que se obtiene con el centroide del objetivo con el fin de ver de forma gráfica la respuesta del KF comparada con la medida real. Posteriormente, en la Tabla 5.1 se presenta el RMSE (Root Mean Square Error) de la predicción y corrección del KF para la distancia r , la posición en x y la posición en y comparados con la medida real del objetivo.

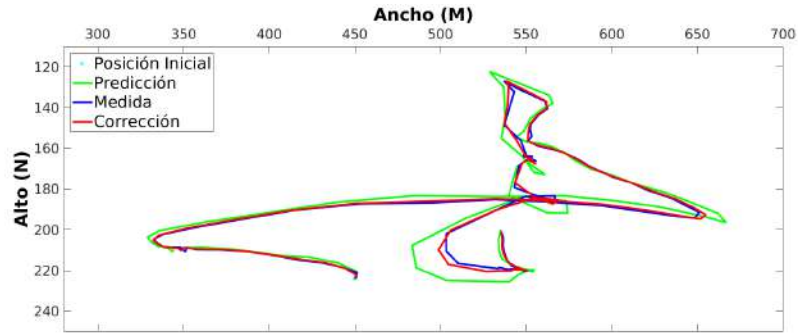


Figura 5.1: Predicción y Corrección del KF comparadas con la medida real de la posición del objetivo.

Tabla 5.1: RMSE de la respuesta del KF.

Filtro Kalman (KF)	RMSE		
	Distancia r	Posición en x	Posición en y
Predicción KF	2.8755	2.7266	1.5307
Corrección KF	1.3860	1.3134	0.7391

Al igual que en el KF, la Figura 5.2 muestra la respuesta del EKF implementado en Matlab y la Tabla 5.2 presenta el RMSE.

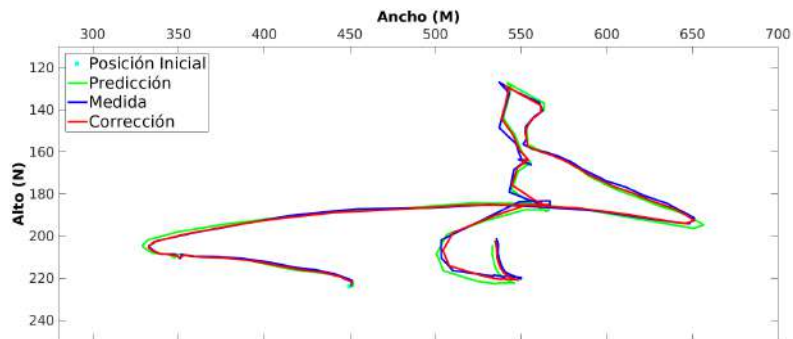


Figura 5.2: Predicción y Corrección del EKF comparadas con la medida real de la posición del objetivo.

Tabla 5.2: RMSE de la respuesta del EKF.

Filtro Kalman Extendido (EKF)	RMSE		
	Distancia r	Posición en x	Posición en y
Predicción EKF	3.7357	3.5131	2.0549
Corrección EKF	1.9564	1.7824	1.2167

Finalmente, el resultado de la implementación del UKF en Matlab se puede ver en la Figura 5.3, tanto su predicción como corrección, además de la medida real del objetivo. Al igual que en el KF y el EKF, también se presenta el RMSE en la Tabla 5.3.

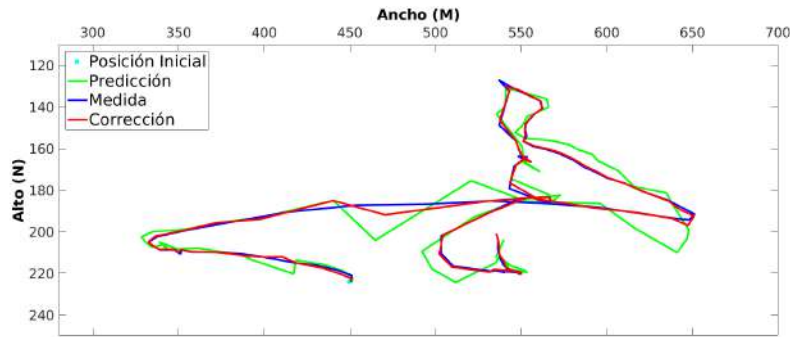


Figura 5.3: Predicción y Corrección del UKF comparadas con la medida real de la posición del objetivo.

Tabla 5.3: RMSE de la respuesta del UKF.

Filtro Kalman Unscented (UKF)	RMSE		
	Distancia r	Posición en x	Posición en y
Predicción UKF	3.5448	3.2895	2.0700
Corrección UKF	1.4799	1.3598	0.8914

5.1.1. Resultados de la selección de estimador estocástico

La selección del estimador estocástico se lleva a cabo después de realizar la comparación entre el KF, EKF y UKF con diferentes recorridos del objetivo. Para ello, en este trabajo se consideran tres recorridos diferentes que se obtienen con la medida de la posición del objetivo y que se muestran en las Figuras 5.4, 5.5 y 5.6, donde además se puede observar la respuesta del KF, EKF y UKF.

Para cada recorrido propuesto se presenta una tabla con los resultados del RMSE de la distancia r , la posición en x y la posición en y del objetivo comparados con la medida real para cada uno de los filtros implementados

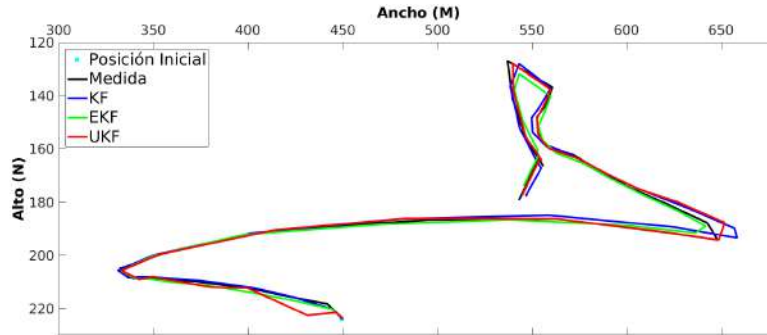


Figura 5.4: Comparación entre el KF, EKF y UKF con primer recorrido de prueba del objetivo.

Tabla 5.4: RMSE para el KF, EKF y UKF con primer recorrido del objetivo.

Estimador Estocástico (Primer recorrido)	RMSE		
	Distancia r	Posición en x	Posición en y
KF	2.1678	2.0789	1.0509
EKF	2.6957	2.4313	1.5709
UKF	1.8221	1.7140	1.0330

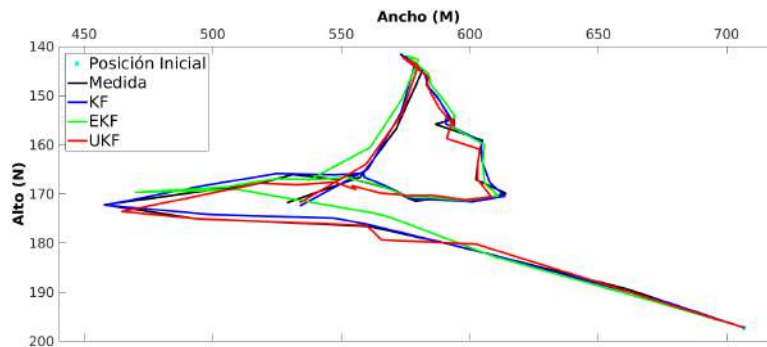


Figura 5.5: Comparación entre el KF, EKF y UKF con segundo recorrido de prueba del objetivo

Tabla 5.5: RMSE para el KF, EKF y UKF con segundo recorrido del objetivo.

Estimador Estocástico (Segundo recorrido)	RMSE		
	Distancia r	Posición en x	Posición en y
KF	2.0447	1.9974	0.8797
EKF	2.3029	2.1926	1.2421
UKF	1.8585	1.7142	1.0767

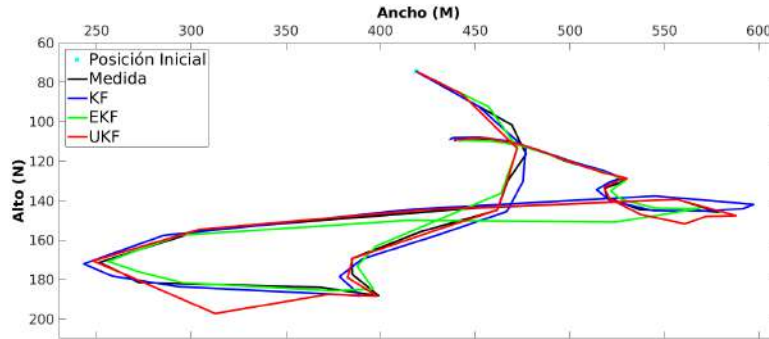


Figura 5.6: Comparación entre el KF, EKF y UKF con tercer recorrido de prueba del objetivo

Tabla 5.6: RMSE para el KF, EKF y UKF con tercer recorrido del objetivo.

Estimador Estocástico (Tercer recorrido)	RMSE		
	Distancia r	Posición en x	Posición en y
KF	2.9455	2.8938	1.1493
EKF	2.9864	2.8180	1.6995
UKF	2.3727	2.2196	1.4549

A partir de los resultados presentados en la Tabla 5.4, 5.5 y 5.6 se calcula el RMSE promedio de cada filtro, quedando como resultado lo que se muestra en la siguiente tabla:

Tabla 5.7: RMSE promedio de los tres recorridos del objetivo para el KF, EKF y UKF.

Estimador Estocástico	RMSE Promedio		
	Distancia r	Posición en x	Posición en y
KF	2.1678	2.0789	1.0509
EKF	2.6957	2.4313	1.5709
UKF	1.8221	1.7140	1.0330

Con los resultados anteriores se decide trabajar con el UKF, el cual nos da una posición más aproximada a la posición real del objetivo, de acuerdo a las trayectorias realizadas por el mismo (en su mayoría con un comportamiento no lineal).

5.2. Resultados para objetivo visto de frente

El primer enfoque propuesto consiste en mantener a un objetivo que se encuentra frente al dron dentro del campo de visión de la cámara que tiene a bordo y después de su implementación se obtienen las respuestas en roll que se muestran en la Figura 5.7 y que corresponden a las imágenes tomadas por el dron. Para ver el comportamiento externo del sistema, en la Figura 5.8 se tienen imágenes de cada uno de los movimientos del dron que corresponden a las respuestas del enfoque propuesto.

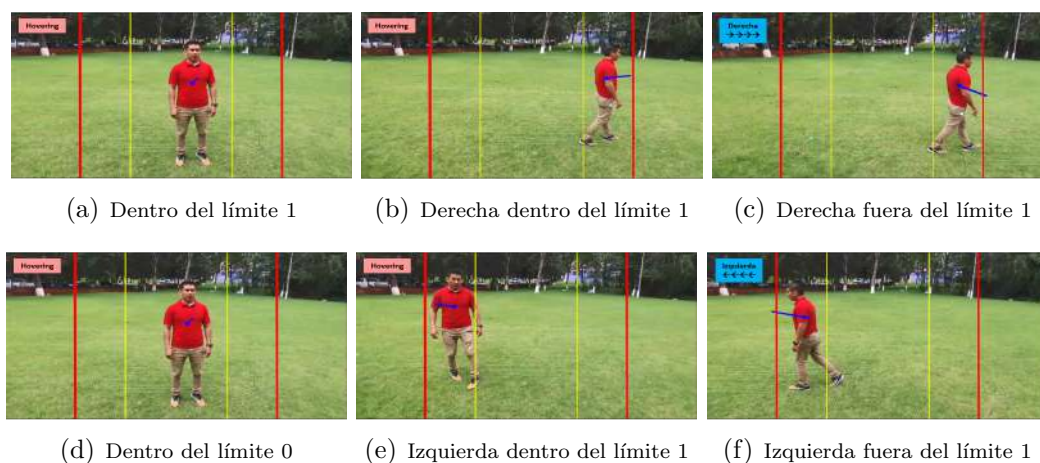


Figura 5.7: Respuesta del primer enfoque propuesto observada desde el dron cuando el objetivo está en diferentes posiciones.

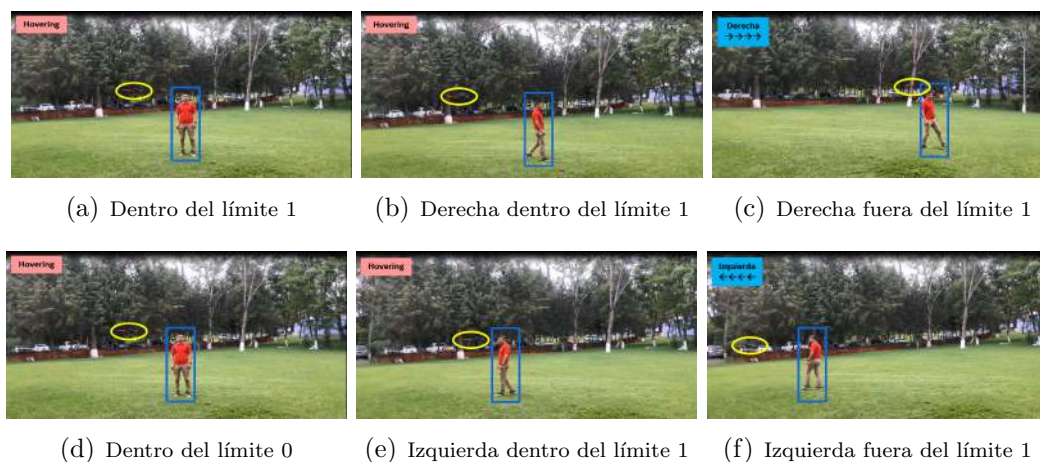


Figura 5.8: Respuesta del dron al primer enfoque propuesto observada desde un punto externo cuando el objetivo está en diferentes posiciones.

Si el objetivo inicialmente se encuentra dentro del límite 1, el dron permanece en modo “hovering” (Figura 5.7a), si el objetivo se mueve a la derecha pero permanece dentro del límite 1 el dron continúa en modo “hovering” (Figura 5.7b) y si el objetivo se mueve a la derecha y sale del límite 1 el dron se mueve a la derecha (Figura 5.7c) hasta que el objetivo se encuentre dentro del límite 0 (Figura 5.7d). Si el objetivo se mueve a la izquierda pero permanece dentro del límite 1 el dron permanece en modo “hovering” (Figura 5.7e) y si el objetivo se mueve a la derecha y sale del límite 1 el dron se mueve a la derecha (Figura 5.7f) hasta que el objetivo se encuentre dentro del límite 0.

Los límites para el primer enfoque propuesto están definidos previamente en el Capítulo de Metodología y se muestran en la Figura 4.8.

5.2.1. Comparación entre el vuelo autónomo implementado y el vuelo a control remoto realizado por un usuario con el objetivo visto de frente

Para validar el primer enfoque propuesto se consideraron tres trayectorias diferentes (trayectoria 1, trayectoria 2, trayectoria 3 (Figura 5.9)) y se hizo una comparación entre el Vuelo Autónomo (VA) implementado y el Vuelo a Control Remoto (VCR) ejecutado por un usuario. Para el VCR, cada trayectoria es realizada por un usuario diferente. La respuesta del dron es considerada correcta si este responde de forma adecuada cuando el objetivo sale del límite 1, es decir, si el objetivo sale del límite 1(-) el dron se debe mover a la izquierda, si el objetivo sale del límite 1(+) el dron se debe mover a la derecha y si el dron no sale del límite 1 el dron se debe mantener en modo “hovering”.

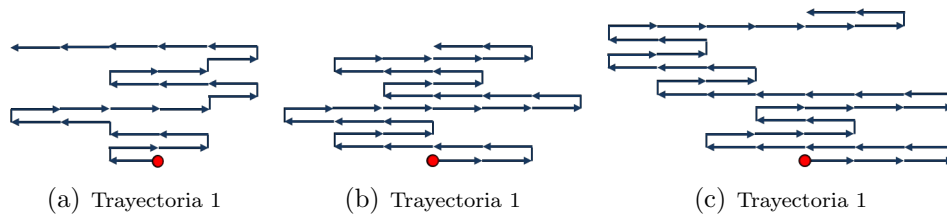


Figura 5.9: Trayectorias de referencia para evaluar el primer enfoque propuesto.

Durante la trayectoria 1, con el VA el dron responde de forma correcta en 10 ocasiones de 10 que el objetivo sale del límite 1, mientras que con el VCR sólo responde

de forma correcta en 7 ocasiones. Para la trayectoria 2, con el VA el dron responde de forma correcta en 9 ocasiones de 10 que el objetivo sale del límite 1, mientras que con el VCR responde de forma correcta en 8 ocasiones. Finalmente, para la trayectoria 3 ambos enfoques responden de forma correcta en 10 ocasiones de 10 que el objetivo sale del límite 1. Los resultados descritos anteriormente se expresan en porcentajes en la Tabla 5.9.

Tabla 5.8: Comparación entre el sistema de Vuelo Autónomo (VA) del primer enfoque propuesto y el Vuelo a Control Remoto (VCR) ejecutado por un usuario

	VA	VCR
Trayectoria 1	100.00 %	70.00 %
Trayectoria 2	90.00 %	80.00 %
Trayectoria 3	100.00 %	100.00 %
Tasa de éxito	96.60 %	83.30 %

El vuelo autónomo implementado en este enfoque corre en tiempo real a una tasa de 30 fps que corresponde a la velocidad de transmisión de la cámara a bordo del dron hacia la Estación de Control en Tierra.

5.3. Resultados para objetivo visto desde arriba

El segundo enfoque propuesto consiste en mantener a un objetivo visto desde arriba dentro del campo de visión de la cámara a bordo del dron. Después de su implementación se obtienen las respuestas en roll y pitch que se muestran en la Figura 5.10. Para este enfoque propuesto el objetivo se debe mantener dentro del límite 1 y el límite 3.

Los límites para el segundo enfoque propuesto están definidos previamente en el Capítulo de Metodología y se muestran en la Figura 4.12.

El comportamiento externo del sistema se puede ver desde una grabación externa, sin embargo, debido a que el dron está arriba del objetivo a una altura considerable no se alcanzan a distinguir de forma clara sus movimientos en una imagen.



Figura 5.10: Respuesta del segundo enfoque propuesto observada desde el dron cuando el objetivo está en diferentes posiciones.

Si el objetivo inicialmente se encuentra dentro del límite 1 y el límite 3, el dron permanece en modo “hovering” (Figura 5.10a), si el objetivo se mueve a la derecha pero permanece dentro del límite 1 y el límite 3 el dron continua en modo “hovering” (Figura 5.10b) y si el objetivo se mueve a la derecha y permanece dentro del límite 3 pero sale del límite 1 el dron se mueve a la derecha (Figura 5.10c) hasta que el objetivo se encuentre dentro del límite 0 (Figura 5.10d). Si el objetivo se mueve a la izquierda pero permanece dentro del límite 1 y el límite 3 el dron continua en modo “hovering” (Figura 5.10e) y si el objetivo se mueve a la izquierda y permanece dentro del límite 3 pero sale del límite 1 el dron se mueve a la izquierda (Figura 5.10f) hasta que el objetivo se encuentre dentro del límite 0 (Figura 5.10g). Si el objetivo se mueve hacia adelante pero permanece dentro del límite 1 y el límite 3 el dron continua en

modo “hovering” (Figura 5.10h) y si el objetivo se mueve hacia adelante y permanece dentro del límite 1 pero sale del límite 3 el dron se mueve hacia adelante (Figura 5.10i) hasta que el objetivo se encuentre dentro del límite 2 (Figura 5.10j). Finalmente, si el objetivo se mueve hacia atrás pero permanece dentro del límite 1 y el límite 3 el dron continua en modo “hovering” (Figura 5.10k) y si el objetivo se mueve hacia atrás y permanece dentro del límite 1 pero sale del límite 3 el dron se mueve hacia atrás (Figura 5.10l) hasta que el objetivo se encuentre dentro del límite 2.

5.3.1. Comparación entre el vuelo autónomo implementado y el vuelo a control remoto realizado por un usuario con el objetivo visto desde arriba

Para validar el segundo enfoque propuesto se consideraron tres trayectorias diferentes del objetivo (trayectoria 4, trayectoria 5, trayectoria 6 (Figura 5.11)) y se hizo una comparación entre el Vuelo Autónomo (VA) implementado y el Vuelo a Control Remoto (VCR) ejecutado por un usuario. Para el VCR, cada trayectoria es realizada por un usuario diferente. La respuesta del dron es considerada satisfactoria si este responde de forma adecuada cuando el objetivo sale del límite 1 o del límite 3, es decir, si el objetivo sale del límite 1(-) el dron se debe mover a la izquierda, si el objetivo sale del límite 1(+) el dron se debe mover a la derecha, si el objetivo sale del límite 3(-) el dron se debe mover hacia adelante, si el objetivo sale del límite 3(+) el dron se debe mover hacia atrás y si el dron se encuentra dentro de los límites 1 y 3 se debe mantener en modo “hovering”.

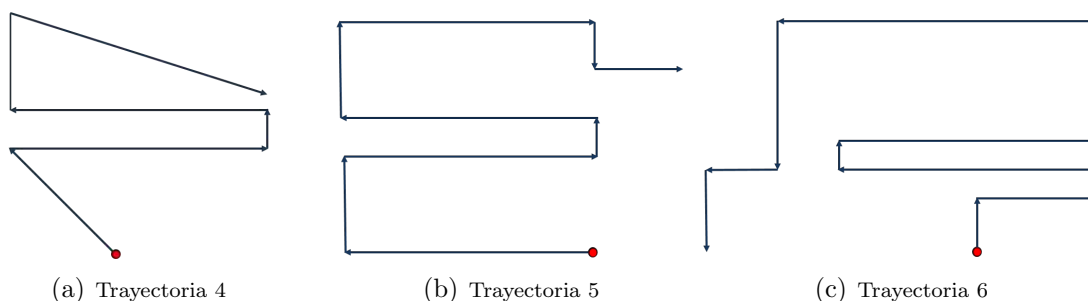


Figura 5.11: Trayectorias de referencia para evaluar el segundo enfoque propuesto.

Durante la trayectoria 4, con el VA el dron responde de forma correcta en 10 ocasiones de 12 que el objetivo sale de los límites indicados, mientras que con el VCR

responde de forma correcta en 11 ocasiones. Para la trayectoria 5, con el VA el dron responde de forma correcta en 10 ocasiones de 10 que el objetivo sale de los límites, mientras que con el VCR responde de forma correcta en 9 ocasiones. Finalmente, para la trayectoria 6 el VA responde de forma correcta en 13 ocasiones de 14 que el objetivo se sale de los límites y el VCR en 12 ocasiones. Los resultados descritos anteriormente se expresan en porcentajes en la Tabla 5.9.

Tabla 5.9: Comparación entre el sistema de Vuelo Autónomo (VA) del segundo enfoque propuesto y el Vuelo a Control Remoto (VCR) ejecutado por un usuario

	VA	VCR
Trayectoria 4	83.30 %	91.66 %
Trayectoria 5	100.00 %	90.00 %
Trayectoria 6	92.85 %	85.71 %
Tasa de éxito	92.05 %	89.13 %

5.4. Sumario

En este capítulo se presentaron los resultados de la metodología propuesta, desde los estimadores estocásticos y su selección, hasta los dos enfoques propuestos para el desarrollo del presente trabajo. A partir de los resultados de la comparación entre el sistema de vuelo autónomo implementado y el vuelo a control remoto ejecutado por un usuario se puede decir que el sistema propuesto puede ser aplicado en tiempo real y facilitar las labores humanas en donde se requiera.

Conclusiones

En esta tesis se presentó un sistema que permite mantener de forma autónoma a un objetivo dentro del campo de visión de la cámara a bordo de un dron. El trabajo presentado se compone de diferentes etapas que son: la adquisición y el procesamiento de la imagen, implementación de estimadores estocásticos, selección del estimador estocástico adecuado y definición de políticas de navegación del dron.

Se utilizó el sistema operativo ROS para establecer comunicación entre el dron y la estación remota en tierra que permite adquirir las imágenes del objetivo con la cámara a bordo del dron para su posterior procesamiento con el lenguaje de programación C++ a fin de que el sistema sea eficiente.

Para la detección del objetivo se realizó la segmentación de la imagen basada en color. Debido a que el color a detectar fue el rojo, se realizó la resta de canales de la imagen RGB adquirida, posteriormente se binarizó y se le aplicaron operaciones morfológicas que permitieron obtener el centroide del objetivo, el cual proporciona su posición dentro de la imagen.

De acuerdo a la literatura, se implementaron tres estimadores estocásticos: el Filtro Kalman, el Filtro Kalman Extendido y el Filtro Kalman Unscented y partir de los resultados obtenidos de la comparación entre ellos con diferentes recorridos del objetivo, se tomó la decisión de trabajar con el Filtro Kalman Unscented para estimar estados observables y no observables y predecir una posición futura del objetivo que permite al dron anticipar su movimiento.

Para establecer las políticas de navegación del dron se propusieron dos enfoques:

el primero consiste en mantener a un objetivo visto de frente dentro del campo de visión de la cámara a bordo del dron de forma autónoma y el segundo enfoque consiste en mantener a un objetivo dentro del campo de visión de la cámara a bordo del dron visto desde arriba de forma autónoma. Para la implementación del primer enfoque se consideró el movimiento en roll del dron si la posición futura del objetivo se sale del límite establecido, de lo contrario el dron permanece en modo “hovering”. Para el segundo enfoque además del movimiento en roll también se consideró el movimiento en pitch del dron, ya que como el objetivo es visto desde arriba, este también se puede mover hacia atrás y hacia adelante y salir de los límites establecidos.

Finalmente, para la evaluación de los enfoques presentados se realizaron pruebas en tiempo real obteniendo resultados satisfactorios comparados con el vuelo realizado por pilotos humanos, es por ello, que el sistema puede ser implementado en áreas donde se requiera mantener a un objetivo dentro del campo de visión de la cámara a bordo de un dron con las ventajas de que no necesita un piloto humano y disminuye la presentación de riesgo por errores.

6.1. Limitaciones

El trabajo presentado en esta tesis se encuentra limitado a detectar objetivos de color rojo específicamente, además de que la comunicación entre la estación de control en tierra y el dron, en diversas ocasiones, presentó interferencia que no permitió una buena toma de la imagen.

6.2. Trabajo a futuro

Como trabajo a futuro se plantea implementar a bordo del dron el sistema presentado en este trabajo con una mejora en el procesamiento de la imagen adquirida a fin de que, a partir de características específicas del objetivo, el dron sea capaz de mantenerlo dentro del campo de visión de la cámara que tiene a bordo con la mayor eficiencia posible y la mínima interferencia en la comunicación.

Apéndice

Productos obtenidos

Keeping a Moving Target within the Field of View of a Drone's Onboard Camera via Stochastic Estimation

Beatriz Hernandez-Hernandez¹, Jose Martinez-Carranza² and Jose Rangel-Magdaleno³

Abstract—The use of Drones in areas such as cinema, sports, social events and even video selfies has been increasing due to their flexibility to capture video in scenarios where there is an interest to keep a target within the field of view of the drones onboard camera. In order to remove the dependence of the pilot that controls the drone, in this work we present a system for an autonomous flight control of the drone with the goal of keeping a target within the field of view of its onboard camera. For the latter, the images were captured by a camera onboard the drone, whose output was combined with a stochastic estimator of the target states, based on the Unscented Kalman Filter, to generate control commands so that the drone performs such recording autonomously. The system was validated with real-time tests involving different targets moving with different trajectories and compared against human pilots. Our approach kept the target within the field of view with a 96.6% of success compared to 83.3% of success obtained by human pilots. The latter indicates that our approach has the potential to be used in applications where autonomous drones could be used for aerial video recording, with a special interest in keeping a target within the field of view of the drone's camera.

I. INTRODUCTION

In recent years, the use of Micro Aerial Vehicles (MAVs), also known as drones, has become more important because of its low cost, its size, and especially the application they have in different areas of industry and everyday life. One of the applications that is taking relevance today is the use of drones for video capture and photography in both industry and recreational applications. The increase in its use has had a considerable social impact by allowing new ways of producing film and television, and even for the acquisition of personal photographs, e.g., selfies or video selfies.

Motivated by the above, in this paper we investigate how to leverage applications involving drones whose goal is that of capturing video or photographs of a target in an autonomous manner. Thus, in this work, what is sought is for the drone to execute two tasks: i) to remain in *hovering* when the target remains within the field of view, even if the target moves, as long as it remains in the field of view the drone execute hovering; ii) to execute autonomous flight in order to keep the target within its field of view.

To address the above, we propose a system that allows a visual estimation of the target states from the information

captured by the camera onboard the drone. These states include the position of the target in the image, its speed and the direction of movement. From these data, drone speed control rules are built, which enable the drone to react to the situations mentioned above, i.e., the target remains within the field of view or the target moves and is about to leave the field of view. For the former, the controller will demand the drone to execute hovering, for the latter the controller will command the drone to move in the direction of the target with a speed proportional to the one of target's motion.

To assess our proposed autonomous system, we carried out a comparison of its performance against that of human pilots with different level of skills and for the target moving along different paths.

To present our proposed approach in detail, this paper has been organised as follows: section II revises relevant related work related to autonomous flight, target tracking and autonomous flight and with special emphasis on cinematic video recording applications; section III describes the target detection system based on processing of video imagery acquired by the onboard camera; IV describes the state estimators that were implemented and from which the best estimator was selected in order to estimate the target's states; V reports the implementation of the reactive control that decides the movement of the drone according to the target's movement; VI describes our experimental framework; finally, our conclusions are discussed in section VII.

II. RELATED WORK

The work presented in this paper addresses different disciplines focused in drones, among them are: control for autonomous flight, target tracking, and autonomous flight for cinema type recordings. In this section a literature review corresponding to the mentioned disciplines is performed.

A. Drone control for autonomous flight

In the literature there is a great number of works that focus on the control of a drone to fly autonomously and can be used in different applications, as shown by Zulu and John in [1], where they perform an analysis of different control algorithms for the autonomous flight of a drone that allows to obtain the advantages and disadvantages of each one of them. However, in many cases it is necessary to make sophisticated decisions in autonomous vehicles, which is why, in [2], Veres et al. present a paper that shows the fundamentals that allow a machine to be autonomous, in addition to presenting different methodologies which can carry out the type of decisions mentioned. We present an approach based on

¹Beatriz Hernandez-Hernandez, Student at the Instituto Nacional de Astrofísica Óptica y Electrónica, Puebla, Mexico, email: bettyhh@inaoep.mx

²Jose Martinez-Carranza, Associate Professor in the Computer Science Department of the Instituto Nacional de Astrofísica Óptica y Electrónica, Puebla, Mexico, email: carranza@inaoep.mx

³Jose Rangel-Magdaleno, Full time Professor with the Electronics Department, Instituto Nacional de Astrofísica Óptica y Electrónica, Puebla, Mexico, email: jrangel@inaoep.mx

reactive control for the drone's autonomous flight, which reacts while the stimulating condition is present, otherwise the drone's reaction disappears.

B. Autonomous tracking of a target with a drone

The targets tracking throughout history has taken great importance in different areas, which is why there are works that employ different approaches to carry it out. In [3], Vyskovsky designed and implemented a method for the autonomous tracking of a target by adapting the tracking-learning-detection (TLD) algorithm introduced by Kalal [4]. In [5] Cuevas et al. use the Kalman filter and the Kalman filter extended as state estimators to obtain a prediction of the trajectory of the target and to be able to follow up.

C. Autonomous flight for cinema type recording

In the cinematographic field, the use of drones has been increasing for making scenes where humans have difficulty accessing them or for better decision. In [6], Fleureau et al., carry out the autonomous capture of movie scenes in real time with a rotary wing drone however the follow up they realize with planned trajectories. Joubert et al., [7] developed a semi-autonomous system with a drone camera that should be positioned in relation to people in a scene according to the rules of cinematography, seeking to capture visually pleasing images of the activity is carrying out but is not able to respond aggressively to the movement of people.

In our approach, the target is kepted within the field of view of the drone obeying the reactive control that depends on the prediction of the Kalman Unscented filter, previously selected as the best state estimator compared to the Kalman filter and the extended Kalman filter which has the ability to respond aggressively to the movement of the target since it is based on a non-linear movement.

III. VISION BASED DETECTION SYSTEM

This work uses a vision based detection system which is based on the target's color and centroid. This system has been divided into two main stages: the acquisition of the image and its processing as described below.

A. Image acquisition

Image acquisition process is performed using the operative system ROS through a network consisting of different nodes and a master, as shown Fig. 1.

For the present work the nodes are: one that corresponds to the camera of the drone for the taking of the image and one of C++ that allows to visualize and store the image. The acquired images by the drone are in the RGB color space.

B. Image processing

In this stage the segmentation of the image based on color intensity is performed, whose purpose is to isolate the pixels that do not correspond to the target to detect. Since the image is in the RGB color space, the first step is to decompose the image into its respective components R, G, and B. Then, subtract the components G and B to the component R. Later, perform the binarization and the morphological dilation and

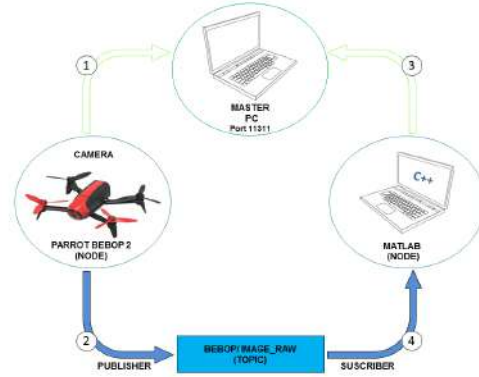


Fig. 1: Image acquisition with ROS.

erosion operations that finally allow to find the position of the target to be detected by its centroid (Fig. 2).

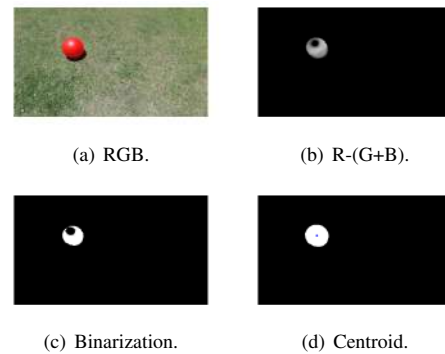


Fig. 2: Image processing.

IV. STOCHASTIC ESTIMATION

A. Kalman Filter

The Kalman Filter (KF) is a set of mathematical equations that provides an efficient recursive environment to estimate the state of a process in such a way as to minimize the mean square error [8].

To describe the KF algorithm it is considering a lineal dynamic system represented by the following equations known as process equation (1) and measurement equation (2):

$$x_k = \mathbf{F}x_{k-1} + w \quad (1)$$

$$y_{k-1} = \mathbf{H}x_{k-1} + v \quad (2)$$

where \mathbf{F} is the transition matrix of the state x_k , from $k-1$ to k , w is the noise of the process that is normally distributed with a mean equal to zero and a covariance matrix \mathbf{Q} , y_{k-1} is the observation state in $k-1$, \mathbf{H} is the measurement matrix, and v is the noise of the measurement that is normally distributed with a mean equal to zero and a covariance matrix \mathbf{R} .

The KF consists of two stages: prediction and correction. In the prediction stage, estimated state and covariance error

are propagated. In the correction stage a measurement made to improve the prediction is incorporated.

B. Extended Kalman Filter

All systems have a degree of non-linearity and depending on the type of process and its operation region can be approximated with a linear model, where the KF can be used as a state estimator. However, in some cases this linear approximation is not adequate enough, then it is necessary a state estimator that uses a nonlinear model of the system and the most common is the Extended Kalman Filter (EKF).

The EKF solves the problem of estimating the state generated by a nonlinear system using the Taylor series expansion that approximates the nonlinear state and observation equation, over the estimated current value of the state with a first order precision; similarly, it provides an estimate of the minimum state variance based on the statistical information model [9].

To describe the EKF algorithm it is considering a nonlinear dynamic system represented by the following equations,

$$x_k = \mathbf{f}[x_{k-1}, w] \quad (3)$$

$$y_{k-1} = \mathbf{h}[x_{k-1}, v] \quad (4)$$

where \mathbf{f} is the nonlinear dynamic model of the system from $k-1$ to k , w is the noise of the process that is normally distributed with a mean equal to zero and a covariance matrix \mathbf{Q} , y_{k-1} is the observation state in $k-1$, \mathbf{h} is the measurement matrix, and v is the noise of the measurement that is normally distributed with a mean equal to zero and a covariance matrix \mathbf{R} . The EKF, like the KF is composed of a prediction stage and a correction stage.

C. Unscented Kalman Filter

The Unscented Kalman Filter (UKF) is a state estimator that has a behavior equivalent to KF, but is focused on nonlinear systems without the linearization steps required by the EKF.

The state distribution is represented by a Gaussian random variable (GRV) as in the EKF, but is now specified using a minimum set of carefully selected sampling points. These sampling points completely capture the true mean and covariance of the GRV, and when they are propagated through a nonlinear system, they capture the posterior mean and covariance with a 3rd order precision, according to Taylor expansion, for any non-linearity [10].

To understand the UKF, it is first necessary to explain the unscented transformation (UT), which is a method for calculating the statistic of a random variable undergoing a nonlinear transformation. This method is based on the intuitive idea that it is easier to approximate a Gaussian distribution than to approximate an arbitrary nonlinear function [11].

Given a random variable x of dimension L , with mean \bar{x} and covariance \mathbf{P}_x and a second random variable y that is related to x through a nonlinear function

$$y = f(x)$$

to obtain the mean of y , \bar{y} and its covariance \mathbf{P}_y it is necessary to generate a sigma points matrix with $2L+1$ sigma vectors χ_i and its respective weights W_i according to:

$$\begin{aligned} \chi_0 &= \bar{x} \\ \chi_i &= \bar{x} + \left(\sqrt{(L+\lambda)\mathbf{P}_x} \right)_i & i = 1, \dots, L \\ \chi_i &= \bar{x} + \left(\sqrt{(L+\lambda)\mathbf{P}_x} \right)_{i-L} & i = L+1, \dots, 2L \\ W_0^m &= \lambda / (L+\lambda) \\ W_0^c &= \lambda / (L+\lambda) + (1 - \alpha^2 + \beta) \\ W_i^m &= W_i^c = 1/2 \{ (L+\lambda) \} & i = 1, \dots, 2L \end{aligned}$$

where $\lambda = \alpha^2(L+k) - L$ is a scaling parameter. α determines the points dispersion around \bar{x} (usually a small positive value, e.g. $1 \leq \alpha \leq 1e-4$, k is a secondary scaling parameter (usually 0 or $3-L$) and β is a parameter that incorporates a priori knowledge of the \mathbf{x} (for Gaussian distributions $\beta = 0$). $\left(\sqrt{(L+\lambda)\mathbf{P}_x} \right)_i$ is the i -th column of the root of the square matrix [10]. These sigma vectors are propagated through the nonlinear function, then a cloud of transformed points is obtained,

$y_i = f(\chi_i) \quad i = 0, \dots, 2L$
which their statistics are calculated and with their respective calculated weights the mean and covariance is obtained.

UKF can be considered as the result of incorporating the UT to the EKF to improve the approximations that are made of the first two moments of a random variable that results from propagating another random variable through a nonlinear transformation [12]. In Algorithm (1), the equations of the UKF that assume a nonlinear model of the system represented by process equation (3) and measurement equation (4), are presented.

D. Stochastic estimator implementation

The three stochastic estimators described in the previous sections were implemented in this work and a comparison is made among them, in order to select the one with the least mean square error (RMSE), whose results are observed in section VI. For the KF it is consider a uniform linear trajectory described by the following process equation:

$$\begin{bmatrix} x_k \\ y_k \\ \Delta x_k \\ \Delta y_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \Delta x_{k-1} \\ \Delta y_{k-1} \end{bmatrix} + w$$

where k is the frame, x is the target position in x , y is the target position in y , Δx is the increment in x of the target position, Δy is the increment in y of the target position and w is the noise of the process which is normally distributed with $N(0, Q)$. The measurement equation is given by:

$$\begin{bmatrix} x_{mk-1} \\ y_{mk-1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \Delta x_{k-1} \\ \Delta y_{k-1} \end{bmatrix} + v$$

where x_m is the measurement of the position in x of the target, y_m is the measurement of the position in y of the target and v is the noise of the measurement which is normally distributed with $N(0, R)$.

Initialize with:	$\hat{x}_0 = E[x_0]$ $\mathbf{P}_0 = E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T]$ $\hat{x}_0^a = E[x^a] = [\hat{x}_0^T \quad \mathbf{0} \quad \mathbf{0}]$ $\mathbf{P}_0^a = E[(x_0^a - \hat{x}_0^a)(x_0^a - \hat{x}_0^a)^T] = \begin{bmatrix} \mathbf{P}_0 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R} \end{bmatrix}$
For $k \in \{1, \dots, \infty\}$ Calculate sigma points:	$\mathbf{x}_{k k-1}^i = [\hat{x}_{k-1}^i \quad \hat{x}_{k-1}^i \pm \sqrt{(L+\lambda)\mathbf{P}_{k-1}^i}]$
Prediction State	$\mathbf{x}_{k k-1}^x = \mathbf{F}[\mathbf{x}_{k-1}^x, \mathbf{x}_{k-1}^w]$ $\hat{x}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \mathbf{x}_{i,k k-1}^x$
Covariance	$\mathbf{P}_k^- = \sum_{i=0}^{2L} W_i^{(c)} [\mathbf{x}_{i,k k-1}^x - \hat{x}_k^-] [\mathbf{x}_{i,k k-1}^x - \hat{x}_k^-]^T$
Measurement	$\mathbf{y}_{k k-1} = \mathbf{H}[\mathbf{x}_{k k-1}^x, \mathbf{x}_{k-1}^v]$ $\hat{y}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \mathbf{y}_{i,k k-1}$
Correction Innovation covariance	$\mathbf{P}_{\hat{y}_k \hat{y}_k} = \sum_{i=0}^{2L} W_i^{(c)} [\mathbf{y}_{i,k k-1} - \hat{y}_k^-] [\mathbf{y}_{i,k k-1} - \hat{y}_k^-]^T$ $\mathbf{P}_{\mathbf{x}_k \mathbf{y}_k} = \sum_{i=0}^{2L} W_i^{(c)} [\mathbf{x}_{i,k k-1}^x - \hat{x}_k^-] [\mathbf{y}_{i,k k-1} - \hat{y}_k^-]^T$
Filter Gain	$\mathbf{K} = \frac{\mathbf{P}_{\mathbf{x}_k \mathbf{y}_k}}{\mathbf{P}_{\hat{y}_k \hat{y}_k}}$
State	$\hat{x}_k = \hat{x}_k^- + \mathbf{K}(\mathbf{y}_k - \hat{y}_k^-)$
Covariance	$\hat{\mathbf{P}}_k = \mathbf{P}_k^- - \mathbf{K} \mathbf{P}_{\hat{y}_k \hat{y}_k} \mathbf{K}^T$
where $x^a = [x^T \quad w^T \quad v^T]^T$, $\mathbf{x}^a = [(\mathbf{x}^x)^T \quad (\mathbf{x}^w)^T \quad (\mathbf{x}^v)^T]^T$, λ =scaling parameter, L =dimension of augmented state, \mathbf{Q} =process noise cov., \mathbf{R} =measurement noise cov., W_i =weights.	

Algorithm 1: Unscented Kalman Filter equations [10]

For the EKF and UKF a nonlinear movement of the target is assumed, thus we use the following process model:

$$\begin{bmatrix} x_k \\ y_k \\ r_k \\ \theta_k \\ \dot{r}_k \\ \dot{\theta}_k \end{bmatrix} = \begin{bmatrix} x_{k-1} + [r_{k-1} + (\dot{r}_{k-1} + \alpha)\Delta t] \cos[\theta_{k-1} + (\dot{\theta}_{k-1} + \Omega)\Delta t] \\ y_{k-1} + [r_{k-1} + (\dot{r}_{k-1} + \alpha)\Delta t] \sin[\theta_{k-1} + (\dot{\theta}_{k-1} + \Omega)\Delta t] \\ r_{k-1} + (\dot{r}_{k-1} + \alpha)\Delta t \\ \theta_{k-1} + (\dot{\theta}_{k-1} + \Omega)\Delta t \\ \dot{r}_{k-1} + \alpha \\ \dot{\theta}_{k-1} + \Omega \end{bmatrix}$$

where k , x , y represent the same as in the KF process equation, r represents the total displacement in x and y , θ the angular displacement, \dot{r} the speed of r , $\dot{\theta}$ the speed of θ , α the process noise in \dot{r} and Ω represents the process noise in $\dot{\theta}$. According to the above, the process noise is:

$$w_k = \begin{bmatrix} \alpha \\ \Omega \end{bmatrix}$$

which is normally distributed with $N(0, Q)$. The measure-

ment equation is given by:

$$\begin{bmatrix} x_{mk} \\ y_{mk} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ r_{k-1} \\ \theta_{k-1} \\ \dot{r}_{k-1} \\ \dot{\theta}_{k-1} \end{bmatrix} + v_k$$

where x_m , y_m and v_k represent the same as in the KF.

V. REACTIVE CONTROL

The Parrot BEBOP drone which is manufactured by Parrot is used in this study and is a low cost platform with four rotors symmetrically mounted. When you change the speed of the rotors you get a desired effect that causes the movement, and can be in pitch, roll, yaw or altitude.



Fig. 3: Open loop control.

In order to keep a target within the field of vision of the drone is realized the movement in roll for which is implemented a reactive control, that also is known like control of open loop (Fig. 3), and that takes to the autonomous flight. The open loop control is one in which the output has no effect on the control action, i.e. the output is not compared to the reference input. Thus, each reference input corresponds to a fixed operating condition [13].

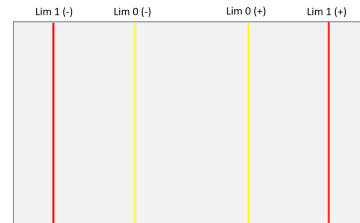


Fig. 4: Field of view limits.

The field of view of the drone is limited as shown in Figure 4. Its movement is only carried out when the target leaves the limit 1 and stops moving until it is within the limit 0. It should be mentioned that the target to keep within the field of vision of the drone, is a person wearing red clothes and that the ball seen in Fig. 2 is only used to determine the type of processing performed on the image and to select the stochastic estimator. The behavior of the system is based on the flow diagram presented in Figure 5 and depends on the image captured by the onboard camera of the drone and the response of the selected stochastic estimator.

VI. EXPERIMENTS

The experimental framework is divided in two stages: the first consists of a comparison of the stochastic estimators described in section IV; the second one consists of the implementation in C++ of the autonomous flight system described in section V, as much as a comparison of such

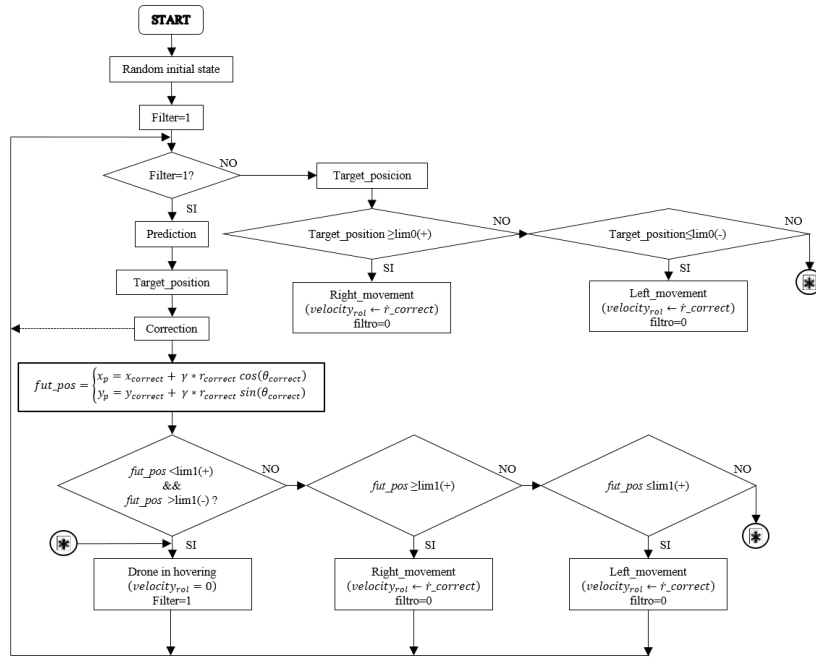


Fig. 5: Flowchart of system behaviour.

system against the performance of remote control flight executed by different users.

A. Comparison of stochastic estimators

The behavior of the implemented stochastic estimators can be observed in Figure 6 and in Table I a comparison is made between them by the RMSE that is derived from the difference between the measured real value (reference) and the value obtained by each filter.

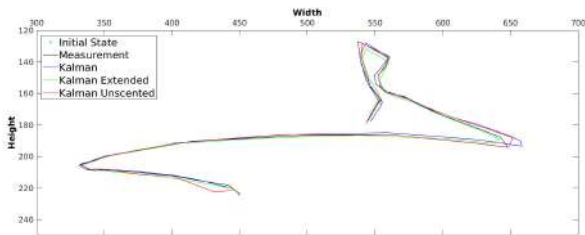


Fig. 6: Trajectory comparison for KF, EKF, UKF and reference.

TABLE I: Error RMSE of distance (r) and position (x, y) for KF, EKF and UKF.

Algorithm	RMSE		
	r	x	y
KF	2.1678	2.0789	1.0509
EKF	2.6957	2.4313	1.5709
UKF	1.8221	1.7140	1.0330

From the values of the table it can be observed that the UKF is the estimator that presents less variations with respect to the reference, this means that it is more adjusted to the

real values. It is for this reason that the decision is made to work with the UKF to implement the reactive control in the drone to keep a target within its field of vision autonomously.

B. Response of the reactive control

When the reactive control is implemented, three different responses can be expected: drone movement to the right, drone movement to the left and drone to hovering mode (positive role, negative role and zero role respectively) and in Figure 7 the three frames are shown which correspond to each of them.



Fig. 7: Examples of Drone's motion when the target leaves the field of view.

To validate our approach, three different target's trajectories are considered and a comparison is made between the implemented autonomous flight (AF) and the remote controlled flight (RCF) executed by a user. For the RCF, each trajectory is performed by a different user, defined as beginner, intermediate and advanced, according to their ease of flight control, for trajectories 1, 2 and 3, respectively. The drone's response is considered correct if it responds properly when the target leaves the limit 1, i.e. if the target leaves the left of the limit1 (-) the drone must move to the left, if it goes to the right of the limit1 (+) the drone must move to the right, otherwise it must be kept in hovering mode (Figure 8).

During trajectory 1, with the AF the drone responds correctly in 10 occasions out of 10 that the target leaves



(a) Autonomous flight with the drone moving to the left.



(b) Autonomous flight with the drone moving to the right

Fig. 8: Autonomous flight with our proposed methodology for keeping the target within the field of view of the onboard camera: the first and third rows show the onboard camera's field of view; the second and fourth rows show an external view where the drone can be seen flying in the corresponding direction to keep the target in within its field of view.

the limit1 while with the RCF only responds correctly in 7 occasions. For trajectory 2, with the AF the drone responds correctly 9 times out of 10 that the target leaves the limit1 while with the RCF it responds correctly in 8 occasions. Finally, for trajectory 3, Both approaches respond correctly 10 occasions out of 10 that the target leaves the limit1. The above-described results expressed in percentages are shown in Table II. Finally, we should highlight that our autonomous flight system run in real time at a frame rate of 30 fps, which is the transmission speed of the onboard camera to the Ground Control Station.

TABLE II: Comparison between our proposed Autonomous Flight System (AF) against Remote Controller Flight by a user (RCF).

	AF	RCF
Trajectory 1	100%	70%
Trajectory 2	90%	80%
Trajectory 3	100%	100%
Successful rate	96.6%	83.3%

VII. CONCLUSIONS

We have presented a methodology that enables a drone to keep a moving target within the field of view in an autonomous manner, by only processing the video images captured by the onboard camera. In our approach, we have implemented three stochastic estimators: The Kalman Filter, the Extended Kalman Filter and the Unscented Kalman Filter in order to select the estimator with the best performance results in terms of estimation of the target's states. From the latter, we chose Unscented Kalman Filter to estimate target's position, its velocity and the direction in which the target is moving in the image. These values enable the controller

to command the drone to anticipate the movement of the target autonomously and only when the target is about to leave the field of view. Otherwise, the drone remains in hovering. Finally, real-time experiments of our proposed system were carried out with successful results compared to those obtained by human pilots. A video of experiments can be seen at: <https://youtu.be/URB5MFm2BPw>

Our approach could be used in areas such as television and cinema, where the use of drones has a steady growth, with particular emphasis where a target has been filmed within a specific camera field of view.

ACKNOWLEDGMENT

The first author is supported by the Mexican National Council for Science and Technology (CONACyT) studentship number 702914/582708. This work has been partially funded by the Royal Society through the Newton Advanced Fellowship with reference NA140454.

REFERENCES

- [1] A. Zulu and S. John, "A review of control algorithms for autonomous quadrotors," *arXiv preprint arXiv:1602.02622*, 2016.
- [2] S. M. Veres, L. Molnar, N. K. Lincoln, and C. P. Morice, "Autonomous vehicle control systems a review of decision making," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 225, no. 2, pp. 155–195, 2011.
- [3] T. Krajník, V. Vonásek, D. Fišer, and J. Faigl, "Ar-drone as a platform for robotic research and education," *Research and Education in Robotics-EUROBOT 2011*, pp. 172–186, 2011.
- [4] Z. Kalal, K. Mikolajczyk, and J. Matas, "Face-td: Tracking-learning-detection applied to faces," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pp. 3789–3792, IEEE, 2010.
- [5] E. V. Cuevas, D. Zaldivar, and R. Rojas, "Kalman filter for vision tracking," 2005.
- [6] J. Fleureau, Q. Galvane, F.-L. Tardieu, and P. Guillotel, "Generic drone control platform for autonomous capture of cinema scenes," in *Proceedings of the 2nd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, pp. 35–40, ACM, 2016.
- [7] N. Joubert, D. B. Goldman, F. Berthouzoz, M. Roberts, J. A. Landay, P. Hanrahan, *et al.*, "Towards a drone cinematographer: Guiding quadrotor cameras using visual composition principles," *arXiv preprint arXiv:1610.01691*, 2016.
- [8] G. Bishop and G. Welch, "An introduction to the kalman filter," *Proc of SIGGRAPH, Course*, vol. 8, no. 27599-23175, p. 41, 2001.
- [9] L. Sánchez, J. Ordóñez, and S. Infante, "Filtro de kalman extendido y filtro de partículas kalman extendido para problemas de estimación no lineal," *Revista INGENIERÍA UC*, vol. 20, no. 1, 2013.
- [10] E. A. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pp. 153–158, Ieee, 2000.
- [11] S. J. Julier and J. K. Uhlmann, "A new extension of the kalman filter to nonlinear systems," in *Int. symp. aerospace/defense sensing, simul. and controls*, vol. 3, pp. 182–193, Orlando, FL, 1997.
- [12] A. Pascual, "Ekf y ukf: dos extensiones del filtro de kalman para sistemas no lineales aplicadas al control de un péndulo invertido," *Monografía para el curso: Tratamiento Estadístico de Señales*, p. 35, 2004.
- [13] O. Katsuhiko, "Modern control engineering," 2010.

Bibliografía

- [1] “Drones contra el fuego.” <http://www.leonoticias.com/leon/201605/25/drones-contra-fuego-20160525130244.html>, Agosto 2017.
- [2] “Ordene su dron ajustado a sus necesidades.” <http://dronecenter.blogspot.mx/2015/02/ordene-su-drone-ajustado-sus-necesidades.html>, Agosto 2017.
- [3] “¿Qué se puede hacer con un dron?.” <https://dronografia.es/legal/218-que-se-puede-hacer-con-un-drone.html>, Agosto 2017.
- [4] “Drones al rescate.” <https://billionbytes.es/drones-al-rescate-9049>, Agosto 2017.
- [5] S. M. Veres, L. Molnar, N. K. Lincoln, y C. P. Morice, “Autonomous vehicle control systems—a review of decision making,” *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 225, no. 2, pp. 155–195, 2011.
- [6] Y. Bi y H. Duan, “Implementation of autonomous visual tracking and landing for a low-cost quadrotor,” *Optik-International Journal for Light and Electron Optics*, vol. 124, no. 18, pp. 3296–3300, 2013.
- [7] E. V. Cuevas, D. Zaldivar, y R. Rojas, “Kalman filter for vision tracking,” 2005.
- [8] C.-T. Dang, H.-T. Pham, T.-B. Pham, y N.-V. Truong, “Vision based ground object tracking using ar. drone quadrotor,” en *Control, Automation and Information Sciences (ICCAIS), 2013 International Conference on*, pp. 146–151, IEEE, 2013.

- [9] V. Mariappan, M. Lee, J. Cho, y J. Cha, “Onboard vision based object tracking control stabilization using pid controller,” *International Journal of Advanced Culture Technology (IJACT)*, vol. 4, no. 4, pp. 81–86, 2016.
- [10] Z. Kalal, K. Mikolajczyk, y J. Matas, “Tracking-learning-detection,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012.
- [11] P. Chawla y Nivedita, “Object Tracking in Simulink using Extended Kalman Filter,” *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 4, no. 7, 2015.
- [12] S. J. Julier y J. K. Uhlmann, “A new extension of the kalman filter to nonlinear systems,” en *Int. symp. aerospace/defense sensing, simul. and controls*, vol. 3, pp. 182–193, Orlando, FL, 1997.
- [13] H. Leyton y V, “Desarrollo, implementación y prueba de un filtro de kalman del tipo ukf para un vehículo aéreo no tripulado,” Master’s thesis, Universidad EAFIT, 2009.
- [14] N. B. da Silva, D. B. Wilson, y K. R. Branco, “Performance evaluation of the extended kalman filter and unscented kalman filter,” en *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*, pp. 733–741, IEEE, 2015.
- [15] U. DoD, *Department of Defense Dictionary of Military and Associated Terms*. US Department of Defense Joint Publication, 2015.
- [16] K. P. Valavanis y G. J. Vachtsevanos, *Handbook of unmanned aerial vehicles*. Springer Publishing Company, Incorporated, 2014.
- [17] B. E. Demir, R. Bayir, y F. Duran, “Real-time trajectory tracking of an unmanned aerial vehicle using a self-tuning fuzzy proportional integral derivative controller,” *International Journal of Micro Air Vehicles*, vol. 8, no. 4, pp. 252–268, 2016.
- [18] “About ROS.” www.ros.org/about-ros.html, Junio 2017.
- [19] “ROS Concepts.” www.wiki.ros.org/ROS/Concepts.html, Junio 2017.

- [20] O. Marques, *Practical image and video processing using MATLAB*. John Wiley & Sons, 2011.
- [21] W. Gonzalez y R. E. Woods, “Eddins, digital image processing using matlab,” *Third New Jersey: Prentice Hall*, 2004.
- [22] H.-D. Cheng, X. H. Jiang, Y. Sun, y J. Wang, “Color image segmentation: advances and prospects,” *Pattern recognition*, vol. 34, no. 12, pp. 2259–2281, 2001.
- [23] A. K. Jain, *Fundamentals of digital image processing*. Prentice-Hall, Inc., 1989.
- [24] “MATLAB.” www.mathworks.com, Julio 2017.
- [25] G. Bishop y G. Welch, “An introduction to the kalman filter,” *Proc of SIGGRAPH, Course*, vol. 8, no. 27599-23175, p. 41, 2001.
- [26] L. Sánchez, J. Ordonez, y S. Infante, “Filtro de kalman extendido y filtro de partículas kalman extendido para problemas de estimación no lineal,” *Revista INGENIERÍA UC*, vol. 20, no. 1, 2013.
- [27] E. A. Wan y R. Van Der Merwe, “The unscented kalman filter for nonlinear estimation,” en *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pp. 153–158, Ieee, 2000.
- [28] A. Pascual, “EKF y UKF: dos extensiones del filtro de kalman para sistemas no lineales aplicadas al control de un péndulo invertido,” *Monografía para el curso: Tratamiento Estadístico de Señales*, p. 35, 2004.
- [29] “Parrot BEBOP 2 — Sitio Web Oficial de Parrot.” <https://www.parrot.com/es/drones/parrot-bebop-2#peso-ligero>, Enero 2017.
- [30] C. J. Willmott y K. Matsuura, “Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance,” *Climate research*, vol. 30, no. 1, pp. 79–82, 2005.