

**UNIVERSIDAD POLITÉCNICA DE MADRID**

Escuela Técnica Superior de Ingeniería Aeronáutica y del Espacio



**Análisis y diseño de algoritmos de estimación de trayectorias basados en el paradigma de múltiples modelos**

Tesis Doctoral

**Álvaro Arroyo Cebeira**

*Máster Universitario en Ingeniería Aeronáutica*

2023



Departamento de Sistemas Aeroespaciales, Transporte Aéreo y Aeropuertos

Escuela Técnica Superior de Ingeniería Aeronáutica y del Espacio



**Análisis y diseño de algoritmos de estimación de trayectorias basados  
en el paradigma de múltiples modelos**

Tesis Doctoral

**Álvaro Arroyo Cebeira**

*Máster Universitario en Ingeniería Aeronáutica*

**Mariano Asensio Vicente**

Doctor Ingeniero Aeronáutico

2023





**POLITÉCNICA**

Tribunal nombrado por el Sr. Rector Magfco. de la Universidad Politécnica de Madrid, el día.....de.....de 20....

Presidente: \_\_\_\_\_

Vocal: \_\_\_\_\_

Vocal: \_\_\_\_\_

Vocal: \_\_\_\_\_

Secretario: \_\_\_\_\_

Suplente: \_\_\_\_\_

Suplente: \_\_\_\_\_

Realizado el acto de defensa y lectura de la Tesis el día.....de.....de 20..  
en la E.T.S.I. /Facultad.....

Calificación .....

EL PRESIDENTE

LOS VOCALES

EL SECRETARIO



# Agradecimientos

A mi director, Mariano Asensio Vicente por sus enseñanzas, claridad, paciencia y disponibilidad que me han permitido seguir creciendo y compaginar mi actividad profesional con el doctorado.

A mis padres, que me motivaron a empezar y más importante a continuar el curso de doctorado.

A mis hermanos, que me han cubierto en otras tareas familiares durante estos años.

A Carlos H. Medel, que me dio la oportunidad trabajar a su lado y de aprender de él, tanto de su conocimiento como de su liderazgo. También quería agradecer a Henk Blom por su cercanía y sus ganas insaciables de transmitir su extenso conocimiento.

A mis amigos y compañeros, que han entendido mi dedicación durante estos años.

Álvaro Arroyo Cebeira

Madrid, 2023



# RESUMEN

Esta tesis está dedicada al desarrollo de un simulador de Monte Carlo que permita analizar de forma exhaustiva algoritmos de estimación, así como al diseño y desarrollo de un filtro basado en el paradigma de múltiples modelos que consiga mejorar el seguimiento de blancos no lineales.

Los algoritmos de estimación de blancos aéreos tienen infinidad de aplicaciones en aviación, como control de tráfico aéreo (ATC), misiones de inteligencia, vigilancia, adquisición de blancos y reconocimiento (ISTAR), sistemas de navegación, o incluso en gestión de tráfico aéreo de sistemas aéreos no tripulados (UTM), dónde los UAS (Unmanned Aerial Systems) tienen una mayor maniobrabilidad y por tanto vuelan trayectorias con muchas mas no linealidades.

Para el simulador de Monte Carlo, se proponen modelos de ruido para cada una de las variables de entrada de los algoritmos de estimación, estos modelos simulan el ruido de medición de acuerdo al sensor instalado en la aeronave, basándonos en los estándares de la industria. Más tarde el simulador ejecuta los algoritmos de estimación, y finalmente los compara mediante las métricas seleccionadas y los test de hipótesis sobre el error de estimación.

Respecto a la algoritmia de estimación, la dinámica del sistema, es decir, las trayectorias de los blancos, es no lineal; y la ecuación de medida que utilizan las aeronaves (coordenadas radar) es no lineal también, por lo que esta tesis se centrará sobre filtros de estimación no lineales. El estado del arte en este campo es el sistema ACAS X (Airborne Collision Avoidance Systems), por lo que se hará un profundo análisis de sus algoritmos.

Por último, se presentan las simulaciones de 4 encuentros, para validar y comparar los distintos algoritmos aplicados.



# ABSTRACT

This thesis poses the development of a Monte Carlo simulator allowing a comprehensive analysis of estimation algorithms; as well as the design and development of an estimation algorithm based on the multiple model paradigm which improves the tracking of non-linear targets.

The estimation algorithms of aerial targets have a lot of applications in aviation, such as Air Traffic Control (ATC), Intelligence, Surveillance, Target Acquisition and Reconnaissance (ISTAR) missions, navigation systems, or even UAS (Unmanned Aerial Systems) Traffic Management (UTM), where drones have a greater manoeuvrability, hence they fly highly nonlinear trajectories.

For the Monte Carlo simulator, noise models per input variable are proposed. These models simulate the measurement noise according to the airborne sensor, based on the industry standards. Then the simulator calls the estimation algorithms, and finally a comparison is done based on the chosen metrics and hypothesis testing for the estimation error.

Regarding the estimation algorithms, the dynamic of the system, i.e. the target trajectories, is non linear; and the measurement equation used by the aircraft (radar coordinates) is non linear as well, hence this thesis focuses on nonlinear filters. The state of the art on this field is ACAS X (Airborne Collision Avoidance Systems), so a comprehensive analysis of the algorithm is conducted.

Finally, the simulation results of 4 encounters are presented to validate and compare both algorithms.



# Contenidos

RESUMEN .....	III
ABSTRACT .....	V
CONTENIDOS .....	VII
<b>1. INTRODUCCIÓN .....</b>	<b>11</b>
<b>2. MATERIALES Y MÉTODOS .....</b>	<b>14</b>
<b>3. ESTADO DEL ARTE.....</b>	<b>17</b>
3.1 ACAS X .....	18
3.1.1 <i>Seguimiento del intruso</i> .....	21
<b>4. SIMULADOR DE MONTE CARLO .....</b>	<b>31</b>
4.1 GENERACIÓN DE NÚMEROS PSEUDO-ALEATORIOS.....	32
4.2 MODELOS DE RUIDO.....	36
4.2.1 <i>Modelo auto regresivo de primer orden</i> .....	38
4.2.2 <i>Altitud</i> .....	39
4.2.3 <i>Rumbo</i> .....	54
4.2.4 <i>Distancia oblicua</i> .....	56
4.2.5 <i>Azimut</i> .....	61
<b>5. PARADIGMA DE MULTIPLES MODELOS .....</b>	<b>67</b>
5.1 SISTEMAS ESTOCÁSTICOS HÍBRIDOS .....	76
5.2 CADENAS DE MARKOV .....	77
5.3 MODELOS.....	81
<b>6. MÉTRICAS DE COMPARACIÓN .....</b>	<b>87</b>
6.1 SELECCIÓN DE ENCUENTROS .....	87
6.2 MÉTRICAS.....	88
6.2.1 <i>Tipos de errores</i> .....	88
6.2.2 <i>Test de hipótesis de las regiones de probabilidad</i> .....	92

6.2.3	<i>Test de hipótesis de la diferencia de error</i> .....	93
6.2.4	<i>Test de hipótesis de dos muestras (2 sample t test)</i> .....	96
6.2.5	<i>Test de hipótesis Kolmogorov-Smirnov (test K-S)</i> :.....	97
6.2.6	<i>Kernel Density Estimation (KDE)</i> .....	97
<b>7</b>	<b>RESULTADOS</b> .....	<b>99</b>
7.1	ENCUENTRO 1 .....	101
7.1.1	<i>Geometría</i> .....	101
7.1.2	<i>Métricas</i> .....	103
7.2	ENCUENTRO 2 .....	116
7.2.1	<i>Geometría</i> .....	116
7.2.2	<i>Métricas</i> .....	117
7.3	ENCUENTRO 3 .....	131
7.3.1	<i>Geometría</i> .....	131
7.3.2	<i>Métricas</i> .....	133
7.4	ENCUENTRO 4 .....	146
7.4.1	<i>Geometría</i> .....	146
7.4.2	<i>Métricas</i> .....	147
<b>8</b>	<b>CONCLUSIONES Y FUTURAS LÍNEAS DE TRABAJO</b> .....	<b>161</b>
	<b>BIBLIOGRAFÍA</b> .....	<b>163</b>
	<b>ANEXO A: ESTIMACIÓN BAYESIANA</b> .....	<b>169</b>
3.2	FILTRO DE KALMAN.....	175
3.2	FILTRO DE KALMAN AUMENTADO.....	184
3.3	FILTRO DE KALMAN EXTENDIDO .....	186
3.4	FILTRO DE KALMAN UNSCENTED .....	189
	<b>ANEXO B: MÉTODO DE MONTE CARLO</b> .....	<b>195</b>

## **Acrónimos y siglas**

ACAS: Airborne Collision Avoidance Systems

AKF: Augmented Kalman Filter

CI: Confidence Interval

EKF: Extended Kalman Filter

IMM: Interacting Multiple Model

KDE: Kernel Density Estimation

KF: Kalman Filter

MC: Monte Carlo

MM: Multiple Models

NCT: Nearly Coordinated Turn

NCV: Nearly Constant Velocity

NEES: Normalized Estimation Error Squared

RMSE: Root Mean Square Error

SHS: Stochastic Hybrid Systems

TCAS: Traffic Collision Avoidance Systems

UAS: Unmanned Aerial Systems

UKF: Unscented Kalman Filter

UTM: UAS Traffic Management



# 1. INTRODUCCIÓN

Este capítulo pretende definir el alcance, objetivos, y motivación de esta investigación; por lo tanto, intentaremos responder al que, por qué y para qué de esta tesis.

A la pregunta del que, la respuesta es los algoritmos de estimación, que han sido un tema muy relevante durante los últimos años debido al interés en aplicarlos a problemas no lineales y la fusión de datos. Además de la aparición de los drones en el espacio aéreo lo que implica la necesidad de unos algoritmos de estimación con mejor desempeño dada la mayor agilidad de estos al ejecutar maniobras y su menor tamaño.

Una de las mayores contribuciones a la estimación bayesiana fue el filtro de Kalman en 1960, para estimar sistemas lineales, ya que es una solución óptima analítica al estimador de Bayes. Este filtro fue el adoptado en los algoritmos de evitación de colisiones TCAS II, que son los que actualmente se utilizan en aviación civil, y no ha sido hasta el desarrollo de ACAS X, cuyos estándares se publicaron en 2018, cuando se han introducido filtros no lineales como UKF, pero que no son suficientes para trayectorias altamente no lineales. De aquí surge la motivación de esta tesis de proponer algoritmos de estimación no lineales y compararlos con el estado del arte actual.

A continuación conviene aclarar el contexto y el marco teórico en el que se ubica la investigación, que se trata del tratamiento de medidas para una dinámica (ecuación de estado) y ecuación de medida no lineales, y adicionalmente en un ambiente donde el ruido no se puede simplificar a ruido gaussiano blanco; por tanto el marco teórico es la teoría de la estimación en general, y en concreto la estimación bayesiana.

Durante la justificación de la investigación conviene analizar la importancia (el por qué), la pertinencia y la utilidad de la investigación (el para qué); para ello debemos primero entender la aplicación a la que se dirige esta investigación.

Toda esta técnica de filtrado objeto de investigación durante esta tesis esta aplicada a los algoritmos de evitación de colisiones entre aeronaves, conocidos como ACAS. Por tanto, la importancia de la mejora de las técnicas de filtrado de las medidas de vigilancia radica en el aumento de la exactitud y la mejora de la conciencia situacional de las aeronaves; y consecuentemente el aumento de la seguridad operacional de las aeronaves.

La versión utilizada actualmente (en el momento de publicación de esta tesis) en el espacio aéreo es TCAS II, pero la última versión de los sistemas ACAS es ACAS X, que consta de las siguientes variantes:

ACAS Xa: es la versión de ACAS X de propósito general, es decir, el sucesor de TCAS II, su estándar fue publicado en 2018 por la RTCA [1] y [2], y por EUROCAE [3], aunque su aprobación regulatoria aún está pendiente.

ACAS Xo: es una extensión de ACAS Xa para operaciones especiales como aproximaciones paralelas a un mismo aeropuerto.

ACAS Xu: variante de ACAS X para para RPAS/UAS (Remotely Piloted Aircraft Systems/ Unmanned Aerial Systems), esta lógica incorpora resolución de conflictos en el plano horizontal. Su estándar fue publicado en 2020 por la RTCA [4] y EUROCAE [5].

ACAS Xr: una versión futura de ACAS X para helicópteros. Se espera que sus estándares se publiquen en 2024.

ACAS Xp: una versión futura de ACAS X que se basa solamente en vigilancia pasiva (ADS-B) para hacer el seguimiento de las aeronaves intrusas, sin necesidad de interrogar activamente a otras aeronaves en el espacio aéreo.

Su uso está destinado para aviación general, donde actualmente no es obligatorio TCAS II.

La pertinencia de esta investigación es ahora más relevante que nunca, ya que por primera vez en la historia, un sistema ACAS tendrá una variante para UAS, lo que implica una mayor exactitud en la posición tanto vertical como horizontal de las aeronaves/UAS, además ya hay sistemas de tráfico aéreo para drones (UTM: UAS Traffic Management), por lo que se espera que estos algoritmos se apliquen a drones en el futuro próximo.

Por último, la factibilidad y viabilidad se puede analizar también a través de la utilidad de los resultados obtenidos, en este caso es clave para mejorar la seguridad de las operaciones, ya que el mejorar la exactitud de la localización del resto de aeronaves o UAS en el espacio aéreo, es decir, mejorar la conciencia situacional, esto puede ayudar a reducir el número de falsos positivos, elegir la maniobra evasiva que mejor se ajuste al encuentro.

Los objetivos de esta investigación son:

1. Desarrollar una plataforma estándar de simulación que permite analizar y medir el desempeño de diferentes algoritmos de estimación
2. Diseñar algoritmos que permitan estimar con exactitud una trayectoria no lineal.
3. Comparación de los algoritmos de ACAS X con los algoritmos desarrollados durante esta tesis.

En consecuencia, la contribución de esta investigación es una plataforma de simulación de Monte Carlo que modela los errores en las variables de entrada de los algoritmos, así como un nuevo algoritmo propuesto para la estimación de la posición de los intrusos (aeronaves o UAS).

## 2. MATERIALES Y MÉTODOS

En una investigación o tesis doctoral es esencial tener una gran cantidad de datos a analizar para corroborar los modelos desarrollados. En nuestro caso la gran cantidad de datos se generarán a través de simulaciones que utilizan el método de Monte Carlo, y estos datos servirán para analizar el desempeño de los algoritmos de seguimiento de aeronaves.

Los recursos pueden ser de cualquier tipo ya sean materiales, métodos, equipos, estancias predoctorales, o hipótesis utilizadas durante la investigación.

Para llevar a cabo esta investigación se han necesitado los siguientes recursos, materiales y equipos:

1. Programador de simulación o “Simulation Scheduler”: en toda plataforma de simulación hace falta un “scheduler” que ordene las tareas que hacen falta y las ejecute de forma secuencial, de forma que la simulación realice las mismas tareas y en el mismo orden que el proceso real que estamos simulando.
2. Método de Monte Carlo: como veremos más adelante este método se utiliza para evaluar expresiones matemáticas complejas que tienen un carácter estocástico; para ello se sustituye cualquier variable/factor con incertidumbre (entradas del sistema) por una distribución de probabilidad, y se ejecutan miles de simulaciones con el objetivo de obtener la variable de salida como otra distribución que se suele parametrizar con su valor medio y variabilidad.
3. Algoritmos ACAS X (Anexo 1): el volumen II de los MOPS de ACAS X contienen una implementación de la lógica de ACAS Xa (versión de ACAS X para la aviación civil) en el lenguaje de programación Julia, durante esta tesis se codificará la misma lógica en Matlab para poder comparar los algoritmos en la misma plataforma de simulación.

4. Algoritmos propios (Anexo 2): Durante el desarrollo de esta tesis, se ha desarrollado un algoritmo propietario de seguimiento de aeronaves basado en el filtro IMM, en el lenguaje de programación Matlab.
5. Matlab IDE: el entorno de desarrollo integrado (IDE) Matlab ha sido fundamental para toda la implementación tanto de la algoritmia de evitación de colisiones como de la plataforma de simulación.
6. Estándares industriales: principalmente los MOPS (Minimum Operational Performance Standard) de ACAS X, y el Anexo 10 de ICAO. Estos estándares nos permitirán definir los modelos de ruido a utilizar durante la simulación de Monte Carlo.
7. Hipótesis: Para llevar a cabo una simulación de un proceso real (aproximación entre dos aeronaves) así como para modelar el ruido de las medidas es necesario formular varias hipótesis (e.g. tipo de ruido, tipo de trayectorias, o equipamiento de las aeronaves) que se exponen en cada uno de los capítulos pertinentes.
8. Bibliografía: artículos y libros sobre el estado del arte en simulación y algoritmos de estimación.
9. Estancia predoctoral en el Instituto Nacional de Informática (NII) de Japón, investigando sobre la automatización del tráfico aéreo de drones (configuración dinámica del espacio aéreo, planificación de trayectorias, algoritmos de estimación y tracking).
10. Equipo: La principal función del ordenador es manejar una gran cantidad de datos, ya que cada encuentro cuenta con muchos puntos de ruta ("waypoint"), y a su vez para cada encuentro se hacen varias ejecuciones de Monte Carlo ("Monte Carlo runs"). Dada la gran cantidad de datos a manejar el equipo necesitará 16 GB de RAM y un procesador con 24 MB de memoria cache para asegurar una velocidad adecuada de las simulaciones.

Uno de los objetivos de describir la metodología es explicar todos los pasos y recursos para garantizar la reproducibilidad de los resultados (pilar fundamental del método científico), ya que todo experimento científico debe ser reproducible y repetible.

Dado que la disciplina científica en la que trabajamos involucra la estadística y la computación, debemos definir reproducibilidad como la posibilidad de replicar exactamente los resultados a partir de los datos originales y el código informático empleado, y repetibilidad como la posibilidad de obtener resultados consistentes al replicar un estudio con un conjunto distinto de datos.

Con tal fin, es fundamental especificar:

1. Generador de números pseudoaleatorios: el algoritmo usado y la semilla se detallan en el capítulo 4.1, ya que aquí radica la variabilidad del experimento.
2. Modelos de ruido: Los modelos que se definen en el capítulo 4.2, se han desarrollado de acuerdo con los estándares de la industria, y están detallados en una de las publicaciones desarrolladas durante esta tesis [6]

### 3. ESTADO DEL ARTE

La técnica de estimación óptima, en términos de minimizar el error medio cuadrático, es el filtro de Kalman, aun así sus limitaciones (ver Anexo A) hacen que surjan nuevas técnicas para sistemas no lineales (EKF, UKF), o para ruido no gaussiano blanco (AKF), e incluso sigue siendo un tema de actualidad y hay varias investigaciones recientes sobre cómo mejorar estas técnicas de estimación como el filtro UKF fuerte (“Strong UKF”) de [7], donde se propone un factor (“fading factor”) variante en el tiempo que debilite las observaciones pasadas (modifica la matriz de covarianza del valor predicho), este factor se calcula en tiempo real según el residuo del filtro, y por tanto es un filtro adaptativo que evita la divergencia cuando hay incertidumbre en el proceso, o el filtro UKF modificado de [8] que intenta añadir ese “fading factor” solo cuando se detecta incertidumbre en el modelo de la dinámica del sistema, para así evitar perder exactitud cuando el modelo se ajusta bien al comportamiento real del sistema.

Por otra parte, otros autores entienden la trayectoria de la aeronave como un sistema estocástico híbrido, y consecuentemente utilizan algoritmia basada en el paradigma de múltiples modelos como [9], donde en el capítulo 4, sección 19, los autores Henk Blom y Yaakov Bar-Shalom describen las técnicas de filtrado de sistemas estocásticos híbridos no lineales, y en [10] que propone un filtro IMM compuesto de filtros de partículas; el inconveniente del filtro de partículas es su tiempo de ejecución, ya que al aumentar el número de partículas, mejorará la exactitud, pero empeorará la velocidad de ejecución. Finalmente otros autores recientes plantean un filtro IMM-UKF [11], y [12], pero en él los filtros UKF no contemplan diferentes dinámicas del sistema (diferentes ecuaciones diferenciales), simplemente se utilizan diferentes covarianzas para el ruido del proceso, o [13] donde se proponen un filtro EKF-IMM, y un filtro UKF-IMM sin aumentar, y donde no se comparan los

diferentes algoritmos de forma exhaustiva (sin test de hipótesis y sólo con 100 ejecuciones de Monte Carlo).

En esta tesis se propone una solución diferente dónde habrá varias dinámicas del proceso (paradigma de múltiples modelos, en concreto un filtro IMM), y se ponderará más la que más se ajuste a la dinámica real del sistema, y cada modelo implementará un filtro UKF aumentado (IMM-AUKF), habiendo en total dos modos de operación (movimiento lineal y maniobrando).

A continuación, se analiza profundamente el estado del arte en la industria de la aviación, que utiliza lógicas de estimación embarcadas en los aviones.

### **3.1 ACAS X**

Tras la primera colisión en el aire en 1956 sobre el Gran Cañón del Colorado en Estados Unidos, se empezaron a diseñar los primeros sistemas de evasión de colisiones. El objetivo principal de estos sistemas CAS (“Collision Avoidance System”) es reducir la incidencia de colisiones en el aire o MAC (“Mid Air Collision”), y de cuasi-colisiones en el aire o NMAC (“Near Mid Air Collision”). Todos los algoritmos de evitación de colisiones constan de dos módulos:

1. STM (“Surveillance Tracking Module”): este componente contiene la lógica de estimación de trayectorias, que es objeto de análisis en esta sección.
2. TRM (“Threat Resolution Module”): este componente se encarga de generar la maniobra evasiva, tanto en dirección como en intensidad, basándose en el vector de estado de la aeronave intrusa (estimado por el módulo STM).



Ilustración 1. Arquitectura lógica de un sistema de evitación de colisiones

En concreto ACAS X pretende reducir el número de falsos positivos respecto de versiones anteriores como TCAS II, para así reducir el número de alertas que ve el piloto. Con tal propósito los algoritmos de estimación y seguimiento serán claves para proporcionar la exactitud adecuada del estado de los blancos y así tomar la mejor decisión sobre si el intruso es una amenaza real o no (conciencia situacional).

Conviene definir lo que se considerará de aquí en adelante un evento MAC y un NMAC:

MAC: es un accidente en el que dos aeronaves entran en contacto mientras ambas se encuentran en el aire [14].

NMAC: De acuerdo con los MOPS de TCAS II [15] una cuasi-colisión en el aire ocurre cuando dos aeronaves se aproximan por debajo de los 100 ft verticalmente y 500 ft horizontalmente.

Los primeros sistemas de evasión de colisiones surgieron en la década de los 80 y se basan en el estándar de OACI ACAS [16], primero surgió TCAS I, que solo proporciona alertas de tráfico, conocidos como TA (“Traffic advisories”), y más tarde TCAS II, que proporciona TAs y resolución de conflictos en el plano vertical o RA (“Resolution Advisory”).

El salto de TCAS II a ACAS X ha sido muy importante, ya que ACAS X, que se desarrolló desde 2008 hasta 2018, se ha podido beneficiar de avances recientes como la programación dinámica, vigilancia pasiva, y algoritmos numéricos, que no existían en el momento de introducción de TCAS II, 1989

(TCAS II se empezó a desarrollar a mediados de la década de los 70). Entre todos estos avances se incluyen los algoritmos de estimación, ya que los algoritmos de estimación usados en TCAS II están obsoletos hoy en día (e.g. filtro  $\alpha - \beta$ ). Gracias a estos avances ACAS X mejorará la seguridad operacional a la vez que reduce las alertas de tráfico que presenta al piloto, además es compatible con los futuros conceptos operacionales de SESAR Y NextGen que pretenden reducir la separación entre aeronaves.

El primer algoritmo a analizar en esta tesis es el especificado por la FAA en el nuevo protocolo de evasión de colisiones ACAS X, que incorpora nuevas técnicas que no estaban disponibles cuando se desarrolló TCAS II, como el algoritmo de estimación UKF [17], que fue desarrollado en el año 2000, y que fue incluido en el desarrollo de ACAS X, por lo que podemos considerar los algoritmos de estimación de ACAS X como el estado del arte actual en lo que se refiere a vigilancia de aeronaves. Por tal motivo ACAS X será el algoritmo de referencia (“benchmarking”) en el capítulo 7, y se comparará con otros algoritmos de estimación basados en el paradigma de múltiples modelos.

Para analizar la lógica ACAS X debemos recurrir a los MOPS (“Minimum Operational Performance Standards”), que contiene el estándar para equipamiento específico y la información necesaria para entender las características y requisitos requeridos por el equipo [18].

En este caso constan de dos partes:

Volumen I [1]: define las características del equipamiento y requisitos del sistema, el desempeño mínimo en condiciones operacionales estándar y bajo estrés, así como los procedimientos de prueba para su validación.

Volumen II [2]: contiene la descripción del algoritmo o ADD (“Algorithm Design Description”), el algoritmo se presenta en el lenguaje de programación Julia con comentarios informativos para el lector.

Aunque el algoritmo se presenta en Julia, durante la simulación de esta tesis [capítulo 7] se ejecutará una implementación del algoritmo en Matlab, por lo que es necesario transformar el código a Matlab, para ello es importante además de entender el flujo de control del código, conocer los tipos de datos que se definen en Julia y su equivalencia con Matlab, como se explican en el módulo 1 de [19] así como la implementación de UKFs en Julia, sección 19.5 de [20].

Tal y como podemos deducir de los MOPS, ACAS X está preparado para trabajar con vigilancia activa y pasiva.

La vigilancia activa se realiza con el radar secundario (SSR), cuyo transpondedor (XPDR) puede operar en dos modos, el modo S y el modo C, el sistema ACAS X procesa ambos mensajes, pero para el desarrollo de la tesis, se implementará el modo S, ya que la cuantificación de la altitud de la aeronave es de 25 ft, mientras que en el modo C es de 100 ft.

La vigilancia pasiva se realiza a través de la vigilancia dependiente automática o ADS (“Automatic Dependent Surveillance”) con ADS-B y/o ADS-R, la lógica que procesa los mensajes ADS-B/R es opcional en ACAS X, ya que no todas las aeronaves cuentan con ADS.

### **3.1.1 Seguimiento del intruso**

Según los MOPS de ACAS X volumen 2 [2], la lógica de estimación de la aeronave intrusa se divide en el seguimiento en el plano horizontal y en el plano vertical. El procesado se presenta a través del siguiente “call graph” o estructura de llamadas, que es un gráfico de control de flujo, donde se representa la llamada entre las distintas subrutinas en un algoritmo. Cada nodo representa una rutina, cada extremidad representa la llamada a otra rutina, y cada ciclo en el gráfico representa una llamada recursiva a esa misma rutina.



Ilustración 2. Estructura de llamadas de la lógica de estimación del intruso

- Estimación horizontal:
  - Tracker en coordenadas cartesianas: Un filtro UKF (“Uncensted Kalman Filter”) se utiliza para estimar las cuatro dimensiones cartesianas, posiciones, y velocidades  $[x, y, dx, dy]$ , el principal propósito de este tracker es proveer una estimación estable del ángulo de azimuth y la velocidad de aproximación de la aeronave intrusa.
  - Tracker de la distancia oblicua: De nuevo un filtro UKF se utiliza para estimar la distancia oblicua y la velocidad de aproximación relativa. El filtro elegido es un filtro de tercer orden utilizando un modelo de la dinámica de la distancia oblicua al cuadrado (para evitar las no linealidades de que la distancia oblicua es siempre positiva, pero su primera y segunda derivada pueden ser positiva o negativa). Las variables de estado son  $[r, dr, ddr]$  (aunque el filtro estima la aceleración de la distancia oblicua, esa estimación no se envía al TRM).
- Estimación vertical:
  - Tracker vertical: Utiliza un filtro lineal de Kalman (KF) con algunas modificaciones para tener en cuenta las medidas cuantificadas (en cuantos de 25 ft). La distribución inicial de la altitud será una gaussiana centrada en la medida de la altitud, mientras que la velocidad vertical es una gaussiana centrada en cero; las varianzas iniciales para estas distribuciones se toman del fichero de parámetros de ACAS X ya que no se pueden determinar a partir de una sola medida, estos valores son independientes de la cuantización, y son seleccionados.

### 3.1.1.1 Tracker cartesiano

Las rutinas del tracker cartesiano son:

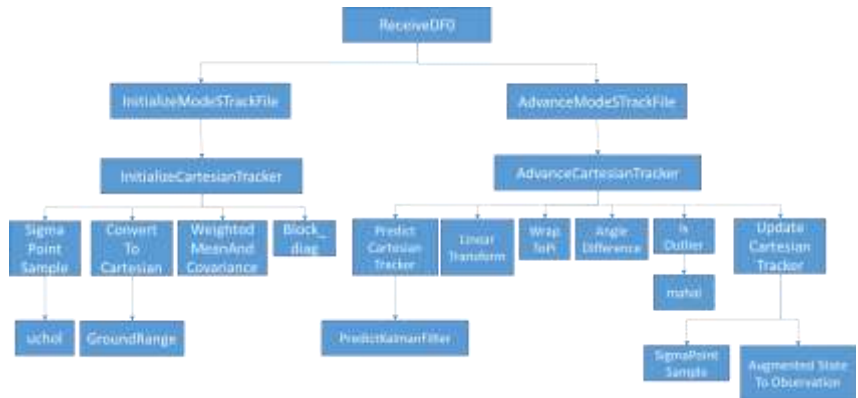


Ilustración 3. Estructura de llamadas de la lógica del tracker cartesiano

- Algoritmo 16, InitializeCartesianTracker:** Este algoritmo inicializa el tracker en coordenadas cartesianas, toma como entrada la pista (“track”) denominada “ModeTrackFile”, que contiene la distancia oblicua medida, denominada “r\_slant” (“slant range”), el rumbo absoluto, denominado “Chi\_abs”, y la altitud relativa, denominada en el código “z\_rel”.

Primero valida las entradas, si el rumbo absoluto del intruso es válido, entonces se comprueba que la distancia oblicua es por lo menos tan grande como la altitud relativa para asegurarnos que ciertas matrices permanezcan como semi-definidas positivas, ya que las matrices de covarianza tienen tendencia a perder su propiedad de semi-definida positiva cuando se propagan, como se explica en la sección 7.1.1 de [21].

La distribución de la posición inicial se obtiene a través de la transformación no lineal de la observación de sus coordenadas nativas (polares) a cartesianas de la siguiente forma:

- La distribución de la observación (caracterizada por los valores observados: distancia oblicua, rumbo absoluto y altitud relativa, y su

matriz de ruido) se muestrea usando los puntos sigma con el algoritmo 17: SigmaPointsSample.

- Para cada muestra se vuelve a comprobar que la distancia oblicua es mayor que la altitud relativa para validar la muestra, y en caso afirmativo cada muestra se convierte a coordenadas cartesianas con el algoritmo 18: ConvertToCartesian.
- Finalmente se calcula la media y covarianza ponderada en el eje de referencia cartesiano con el algoritmo 19: WeightedMeanAndCovariance.
- Algoritmo 17, “SigmaPointSample”: Selecciona los puntos sigma del filtro UKF, con las ecuaciones vistas en el capítulo 3 (Filtro de Kalman Unscented)
- Algoritmo 18, “ConvertToCartesian”: transforma las medidas de la aeronave intrusa en coordenadas polares (distancia oblicua y ángulo de azimut) de cada uno de los puntos sigma a coordenadas cartesianas (x e y).

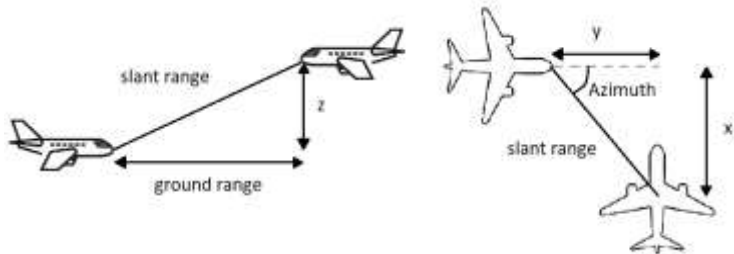


Ilustración 4. Vista en perfil y planta de la aproximación entre dos aeronaves

$$x = r \cdot \sin(\theta)$$

$$y = r \cdot \cos(\theta)$$

Donde “r” es la proyección horizontal de la distancia oblicua, y “ $\theta$ ” es el ángulo de azimut.

- Algoritmo 19, “WeightedMeanAndCovariance”: Combina los puntos sigma y sus pesos asociados para determinar la media ponderada y la matriz de covarianza.
- Algoritmo 20, “AdvanceCartesianTrack”: Este algoritmo primero comprueba la validez de la observación y luego lleva a cabo la detección de valores atípicos

para decidir cómo se actualizará el tracker, si encuentra alguna discrepancia (valor atípico, incongruencia), la pista se reinicializa, en caso contrario se predice la pista, y luego se corrige o actualiza.

- Algoritmo 21, “LinearTransform”: lleva a cabo la transformación de las variables de estado cartesianas a polares, la transformación de la media la hace a través de la ecuación no lineal, mientras que la transformación de la matriz de covarianza la hace a través de la matriz Jacobiana, ya que necesita una ecuación lineal.

$$r = \sqrt{x^2 + y^2}$$

$$\theta = \text{atan}\left(\frac{y}{x}\right)$$

- Algoritmo 22, “isOutlier”: comprueba si la observación es un valor atípico, comparando la estimación y la observación (en coordenadas polares) frente a la distancia de Mahalanobis.
- Algoritmo 23, “PredictCartesianTracker”: predice la pista hasta el tiempo de observación según un modelo lineal de velocidad constante.

$$\begin{pmatrix} x \\ y \\ v_x \\ v_y \end{pmatrix} = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} x \\ y \\ v_x \\ v_y \end{pmatrix} + \begin{bmatrix} 0.5dt^2 & 0 \\ 0 & 0.5dt^2 \\ dt & 0 \\ 0 & dt \end{bmatrix} \cdot \begin{pmatrix} a_x \\ a_y \end{pmatrix} + Q$$

- Algoritmo 24, “UpdateCartesianTracker”: determina la ganancia de Kalman y calcula la distribución estimada.
- Algoritmo 25, “AugmentedStateToObservation”: transforma los puntos sigma a través de la ecuación de medida.
- Algoritmo 27, “GroundRange”: computa la proyección horizontal de la distancia oblicua, es decir, la distancia horizontal, a partir de la distancia oblicua ( $r_{\text{slant}}$ ), y de la altitud relativa ( $z_{\text{rel}}$ ).

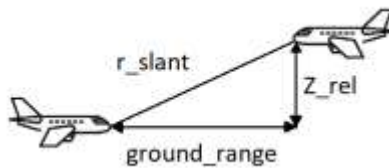


Ilustración 5. Proyección horizontal de la distancia oblicua (ground range)

- Algoritmo 64, “PredictKalmanFilter”: determina la distribución predicha.
- Algoritmo 340, “mahal”: determina la distancia de Mahalanobis para la detección de valores atípicos.
- Algoritmo 342, “Block\_diag”: combina dos matrices en una única matriz diagonal.
- Algoritmo 341, “uchol”: este algoritmo lleva a cabo la descomposición triangular superior de Cholesky, con el fin de determinar la raíz cuadrada de una matriz simétrica y semi-definida positiva.
- Algoritmo 343, “AngleDifference”: devuelve la diferencia angular.
- Algoritmo 346, “WrapToPi”: mapea el ángulo de entrada en radianes al intervalo entre  $[-\pi, \pi]$ .

### 3.1.1.2 Tracker de la distancia

Las rutinas del algoritmo de seguimiento de la distancia son:

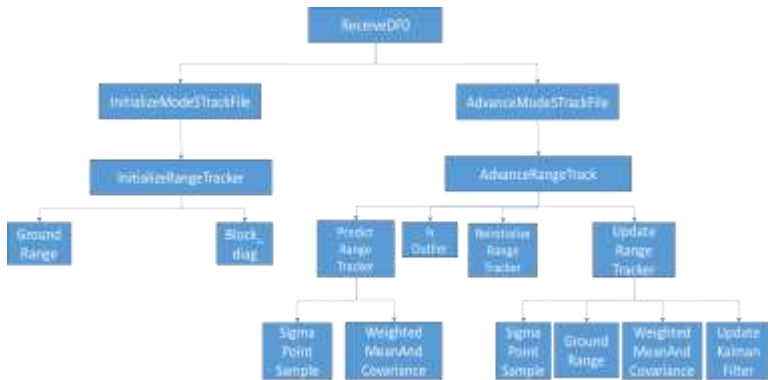


Ilustración 6. Estructura de llamada del tracker de distancia

- Algoritmo 26, “InitializeRangeTracker”: inicializa el tracker y valida las entradas antes de crear una pista. El tracker se inicializa con una distribución gaussiana de media la “ground range” inicial y con varianza equivalente a la de la ecuación de observación, y la velocidad y aceleración se inicializan con una gaussiana centrada en cero.

$$\begin{pmatrix} range \\ speed \\ acceleration \end{pmatrix} = \begin{pmatrix} range_0 [ft] \\ 0 [ft/s] \\ 0 [ft/s^2] \end{pmatrix}$$

$$Q = \begin{bmatrix} 2500 [ft^2] & 0 & 0 \\ 0 & 160000 [ft^2/s^2] & 0 \\ 0 & 0 & 512 [ft^2/s^4] \end{bmatrix}$$

- Algoritmo 27, “GroundRange”: determina la proyección horizontal de la distancia oblicua.
- Algoritmo 28, “AdvanceRangeTrack”: valida la observación, realiza la detección de valores atípicos, y reinicializa el tracker en caso de cualquier tipo de problema (e.g. entrada errónea, o valor atípico), en caso contrario predice y luego corrige la estimación.

- Algoritmo 29, “ReinitializeRangeTracker”: la pista se reinicializa en caso de fallo.
- Algoritmo 30, “PredictRangeTracker”: utiliza un filtro UKF, y además utiliza un modelo de movimiento de la distancia cuadrática, ya que la ecuación tiene una no linealidad, que ocurre cuando una aeronave está tan cerca que pasa de acercarse (velocidad relativa negativa) a alejarse (velocidad relativa positiva).

Por ejemplo para dos aeronaves que se aproximan a velocidad constante, la distancia entre las dos aeronaves evoluciona según la siguiente gráfica, donde podemos ver una singularidad en el mínimo de la función (CPA: Closest Point of Approach), sin embargo al elevar la distancia al cuadrado la singularidad desaparece.

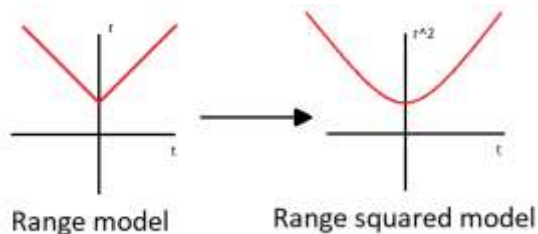


Ilustración 7. modelo de la distancia horizontal al cuadrado

- Algoritmo 31, “UpdateRangeTracker”: Corrige la estimación según las reglas ya vistas del filtro UKF, siendo la ecuación de medida:

$$(r) = (1 \quad 0 \quad 0) \begin{pmatrix} r \\ \dot{r} \\ \ddot{r} \end{pmatrix} + R_k$$

- Algoritmo 32, “UpdateKalmanFilter”: determina la ganancia del filtro de Kalman.

### 3.1.1.3 Tracker vertical

Las rutinas del tracker vertical son:

- Algoritmo 33, InitializeVerticalTracker: Inicializa el tracker, la altitud como una gaussiana centrada en la altitud inicial, y la velocidad vertical como una gaussiana centrada en cero.
- Algoritmo 34, AdvanceVerticalTrack: decide cómo evoluciona la pista, si como un valor atípico, o “coasted”, o progresa todo como una pista normal y por tanto tendrá su paso de predicción y de corrección.
- Algoritmo 35, “PredictVerticalTracker”: utiliza el paso de predicción de un filtro de Kalman estándar con un modelo de velocidad constante.

$$\begin{pmatrix} z \\ \dot{z} \end{pmatrix}_{t+1} = \begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix} \begin{pmatrix} z \\ \dot{z} \end{pmatrix}_t + \begin{bmatrix} 0.5dt^2 \\ dt \end{bmatrix} (\ddot{z})$$

- Algoritmo 36, “UpdateVerticalTracker”: realiza el paso de corrección de un filtro de Kalman estándar, es decir, calcula la ganancia de Kalman; pero no se aplica directamente sobre la altitud observada, ya que está cuantificada, y utiliza la salida de “PredictedAltitude”.
- Algoritmo 37, “PredictedAltitude”: determina la altitud observada para tener en cuenta el efecto de la cuantificación.

Para resumir, ACAS X hace el seguimiento de la aeronave intrusa con tres lógicas distintas de seguimiento, dos para el movimiento horizontal y un tercero para el vertical, ya que ambos movimientos están desacoplados:

1. Tracker cartesiano: estima las coordenadas x e y de la aeronave intrusa relativas a la aeronave propia, mediante un filtro UKF (ya que la ecuación de medida no es lineal) aumentado y de segundo orden; y un modelo de movimiento a velocidad constante.
2. Tracker de distancia: utiliza un filtro UKF de tercer orden, aumentado y con un modelo de movimiento de la distancia cuadrática.
3. Tracker vertical: se utiliza un filtro de Kalman lineal con algunas modificaciones para tener en cuenta las medidas cuantificadas.

Algunos comentarios finales, son:

1. Las lógicas de estimación horizontal y vertical están separadas, ya que ambas dinámicas son independientes y esto permite mejorar el rendimiento del filtro.
2. Para la estimación horizontal se utilizan filtros UKF, lo que tiene sentido dadas las propiedades no lineales que hemos comentado.
3. Para la estimación horizontal se utilizan filtros aumentados, lo cual es necesario, ya que el ruido de aeronaves que maniobran no suele ser blanco y se puede modelar como ruido auto-correlado.
4. ACAS X cuenta con más algoritmos de estimación, para estimar las variables de estado de la aeronave propia, como el rumbo o la altitud, (no filtra la radio-altitud, ni las coordenadas WGS84).

## 4. SIMULADOR DE MONTE CARLO

Es un simulador que aplica el método de Monte Carlo, explicado en el Anexo A de esta tesis; que es un método no determinista, que se basa en el muestreo aleatorio de distribuciones dadas para analizar un proceso estocástico, cuyas expresiones matemáticas no se pueden tratar analíticamente [22]

Al aplicar el método de Monte Carlo muchas veces, obtendremos muchas ejecuciones de Monte Carlo (“Monte Carlo runs”), cada una de ellas es una simulación del encuentro entre aeronaves con distintas realizaciones de los distintos ruidos de medida. En consecuencia, como veremos más adelante aparecerán en los resultados distribuciones gaussianas, esto se debe al teorema del límite central.

N.B. En estadística una realización es una observación de una variable aleatoria, es el valor que realmente tomó la variable; la variable aleatoria no es el valor en sí mismo, sino que es el proceso que gobierna las observaciones.

### **Teorema del límite central**

Para diseñar una buena simulación de Monte Carlo conviene conocer el teorema del límite central; que es un teorema fundamental de probabilidad y estadística. El teorema establece que siendo  $S$  la suma de  $n$  variables aleatorias independientes, con media conocida y varianza no nula pero finita, entonces la función de distribución de  $S$  se puede aproximar a una distribución normal si la suma de estas variables independientes es lo suficientemente grande.

Esto implica que dada una población de una variable (por ejemplo como veremos más adelante, el error de estimación) cuya distribución es desconocida, pero conocemos su media ( $\mu$ ) y su varianza ( $\sigma^2$ ), si se toman muestras de la población de tamaño  $n$ , y este proceso de muestreo se repite  $k$

veces, obtendríamos  $k$  muestras de tamaño  $n$ , si se determina la media de esas muestras (i.e. la suma de  $n$  variables aleatorias), las medias muestrales seguirán una distribución normal si  $n$  es lo suficientemente grande:

$$\bar{x} \sim N\left(\mu, \frac{\sigma^2}{n}\right)$$

En las siguientes subsecciones se definirán el generador de números aleatorios y los modelos de ruido a utilizar durante las simulaciones.

## 4.1 Generación de números pseudo-aleatorios

Existen principalmente dos estrategias para generar números aleatorios:

- Generador de números aleatorios: se basan en hardware, por lo que generalmente son dispositivos mecánicos que resultan lentos. El método más empleado es la utilización del ruido (midiendo el ruido eléctrico en voltaje).
- Generador de números pseudo-aleatorios (PRNG: Pseudo-Random Number Generators): se basan en un algoritmo que contiene operaciones aritméticas, por lo que son más rápidos, pero no son números aleatorios, aunque cumplen el objetivo de que generan una secuencia de números que no se pueden predecir mejor que haciéndolo al azar. Una de sus ventajas radica en que se pueden reproducir, por lo que se pueden elaborar test estadísticos, e.g. test de rachas, para comprobar la calidad de la secuencia de los números obtenidos.

Para las simulaciones de Monte Carlo de esta tesis, se utilizarán generadores de números pseudo-aleatorios, cuya semilla será el tiempo que marca el ordenador.

Algunas de las propiedades que buscamos del algoritmo son:

- No predecible

- Reproducible: si usásemos la misma semilla
  - Periodo alto: el periodo o ciclo no repetitivo es el tiempo que tarda en repetirse la secuencia de números aleatorios
  - Uso de poca memoria
- Algunos de los algoritmos más populares son:
- Método de los cuadrados medios: su funcionamiento se basa en tomar un número de  $2n$  cifras (semilla), elevarlo al cuadrado de forma que el número tendrá  $4n$  cifras, y tomar las  $2n$  cifras centrales poniendo un punto decimal delante, de esta forma generaremos números aleatorios uniformemente distribuidos entre 0 y 1. Aunque presenta algunos inconvenientes, como que tiene bajo periodo y fuerte dependencia con la semilla.

$$3708 \xrightarrow{\quad \wedge 2 \quad} 13\ 7492\ 64 \xrightarrow{\quad} 0.7492$$

Ilustración 8. Ejemplo método de los cuadrados medios.

- Generadores congruenciales: siguen la siguiente fórmula recurrente:

$$X_i = (aX_{i-1} + c) \bmod(m)$$

Donde “a” es el multiplicador, “c” el incremento, “m” el módulo y “ $X_0$ ” la semilla.

- Generadores multiplicativos: es un subtipo de los congruenciales, pero en los que se elimina la operación de suma (el incremento es nulo), por lo que mejora la eficiencia.
- Mersenne Twister: denominado así, porque su periodo es un número primo de Mersenne; el más popular es MT19937, que tiene un periodo de  $2^{19937}-1$ .

### Periodo del generador de números pseudo-aleatorios

En la implementación del simulador en Matlab utilizaremos un generador basado en el algoritmo de Mersenne twister, en concreto “*mt19937ar*”, que tiene un periodo de  $2^{19937}-1$ , descrito en [23].

El periodo del generador es muy importante, especialmente en aplicaciones que utilicen el método de Monte-Carlo, ya que la validez del experimento dependerá en que cada ejecución de Monte Carlo sea independiente, y para ello el periodo del generador debe ser mayor a todos los números aleatorios que vamos a generar durante el experimento. Durante este experimento se:

- Modela el ruido de 4 variables: Altitud, rumbo, azimut, y distancia
- Cada modelo de ruido consta de 2 tipos de errores: sistemático (“bias”) y aleatorio (“jitter”)
- El ruido se genera para cada ciclo ACAS, que coincidirá con cada punto de ruta (“waypoint”); cada encuentro puede tener unos 300 puntos de ruta.
- Cada encuentro consta de dos aeronaves.
- Cada encuentro se simulará unas 10000 veces (ejecuciones de Monte-Carlo)
- Se simularán 10 encuentros

Por tanto, en total se generarán aproximadamente 480,000,000 números aleatorios, que es menor que el periodo del generador de Mersenne Twister, por lo que este algoritmo es apto para esta aplicación.

Otros algoritmos disponibles en Matlab 2022 [24] son:

<b>name</b>	<b>foundation</b>	<b>period</b>
mt19937ar	Mersenne Twister	$2^{19937-1}$
dsfmt19937	Mersenne Twister orientado a SIMD	$2^{19937-1}$
mcg16807	Generador congruencial multiplicativo	$2^{31} - 2$
mlfg6331_64	Generador de Fibonacci retardado multiplicativo	$2^{124}$
mrg32k3a	Generador combindo múltiple recurso	$2^{191}$
philox4x32_10	Generador philox	$2^{193}$
threefry4x64_20	Generador Threefry	$2^{514}$
shr3cong	Generador de registro de desplazamientos más generador lineal congruencial	$2^{64}$
swb2712	Generador modificado de resta con préstamo	$2^{1492}$

Finalmente cabe mencionar que los generadores de números pseudo-aleatorios anteriores generan números aleatorios de acuerdo a una distribución uniforme, por lo que hace falta aplicar técnicas matemáticas para generarlos de acuerdo a una distribución dada, generalmente una distribución normal. Como por ejemplo el método del rechazo de Von Neumann, explicado en la sección 7.3 de [25]. Este método es una técnica potente para generar números aleatorios de acuerdo a una distribución conocida que sea computable; y se basa en la idea geométrica de generar un punto bidimensional aleatorio, si el punto cae dentro de la distribución se acepta, y si cae fuera se rechaza.

Para generar el “punto bidimensional aleatorio”, primero se obtiene un número aleatorio distribuido uniformemente en el rango de la función (“p”) de probabilidad (de  $x_{\min}$  a  $x_{\max}$ ), denominado  $x_{\text{random}}$ , después se computa un segundo número aleatorio uniformemente distribuido entre 0 y el máximo de la función ( $f_{\max}$ ), denominado  $u_{\text{random}}$ . Si  $u_{\text{random}} < p(x_{\text{random}})$ , se acepta el número aleatorio, en caso contrario se rechaza y se vuelve a repetir.

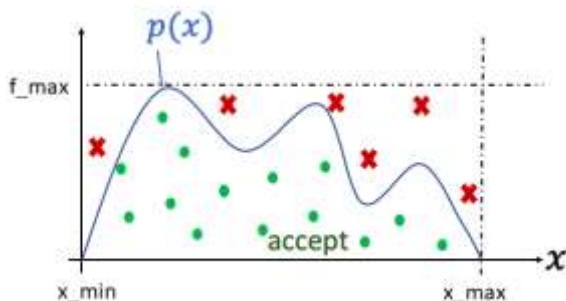


Ilustración 9. zona de aceptación y rechazo

## 4.2 Modelos de ruido

Primero necesitamos definir las variables en las cuales vamos a modelar el ruido, que serán todas las entradas de la lógica de estimación de trayectoria que sean medidas por la aeronave:

- Altitud
- Rumbo
- Distancia oblicua
- Azimut

Hay muchas otras variables que se considerarán deterministas o nulas como por ejemplo:

- Se asume velocidad del viento determinista igual a cero.
- Se asume un tiempo de procesamiento del sistema ACAS determinista igual a cero
- Se asume que todos los sistemas que proveen información al TCAS están sincronizados y proveen la información al inicio del ciclo de procesamiento TCAS (offset = 0)

Finalmente, antes de modelar el error en nuestras variables medidas, necesitamos conocer la incertidumbre, principalmente nos centraremos en la incertidumbre en la medición, es decir, el error en las variables observadas y/o medidas, y sus tipos:

- Error aleatorio o “Jitter”: es un error que no se puede controlar, es fruto del azar, generalmente de media cero.

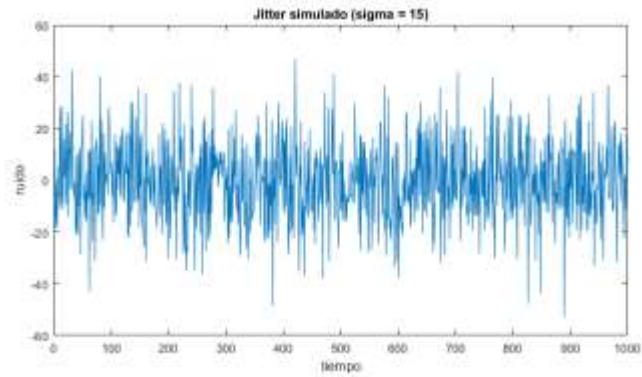


Ilustración 10. Jitter simulado.

- Error de sistemático, sesgo o “Bias”: es un error constante para un mismo equipo de medida (típicamente se da en equipos mal calibrados, o cuando hay unas condiciones ambientales que afectan a la medidas en una cierta manera), un ejemplo es el error ionosférico, debido al retardo de las ondas electromagnéticas al cruzar la ionosfera, causando un error en las medidas conocido y constante.

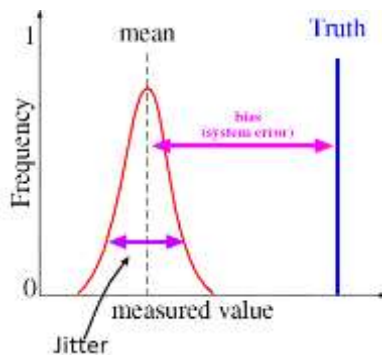


Ilustración 11. Error sistemático.

- Valor atípico o “Outliers”: también conocidos en la literatura como cisnes negros. Es un valor medido o una observación que difiere significativamente del resto de medidas. Una forma de determinar si una medida es un valor atípico consiste en usar la distancia de Mahalanobis, que es una herramienta para medir la distancia normalizada entre dos distribuciones, o entre dos puntos, o entre una distribución y un punto en un espacio multivariable, si medimos la distancia entre dos puntos en un espacio bivariable (bidimensional), o trivariable (tridimensional) donde ambas variables no están correlacionadas (i.e. los ejes forman ángulos de 90°), entonces la distancia de Mahalanobis y la distancia euclídea coinciden.

La distancia de Mahalanobis se puede expresar mediante varias fórmulas según lo que se quiera medir (e.g. distancia entre dos variables aleatorias), pero en nuestro caso lo que queremos medir es la distancia entre una observación  $\vec{x} = (x_1, x_2, \dots, x_n)$ , nótese que al estar en un espacio multivariable, la observación  $\vec{x}$  es un vector de varias variables (de estado o no), respecto de un conjunto de medidas  $\vec{\mu} = (\vec{\mu}_1, \vec{\mu}_2, \dots, \vec{\mu}_n)$  con matriz de covarianza S. Según [26] la distancia de Mahalanobis se puede expresar como:

$$d(x, y) = \sqrt{(\vec{x} - \vec{\mu})^T \cdot S^{-1} \cdot (\vec{x} - \vec{\mu})}$$

La gran diferencia con la distancia euclídea se basa en normalizar la distancia con la varianza de las variables, de esta forma todas las variables del espacio multivariable, tienen la misma importancia independientemente del orden de magnitud de cada una.

#### 4.2.1 Modelo auto regresivo de primer orden

Una hipótesis a menudo utilizada es la de ruido gaussiano blanco; sin embargo, en la realidad el ruido es coloreado, es decir, está auto-correlado en el tiempo, es por esta razón por la que los filtros utilizados en ACAS X están aumentados (ver Capítulo 3: ACAS X), ya que se sabe que las hipótesis del filtro de Kalman (ruido gaussiano blanco y no auto-correlado) no se cumplen.

Por tanto la mayoría de modelos de ruido que vamos a utilizar, implementarán un proceso auto regresivo de primer orden, es decir un proceso de Gauss-Markov en el que la  $u_k$  es ruido blanco, y el ruido final ( $x_k$ ) depende solo del ruido de la época anterior ( $x_{k-1}$ ).

$$x_k = \alpha x_{k-1} + u_k \quad k = 0, 1, 2, \dots$$

Donde  $u_k$  es ruido blanco con desviación típica  $\sigma_u$  y donde el factor de auto-correlación  $|\alpha| < 1$ . El ruido ( $x_k$ ) se caracteriza por tener la siguiente media, varianza y auto-correlación de acuerdo con [27]:

$$\mu_x = E(x_k) = 0$$

$$\sigma_x^2 = \text{var}(x_k) = \frac{\sigma_u^2}{1 - \alpha^2}$$

$$\rho_x(m) = \text{cor}(x_{k+m}, x_k) = \alpha^m \quad (m \geq k)$$

Usando un tiempo de muestreo  $\tau_s$ , entonces la función de auto-correlación para un tiempo  $\tau = m\tau_s$  es:

$$\rho(\tau) = \alpha^{\tau/\tau_s}$$

## 4.2.2 Altitud

La altitud medida por una aeronave sobre sí misma o sobre una aeronave intrusa es la altitud de presión, que es la altitud que corresponde a una presión concreta en condiciones de atmósfera estándar, por tanto, dependiendo de la calibración del altímetro (la presión de referencia) la altitud de presión será el nivel de vuelo, la altitud o la altura. Por tanto, el altímetro barométrico del avión mide la presión atmosférica, y según el

modelo de la presión estándar de la atmósfera (la presión atmosférica desciende con la altitud aproximadamente 1 hPa por cada 27 pies) determina la altitud de presión, más tarde la aeronave intrusa envía la altitud de presión cuantificada a través del transpondedor del avión (en el modo C se manda con una resolución de 100 ft y en Modo S de 25 ft).

El altímetro del avión puede presentar tres calibraciones distintas:

- QNE: el altímetro se calibra con la presión al nivel del mar según el modelo de la atmósfera estándar internacional (ISA), es decir, 1013.25 hPa o 29.92 inHg. En este caso la lectura del altímetro nos daría el nivel de vuelo (FL).
- QNH: el altímetro se calibra con la presión al nivel del mar local, y por tanto la lectura del altímetro da la altitud.
- QFE: el altímetro se calibre con la presión de referencia del aeródromo, por tanto, la lectura del altímetro nos da la altura respecto del aeródromo.

Una vez entendido como funciona el altímetro barométrico de un avión; a continuación, analizaremos los estándares para modelar la estimación de la altitud de presión; para finalmente presentar el modelo que utilizaremos durante las simulaciones.

## *Estándares*

### **ACAS MOPS**

La altitud de presión o barométrica es una entrada importante del TCAS II y ACAS X, y por tanto la operación de TCAS II y ACAS X no se ve afectada por la calibración del altímetro: QNE (nivel de vuelo), QNH (altitud) o QFE (altura), ya que los equipos TCAS/ACAS siempre utilizan como entrada la altitud de presión con la calibración de presión estándar (QNE), por tanto la selección de la presión por parte de la tripulación de vuelo no afecta a los sistemas TCAS [28].

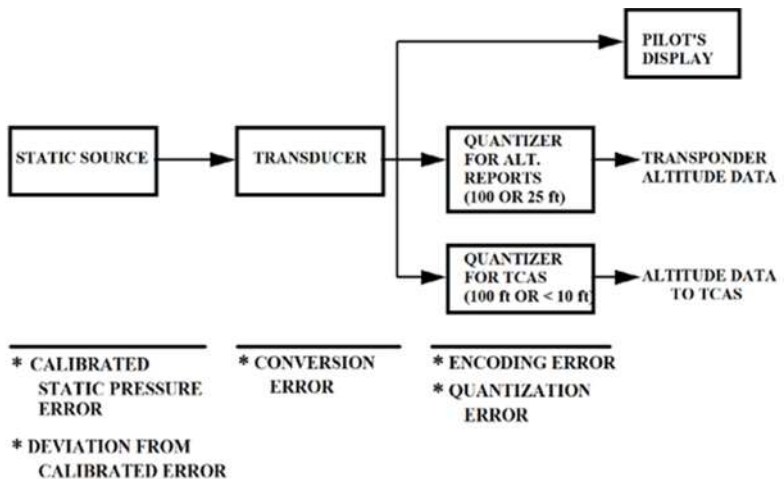


Ilustración 12. Sistema altímetro barométrico y componentes de error [1]

Los requisitos operacionales del Sistema de altimetría son iguales para TCAS II (sección 3.2.8 de [15]) y para ACAS X ([1], Sección 3.2.8). Los requisitos tienen en cuenta fuentes de error estáticas y variables. Las fuentes de error estáticas surgen de la variación en la posición y condiciones físicas de la sonda de presión estática, así como los errores asociados con la transmisión de la presión del aire al transductor.

Las fuentes de error variables incluyen errores derivados del transductor (errores de histéresis, de temperatura, de fricción, y de redondeo entre otros) y del cuantificador (errores de codificación y de redondeo).

En los MOPS de TCAS II y ACAS X [15] y [1] se especifican los límites de error total del sistema de altimetría, y estos límites se deben cumplir un 99.7% del tiempo.

Los errores máximos se muestran en la siguiente tabla ([15] y [3] Sección 3.2.8.2), se asume que el error total del sistema de altimetría sigue una distribución gaussiana, por tanto los errores de la siguiente tabla son los

valores  $3\sigma$ , donde  $\sigma$  es la desviación típica de la distribución gaussiana que describe el error total del sistema de altimetría. Estos errores máximos no distinguen entre errores estáticos y variables.

**Tabla 1. Pressure altimetry performance requirements (TCAS II MOPS [18] Table 3-1, draft ACAS X MOPS [17] Section 3.2.8).**

Altitud [ft]	99.7% error límite [ft]
Mean sea level	135
5,000	144
10,000	156
15,000	174
20,000	195
25,000	213
30,000	234
35,000	258
40,000	285

Respecto a los requisitos de actualización de la altitud, se especifican en el párrafo 2.2.5.5.1 de los MOPS de ACAS X [1].

- El módulo STM deberá (1503) recibir la altitud barométrica de la propia aeronave con una frecuencia mínima de 1 Hz.
- El error en el tiempo de vida de la altitud barométrica de la propia aeronave deberá ser menor de 150 ms

### ICAO Annex 10

En la sección 4.4.2.4 del Anexo 10 de ICAO [29], se describe un modelo para el error del sistema de altimetría para la simulación de ACAS II.

Las condiciones o hipótesis para el correcto funcionamiento de este modelo son: (sección 4.4.2.1, [29])

- Los informes de altitud de la aeronave intrusa, que son las respuestas del Modo C, se expresan en cuantos de 100 ft.
- La altitud de la aeronave propia no está cuantizada, y tiene una precisión de al menos 1 ft.
- Los errores en las medidas de altitud para las dos aeronaves son constantes para todo el encuentro.
- Los errores en las medidas de altitud de ambas aeronaves siguen el siguiente modelo estándar de error de altimetría.

Los errores en las medidas de altitud siguen una distribución Laplaciana, de media cero, y la siguiente función de densidad de probabilidad (sección 4.4.2.4, [29]):

$$p(u) = \frac{1}{2\lambda} \cdot \exp\left(-\frac{|u|}{\lambda}\right)$$

La desviación típica de esta distribución de probabilidad es  $\sigma = \lambda\sqrt{2}$ . Los límites del error durante 99.7% del tiempo se consiguen para  $|u| \approx 5\lambda$ , durante el 95% del tiempo para  $|u| \approx 2.6\lambda$ . Y por último el parámetro  $\lambda$  depende de la altitud de vuelo y del equipamiento ACAS:

- Para una aeronave equipada con ACAS,  $\lambda = \lambda_1$ , definiéndose  $\lambda_1$  en la siguiente tabla

**Tabla 2. parámetro Lambda del modelo estándar del error del sistema de altimetría (Sección 4.4.2.4.2 de [29]).**

Capa de altitud	1	2	3	4	5	6
$\lambda_1$	10 m 35 ft	11 m 38 ft	13 m 43 ft	17 m 58 ft	22 m 72 ft	28 m 94 ft
$\lambda_2$	18 m 60 ft	18 m 60ft	21 m 69 ft	26 m 87 ft	30 m 101 ft	30 m 101 ft

La capa de altitud de un encuentro se determina con la altitud media de las dos aeronaves en el CPA (“Closest Point of Approach”).

**Tabla 3. Capas de altitud para la evaluación de ACAS II (Sección 4.4.1 de [29]).**

Capa de altitud	1	2	3	4	5	6
Desde	-	2300 ft 701 m	5000 ft 1524 m	10000 ft 3048 m	20000 ft 6096 m	41000 ft 12497 m
Hasta	2300 ft 701 m	5000 ft 1524 m	10000 ft 3048 m	20000 ft 6096 m	41000 ft 12497 m	

- Para una aeronave no equipada con ACAS, entonces  $\lambda$  puede tomar cualquier valor  $\lambda_1$  o  $\lambda_2$ , según una probabilidad dependiente de la altitud, como se define en la siguiente tabla:

**Tabla 4. Probabilidades para el parámetro lambda según el modelo estándar de error del Sistema de altimetría para una aeronave no equipada con ACAS (Sección 4.4.2.4.4 de [29]).**

Layer	1	2	3	4	5	6
$P(\lambda_1)$	0.391	0.320	0.345	0.610	0.610	0.610
$P(\lambda_2)$	0.609	0.680	0.655	0.390	0.390	0.390

A continuación, se presenta una comparación del error máximo esperado durante el 99.7% del tiempo y su desviación típica entre el modelo de los MOPS de TCAS II y ACAS X frente al modelo estándar del sistema de altimetría del Anexo 10 de ICAO. Como se puede observar los errores del modelo del Anexo 10 de ICAO son considerablemente mayores que los de los requisitos de los MOPS.

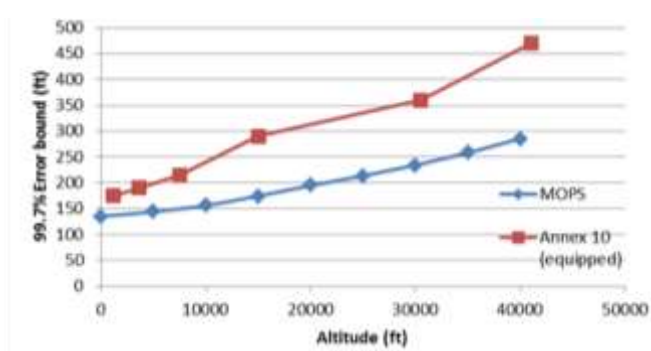


Ilustración 13. Error máximo esperado durante el 99.7% del tiempo en función de la altitud: ACAS MOPS vs Anexo 10 ICAO

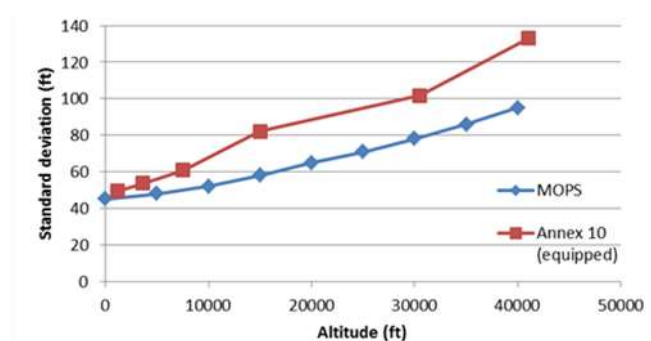


Ilustración 14. Desviación típica del error en función de la altitud: ACAS X MOPS vs Anexo 10 ICAO

### **Titulo 14 del código de reglamentos federales de US**

En la parte 43 del título 14 (“Aeronáutica y espacio”) del código de reglamentos federales de Estados Unidos (eCFR) [30] se especifica las regulaciones para el mantenimiento, mantenimiento preventivo, fabricación y alteración de aeronaves. En concreto en el apéndice E de la parte 43 se especifican los requisitos para las tolerancias de los errores de los sistemas de altimetría de presión como se muestran en la siguiente tabla.

**Table 5. Tolerancias del Sistema de altimetría según el apéndice E de la parte 43 de eCFR Titulo 14.**

<b>Altitud (ft)</b>	<b>Tolerancia (ft)</b>
-1,000	20
0	20
500	20
1,000	20
1,500	25
2,000	30
3,000	30
4,000	35
6,000	40
8,000	60
10,000	80
12,000	90

14,000	100
16,000	110
18,000	120
20,000	130
22,000	140
25,000	155
30,000	180
35,000	205
40,000	230
45,000	255
50,000	280

Si interpretamos estas tolerancias como los valores 2 sigma de una distribución normal, lo podremos comparar con las desviaciones típicas especificadas en los MOPS y en el Anexo 10 de ICAO, como se compara en la siguiente figura, dónde se puede apreciar que la exactitud requerida en la parte 43 es más variable con la altitud (i.e. mayor pendiente) y menor para altitudes de hasta 20000 ft.

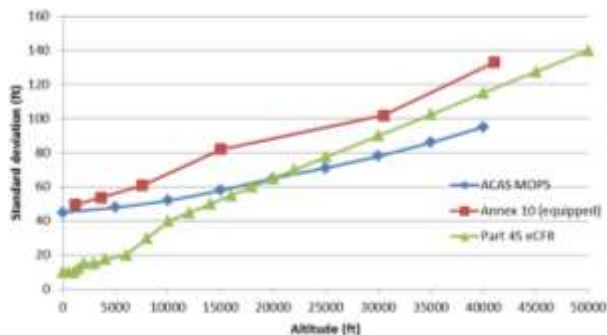


Ilustración 15. Comparación de desviaciones típicas de ACAS X MOPS, Anexo 10 ICAO y eCFR Parte 45.

### ASARP model

En el Proyecto ASARP (ACAS Safety Analysis post RVSM) se usaron medidas reales para llegar a un modelo del error del sistema de altimetría para operaciones RVSM (Reduced Vertical Separation Minima) [31]. En este modelo el error tiene una media de 0.6 ft y una desviación típica de 65 ft, con más del 99% de la distribución en el rango de  $\pm 269$ ft. Y la proporción de la distribución con un error de altimetría mayor de 150 ft es del 2.25%.

Al contrario que en otros modelos especificados en estándares como [29], [3] y [15] no hay incremento del error con la altitud a partir de FL100, como se muestra en la siguiente figura.

Una función empírica de densidad de probabilidad del error de altimetría de una aeronave con operaciones RVSM se muestra en [32]. El modelo de error de ASARP fue utilizado en los estudios de simulación de ACAS para SESAR [33] y [34].

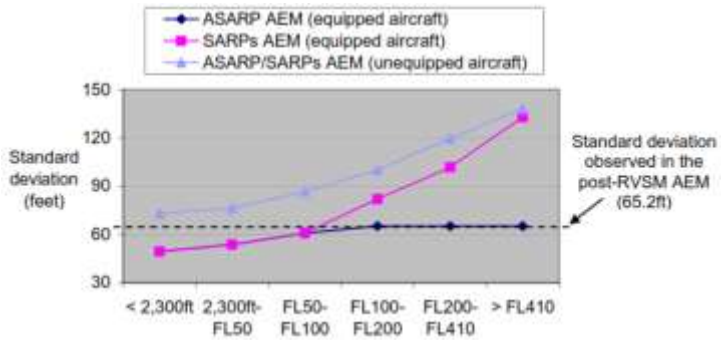


Ilustración 16. Desviaciones típicas según varios modelos de errores de altimetría: ASARP AEM (equipado), SARP AEM (equipado) según Anexo 10 ICAO, ASARP/SARPs AEM (sin equipar) según Anexo 10 de ICAO.

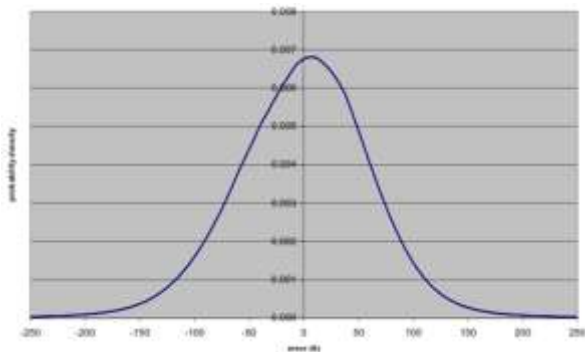


Ilustración 17. Función de densidad de probabilidad del error de altimetría de una aeronave en RVSM.

### *Modelo de ruido para la altitud de presión*

El modelo de altitud de presión que utilizaremos modelará tanto la parte variable como estática del error de estimación. Las hipótesis de trabajo, variables, parámetros y las relaciones entre ellas se detallan a continuación.

## Hipótesis del modelo

- A1.** El error en la altitud medida por el sistema de altimetría consiste en una componente sistemática (bias), que es constante durante todo el encuentro, y una componente variable (jitter), que es variable durante el encuentro.
- A2.** El error sistemático del sistema de altimetría se modela con una función de densidad de probabilidad normal o laplaciana.
- A3.** El error variable del Sistema de altimetría se modela con un proceso auto-regresivo de primer orden con ruido gaussiano blanco.

## Variables

Variable	Unidades	Explicación
$S_{t,PAS,i}^z$	m	Altitud de presión estándar medida por el sistema de altimetría de presión (PAS: Pressure Altimetry System)
$\mathcal{E}_{t,PAS,i}^{z,bias}$	m	Error sistemático (bias) en la altitud de presión, representa las contribuciones estáticas del error
$\mathcal{E}_{t,PAS,i}^{z,jitter}$	m	Error aleatorio (jitter) en la altitud de presión, representa las contribuciones variables al error
$\sigma_{\tau_0,PAS,i}^{z,bias}$	m	Desviación típica del error bias de la altitud de presión, que es dependiente con la altitud
$\sigma_{\tau_0,PAS,i}^{z,jitter}$	m	Desviación típica del error jitter de la altitud de presión, dependiente de la altitud
$\mathcal{E}_{t,PAS,i}^{noise}$	m	Componente de ruido blanco del modelo auto regresivo de primer orden del error del jitter de la altitud de presión

### Configuración determinista del error en la altitud de presión

Para el caso determinista, ambos errores, el sistemático y el variable de la altitud de presión son nulos:

$$\varepsilon_{t,PAS,i}^{s,bias} = 0$$

$$\varepsilon_{t,PAS,i}^{s,jitter} = 0$$

### Configuración estocástica del error en la altitud de presión

El modelo para el error sistemático representa un error constante, cuyo valor se fija al inicio del encuentro. Este valor se escoge de una distribución gaussiana centrada en cero, como está indicado en los MOPS, con una desviación típica dependiente de la altitud.

$$\{\varepsilon_{t \geq \tau_0, PAS, i}^{s,bias}\} = f^N(0, \sigma_{\tau_0, PAS, i}^{s,bias})$$

El modelo se podría parametrizar para tener otra variable que nos indicaría si generar el error sistemático (“bias”) según una distribución normal (modelo según ACAS MOPS), o laplaciana (modelo según Anexo 10 de ICAO).

Por otra parte, la desviación típica de la distribución es dependiente con la altitud de vuelo inicial de la aeronave:

$$\sigma_{\tau_0, PAS, i}^{s,bias} = f_{PAS}^{s,bias}(s_{\tau_0, AC, i}^s; A_{PAS, i}^{bias})$$

La función que implementa la dependencia con la altitud utiliza una matriz  $A_{PAS, i}^{bias}$  con pares de valores altitud-desviación típica:

$$A_{PAS,i}^{bias} = \begin{pmatrix} z_1 & \sigma_1 \\ \vdots & \vdots \\ z_k & \sigma_k \\ \vdots & \vdots \\ z_n & \sigma_n \end{pmatrix}$$

con  $z_k > z_{k-1}$ . En caso de que la altitud no coincide con ninguno de los pares de valores altitud-desviación típica, la función utiliza interpolación lineal entre las altitudes más cercanas de  $A_{PAS,i}^{bias}$  para determinar la desviación típica.

$$f_{PAS}^{z,bias}(S_{t_0,AC,i}^z; A_{PAS,i}^{bias}) = \begin{cases} \sigma_1 & \text{if } S_{t_0,AC,i}^z < z_1 \\ \frac{\sigma_{k+1} - \sigma_k}{z_{k+1} - z_k} (S_{t_0,AC,i}^z - z_k) + \sigma_k & \text{if } z_k \leq S_{t_0,AC,i}^z < z_{k+1}, \text{ for } k = 1, \dots, n-1 \\ \sigma_n & \text{if } S_{t_0,AC,i}^z \geq z_n \end{cases}$$

El modelo de error para el "jitter" en el sistema de altimetría es un modelo auto-regresivo de primer orden:

$$\varepsilon_{t,PAS,i}^{z,jitter} = \begin{cases} \varepsilon_{t,PAS,i}^{noise} & \text{if } t = 0 \\ \alpha_{PAS,i}^{auto} \cdot \varepsilon_{t-T_{PAS,i}^{sample},PAS,i} + \varepsilon_{t,PAS,i}^{noise} & \text{if } t > 0 \end{cases}$$

$$\{\varepsilon_{t,PAS,i}^{noise}\} = f^N \left( 0, \sigma_{PAS,i}^{z,jitter} \cdot \sqrt{1 - (\alpha_{PAS,i}^{auto})^2} \right)$$

## Modelo de la altitud de presión – Resultado final

La altitud de presión medida por el Sistema de altimetría es igual a la altitud real de la aeronave más los dos componentes del error, el error sistemático y el aleatorio:

$$S_{t,PAS,i}^z = S_{t,AC,i}^z + \mathcal{E}_{t,PAS,i}^{z,bias} + \mathcal{E}_{t,PAS,i}^{z,jitter}$$

### Parámetros del modelo

Para la simulación durante esta tesis se utilizarán los siguientes parámetros para el modelo de la altitud de presión.

**Tabla 6. Parámetros del modelo estocástico de la altitud de presión para la simulación. Basados en los MOPS de ACAS [17, 18].**

Parameter	Description	Value	Comment
$A_{PAS,i}^{bias}$	Matriz con las altitudes y sus desviaciones típicas del error "bias" correspondientes	(0, 13.7; 1524, 14.6; 3048, 15.8; 4572, 17.7; 6096, 19.8; 7620, 21.6; 9144, 23.8; 10668, 26.2; 12192, 29.0) m	Los valores se basan en la Tabla 1 y su corrección para el jitter $\sigma_{PAS,i}^{z,jitter}$ , ya que los MOPS consideran el error total
$T_{PAS,i}^{sample}$	Tiempo de muestreo para el Sistema de altitud de presión	1 sec	
$\alpha_{PAS,i}^{auto}$	Factor de auto-correlación del "jitter" para el sistema de altimetría	0.88	Se basa en un tiempo de auto-correlación de 8 s
$\sigma_{PAS,i}^{z,jitter}$	Desviación típica del "jitter" para la altitud de presión	1.5 m	

### 4.2.3 Rumbo

El rumbo de la aeronave propia se estima a través de los sensores del HRS (“Heading Reference System”), que cuenta con giróscopos, acelerómetros y magnetómetros.

En ACAS X, la estimación del rumbo se usa para determinar el ángulo de azimut absoluto de las aeronaves intrusas ( [3], Sección 1.4.1.4). Además, uno de los requisitos de entrada relativo al rumbo, es una tasa mínima de refresco de 1 Hz.

El modelo de ruido utilizado es el descrito en [6] sección 3.D.3: “Heading estimation”.

#### Hipótesis del modelo

Existen dos configuraciones de simulación:

- Configuración determinista: no hay error en la estimación del rumbo.
- Configuración estocástica. El error en la estimación del azimut se describe según un modelo Gauss-Markov autor-regresivo de primer orden con ruido Gaussiano blanco y desviación estándar fija.

#### Variables

Variable	Unidades	Explicación
$\varphi_{I,HRS,i}$	grados	Rumbo de la aeronave medido por el sistema HRS

$\varepsilon_{t,HRS,i}$	grados	Error en el rumbo
$\varepsilon_{t,HRS,i}^{noise}$	grados	Ruido blanco del modelo auto-regresivo

### Modelo determinista del error del rumbo

En el caso determinista, el error en la estimación del rumbo es nulo.

$$\varepsilon_{t,HRS,i} = 0$$

### Modelo estocástico del error del rumbo

Para la configuración estocástica, el modelo de error del rumbo modela un error variante en el tiempo:

$$\varepsilon_{t,HRS,i} = \begin{cases} \varepsilon_{t,HRS,i}^{noise} & \text{if } t = 0 \\ \alpha_{HRS}^{auto} \cdot \varepsilon_{t-1,HRS,i} + \varepsilon_{t,HRS,i}^{noise} & \text{if } t > 0 \end{cases}$$

$$\{\varepsilon_{t,HRS,i}^{noise}\} \in f^N(0, \sigma_{HRS,i}^{noise} \sqrt{1 - (\alpha_{HRS,i}^{auto})^2})$$

### Modelo de estimación del rumbo

El rumbo medido por el Sistema HRS será igual al rumbo verdadero más el error del rumbo modelado:

$$\varphi_{t,HRS,i} = \varphi_{t,AC,i} + \varepsilon_{t,HRS,i}$$

## Parámetros para la simulación

Para las simulaciones de Monte Carlo en el capítulo 8 se utilizará la siguiente configuración:

**Tabla 7. Parámetros del modelo de ruido del rumbo para la configuración estocástica.**

Parametro	Descripción	Valor
$T_{HRS,i}^{sample}$	Tiempo de muestreo del Sistema HRS	1 s
$\alpha_{HRS,i}^{auto}$	Factor de auto-correlación del ruido en el rumbo	0.5
$\sigma_{HRS,i}^{noise}$	Desviación típica del ruido del rumbo	5°

### 4.2.4 Distancia oblicua

La distancia oblicua es determinada por el sistema de radar secundario o SSR (“Secondary Surveillance Radar”), que consta de un transpondedor (XPNDR) para transmitir y recibir las interrogaciones.

La distancia entre las dos aeronaves se determina midiendo el tiempo que tarda la interrogación en llegar a la aeronave intrusa y el tiempo que tarda la respuesta en volver a la aeronave propia, y teniendo en cuenta la velocidad de propagación de la onda de radiofrecuencia.

#### *Estándares*

Los requisitos para la distancia oblicua medida por el transpondedor son prácticamente iguales en TCAS II ( [15], sección 2.2.2.2.3) y ACAS X ( [3], sección 2.2.2.2.3). La precisión de la medida que consigue el sistema TCAS depende de las fuentes de error:

- a. Tiempo de retardo en la respuesta del transpondedor
- b. Diferencia de tiempo de retardo entre la señal capturada por la antena superior y la inferior.
- c. Diferencia de tiempo en la transmisión de la interrogación por la antena superior y la inferior de la aeronave.
- d. Cuantización del reloj TCAS.
- e. Ruido del sistema TCAS.

Para señales recibidas con un nivel de potencia de al menos 6 dB por encima del MTL (“Minimum Transponder Sensitivity Level”) y para transpondedores que cumplan con las características especificadas en los MOPS del transpondedor modo S: ED-73A/DO-181A, TCAS debería obtener una precisión en la medida de la distancia oblicua con un error (“jitter”) inferior a 50 ft RMS para transpondedores Modo C y Modo S; y un error de sesgo (“bias”) de 250 ft para Modo C y 125 ft para Modo S.

En el siguiente apartado también se presenta un modelo de error estándar para la distancia oblicua según el Anexo 10 de ICAO [29](Sección 4.4.2.2). Los errores en la medida de la distancia oblicua de este modelo se basan en una distribución normal con media cero y desviación típica 50 ft (15.24 m). Comparándolo con el descrito anteriormente según los MOPS, se puede observar que el modelo de ICAO no diferencia entre “jitter” y “bias”, ni tampoco tiene en cuenta el efecto del Modo en el que opera el transpondedor (“C” o “S”), y el valor de la desviación típica es igual al “jitter” indicado en los MOPS (50 ft).

### ***Modelo de ruido para la distancia oblicua***

La medida de la distancia oblicua representa la distancia entre la aeronave intrusa y la aeronave propia. El modelo de error incluye una parte estática y una parte variable, que depende del modo del transpondedor. Las hipótesis,

variables, parámetros y relaciones del modelo que se presentan a continuación son las que publicamos en [6] (sección 3.E.1: “Slant range measurement”).

### Hipótesis del modelo

Existen dos configuraciones de simulación:

- Configuración determinista: no hay error en la distancia oblicua.
- Configuración estocástica:
  - El error se compone de una parte estática (“bias”), y una variable (“jitter”)
  - El “bias” se modela con una distribución gaussiana de media cero y desviación típica dependiente del modo del transpondedor.
  - El “jitter” se modela con un modelo auto regresivo de primer orden con ruido gaussiano blanco de desviación típica dependiente del modo del transpondedor.

### Variables

Variable	Unidades	Explicación
$r_{t,i}^j$	m	distancia oblicua estimada de la aeronave i respecto de la aeronave j
$\varepsilon_{t,i}^{r,bias}$	m	Error de sesgo en la distancia oblicua estimada (bias)
$\varepsilon_{t,i}^{r,jitter}$	m	Error de jitter en la medida de distancia oblicua del transpondedor
$m$	{ <i>ModoS</i> , <i>ModoC</i> }	Modo del transpondedor
$\varepsilon_{t,i}^{noise}$	m	Componente de ruido blanco del modelo auto regresivo de primer orden para la distancia oblicua

## Configuración determinista para el modelo de ruido de la distancia oblicua

En el caso determinista, el error de “bias” y “jitter” son nulos:

$$\begin{aligned}\mathcal{E}_{t,TS,i}^{r,bias} &= 0 \\ \mathcal{E}_{t,TS,i}^{r,jitter} &= 0\end{aligned}$$

## Configuración estocástica para el modelo de ruido de la distancia oblicua

En el caso estocástico, el error se modela con un error constante o bias, cuyo valor se determina al comienzo del encuentro ( $\tau_0$ ). Este valor se determina de acuerdo con una distribución gaussiana, cuya desviación típica dependerá del modo del transpondedor:

$$\{\mathcal{E}_{t,i}^{r,bias}\} \in f^N(0, \sigma_i^{r,bias})$$

El modelo de error para el error de “jitter” representa a un error dependiente del tiempo, cuyo valor se determina en los siguientes tiempos de muestreo  $t = \tau_0 + k T^{sample}$  (con  $k = 1, 2, 3\dots$ ).

$$\begin{aligned}\mathcal{E}_{t,TS,i}^{r,jitter} &= \alpha_{TS,i}^{auto,r} \mathcal{E}_{t-T_{TS}^{sample},TS,i}^{r,jitter} + \mathcal{E}_{t,TS,i}^{noise} \\ \{\mathcal{E}_{t,TS,i}^{noise}\} &\square f^N(0, \sigma_{TS,m_{TS,i}^j}^{r,jitter} \sqrt{1 - (\alpha_{TS,i}^{auto,r})^2})\end{aligned}$$

## Modelo de estimación de la distancia oblicua

La distancia oblicua estimada entre la aeronave j respecto de la aeronave I (estimada por tanto por la aeronave i) es igual la distancia real entre las aeronaves más las componentes del error de “bias” y el de “jitter”.

$$r_{t,i}^j = d_{real} + \mathcal{E}_{t,i}^{r,bias} + \mathcal{E}_{t,i}^{r,jitter}$$

## Parámetros de la simulación

Para las simulaciones de Monte Carlo en el capítulo 7 se utilizará el modelo de ruido de ICAO para un transpondedor en modo S.

En las siguientes tablas se representan los parámetros de los dos modelos de ruido vistos anteriormente, el modelo de los MOPS, y el modelo de ICAO.

**Tabla 8. Parámetros para el modelo de error de la distancia oblicua según los MOPS.**

Parameter	Description	Value
$\sigma_{TS,ModeS,i}^{r,bias}$	Desviación típica del error de bias para la distancia oblicua con un transpondedor en Modo S	38.1 m (125 ft)
$\sigma_{TS,ModeC,i}^{r,bias}$	Desviación típica del error de bias para la distancia oblicua con un transpondedor en Modo C	76.2 m (250 ft)
$\sigma_{TS,ModeS,i}^{r,jitter}$	Desviación típica del error de jitter para la distancia oblicua con un transpondedor en Modo S	15.2 m (50 ft)
$\sigma_{TS,ModeC,i}^{r,jitter}$	Desviación típica del error de jitter para la distancia oblicua con un transpondedor en Modo C	15.2 m (50 ft)
$T_{TS,i}^{sample}$	Tiempo de muestreo del transpondedor	1 s
$\alpha_{TS,i}^{auto,r}$	Factor de autocorrelación del jitter para la distancia oblicua	0

**Table 9. Parámetros para el modelo de error de la distancia oblicua según ICAO.**

Parameter	Description	Value
$\sigma_{TS,ModeS,i}^{r,bias}$	Desviación típica del error de bias para la distancia oblicua con un transpondedor en Modo S	0 m
$\sigma_{TS,ModeC,i}^{r,bias}$	Desviación típica del error de bias para la distancia oblicua con un transpondedor en Modo C	0 m
$\sigma_{TS,ModeS,i}^{r,jitter}$	Desviación típica del error de jitter para la distancia oblicua con un transpondedor en	15.2 m (50 ft)

	Modo S	
$\sigma_{TS,ModeC,i}^{r,jitter}$	Desviación típica del error de jitter para la distancia oblicua con un transpondedor en Modo C	15.2 m (50 ft)
$T_{TS,i}^{sample}$	Tiempo de muestreo del transpondedor	1 s
$\alpha_{TS,i}^{auto,r}$	Factor de autocorrelación del jitter para la distancia oblicua	0

### 4.2.5 Azimut

El ángulo de azimut, o “bearing” en inglés, es el ángulo de la aeronave intrusa en el plano horizontal, medido en sentido horario desde el eje longitudinal de la aeronave propia hasta la aeronave intrusa.

### *Estandares y modelos*

#### **ACAS MOPS**

Los requisitos operacionales para las medidas del ángulo de azimut basadas en el transpondedor modo S y C son iguales para TCAS II [15](Sección 2.2.4.6.4) y ACAS X [3](Sección 2.2.4.6.4). Los requisitos sobre la exactitud de las medidas son:

- Para ángulos de elevación desde -10 hasta +10 grados, el error en el azimut no debe exceder 9 grados RMS, o 27 grados de pico, en ausencia de interferencias multi-trayectoria (“multipath”), usando modo S o modo C.
- Para ángulos de elevación mayores de 10 grados hasta 20 grados, el error en el azimut no debe exceder 15 grados RMS o 45 grados de pico, en ausencia de interferencias multi-trayectoria, usando interrogaciones modo S o modo C.
- El error RMS en presencia de interferencias no debe aumentar en más de un grado para interrogaciones modo C, y no debe aumentar más de 3 grados en interrogaciones modo S.

N.B. el ángulo de elevación es el ángulo entre la aeronave intrusa en el plano vertical con respecto al eje longitudinal de la aeronave propia.

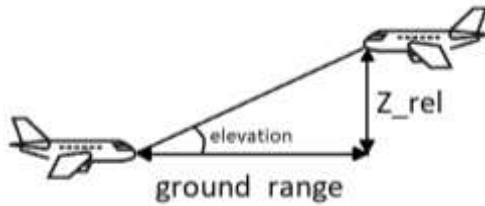


Ilustración 18. Ángulo de elevación

## Anexo 10 de ICAO

Un modelo estándar para el error en la medida del ángulo de azimut para simulaciones de ACAS II se presenta en el Anexo 10 de ICAO [29] (Sección 4.4.2.3). Los errores según este modelo se toman a partir de una distribución normal con media de cero grados y desviación típica de 10 grados. Las medidas del ángulo de azimut para cada aeronave intrusa están disponibles cada ciclo de ACAS siempre y cuando la aeronave intrusa este en un rango de 14 NM, pero no cuando la distancia excede 14 NM (Sección 4.4.2.1a de [2]). Cabe destacar que este modelo del error no incluye ninguna dependencia del error con el ángulo de elevación como se indicaba en los MOPS; y el valor de la desviación típica es similar al error RMS descrito en los MOPS para ángulos de elevación desde -10 hasta +10 grados.

### *Modelo de ruido para medidas de azimut*

El modelo que se presenta a continuación será el usado durante las simulaciones de esta tesis. Este modelo representa la medida del ángulo de azimut de una aeronave (propia) respecto de otra (intrusa). Incluye un modelo de error que describe los diferentes componentes del error, que dependen del modo del transpondedor (Modo S o modo C) y del ángulo de elevación. Las hipótesis del modelo junto con sus variables, parámetros y relaciones entre ellas se presentan a continuación.

## Hipótesis del modelo

- En una simulación determinista (configuración determinista del modelo de ruido) el error en la medida es nulo.
- El error de la medida se describe a través de un modelo auto-regresivo de primer orden con ruido gaussiano blanco y una desviación típica que depende del ángulo de elevación con respecto de la aeronave intrusa y el modo del transpondedor.

## Variables

Variable	Unidades	Explicación
$\xi_{t,XPDR,i}^{j,rel}$	grados	Medida del ángulo de azimut de la aeronave j respecto de la i, (ángulo respecto del eje longitudinal de la aeronave i)
$\varepsilon_{t,XPDR,i}^{\xi,j}$	grados	Error total en la medida del ángulo de azimut de la aeronave j respecto la i
$m_{XPDR,i}^j$	{ModeS, ModeC}	Modo del transpondedor (XPDR) de la aeronave j, según es recibido por la aeronave i
$\vartheta_{t,i}^j$	grados	Ángulo de elevación de la aeronave j respecto de la i
$\xi_{t,i}^{j,abs}$	grados	Ángulo de azimut absoluto de la aeronave j, respecto de la i (ángulo respecto del Norte)

## Configuración determinista para modelo de ruido de azimut

Para el caso determinista, el error total en la medida del ángulo de azimut es nulo.

$$\varepsilon_{t,XPDR,i}^{\xi,j} = 0$$

## Configuración estocástica para modelo de ruido de azimut

Para la configuración estocástica el error total  $\varepsilon_{t,XPDR,i}^{\xi,j}$ , será variable en el tiempo según un modelo auto-regresivo de primer orden, sin componente de error sistemático o bias:

$$\varepsilon_{t,XPDR,i}^{\xi,j} = \alpha^{\xi} \cdot \varepsilon_{t-1,XPDR,i}^{\xi,j} + \varepsilon_{t,XPDR,i}^{\xi,j,noise}$$

$$\{\varepsilon_{t,XPDR,i}^{\xi,j,noise}\} \sim f^N \left( 0, \quad \sigma_{t,XPDR,i}^{\xi,j,noise} \cdot \sqrt{1 - (\alpha_{XPDR,i}^{auto,\xi})^2} \right)$$

La desviación típica del ruido del proceso dependerá del ángulo de elevación de la aeronave i con respecto de la aeronave j:

$$\sigma_{t,XPDR,i}^{\xi,j,noise} = \begin{cases} \sigma_{t,XPDR,L,i}^{\xi,j,noise} & \text{if } |\vartheta_{t,i}^j| \leq \vartheta_{XPDR,i}^{\xi} \\ \sigma_{t,XPDR,H,i}^{\xi,j,noise} & \text{if } |\vartheta_{t,i}^j| > \vartheta_{XPDR,i}^{\xi} \end{cases}$$

$$\vartheta_{t,i}^j = \arcsin \left( \frac{s_{t,j}^z - s_{t,i}^z}{|s_{t,j} - s_{t,i}|} \right) - \theta_{t,i}$$

Donde  $s_{t,j}$  y  $s_{t,i}$  son las posiciones tridimensionales de las dos aeronaves ("i" y "j") y  $\theta_{t,i}$  es el ángulo de cabeceo de la aeronave i.

## Medida del ángulo de azimut según el transpondedor

El transpondedor mide el ángulo de azimut relativo de la aeronave j respecto de la aeronave i, esto quiere decir que lo hace respecto del rumbo de la aeronave i ( $\varphi_{t,i}$ ), además contendrá un término de error modelado anteriormente y estará empaquetado en el rango [-180, 180] grados:

$$\xi_{t,XPDR,i}^{j,rel} = f_{wrap}^{180}(\xi_{t,i}^{j,abs} + \varepsilon_{t,XPDR,i}^{\xi,j} - \varphi_{t,i})$$

$$\xi_{t,i}^{j,abs} = \text{atan2}(s_{t,j}^x - s_{t,i}^x, s_{t,j}^y - s_{t,i}^y)$$

$$f_{wrap}^{180}(\alpha): \alpha \rightarrow \begin{cases} \alpha - 360 & \text{if } \alpha > 180 \\ \alpha & \text{if } -180 \leq \alpha \leq 180 \\ \alpha + 360 & \text{if } \alpha < -180 \end{cases}$$

## Parámetros

Los parámetros del modelo presentado anteriormente utilizarán los siguientes valores para las simulaciones:

Tabla 10. Parámetros del modelo de error del ángulo de azimut ara las simulaciones.

Parámetro	Descripción	Valor	Comentario
$\alpha^{\xi}$	Factor de auto-correlación del error aleatorio	0	No hay correlación
$\vartheta_{XPDR,i}^{\xi}$	Umbral del ángulo de elevación	10 grados	Sección 2.2.4.6.4 [15]
$\sigma_{XPDR,modeS,L,i}^{\xi,j,noise}$	Desviación típica Standard del error del ángulo de azimut para ángulos de elevación bajos e interrogaciones Modo S	12 grados	Sección 2.2.4.6.4 [15]
$\sigma_{XPDR,modeS,H,i}^{\xi,j,noise}$	Desviación típica del error del ángulo de azimut para altos ángulos de elevación e interrogaciones modo S	18 grados	Sección 2.2.4.6.4 [15]
$\sigma_{XPDR,modeC,L,i}^{\xi,j,noise}$	Desviación típica del error del ángulo de azimut para bajos	10 grados	Sección 2.2.4.6.4 [15]

	ángulos de elevación e interrogaciones modo C		
$\sigma_{XPDR,modeC,H,i}^{\xi,j,noise}$	Desviación típica del error del ángulo de azimut para altos ángulos de elevación e interrogaciones modo C	16 grados	Sección 2.2.4.6.4 [15]

## 5. PARADIGMA DE MÚLTIPLES MODELOS

En el Anexo A se hace un repaso de filtros para dinámicas donde la matriz de estado, la matriz de medida y las matrices de covarianza eran conocidas e incluso constantes en el tiempo [Anexo A]. Estos filtros tratan la incertidumbre de la medida, del proceso, de las no linealidades y del ruido coloreado.

Por otro lado, el paradigma de múltiples modelos (MM) intenta adaptarse a ciertos tipos de incertidumbres no vistos hasta ahora como la incertidumbre sobre las entradas del sistema (importante para blancos que maniobran) o incertidumbre en la matriz de estado. El algoritmo MM asume que el blanco se comporta de acuerdo a uno de un número finito de modelos. Cada modelo representa un modo de operación, y pueden ser totalmente diferentes (e.g. diferentes niveles de ruido), este algoritmo funciona muy bien para los llamados sistemas híbridos que tienen incertidumbres en variables discretas (modo de operación del sistema) y continuas (ruido).

En concreto veremos el algoritmo IMM (“Interacting Multiple Model”) dinámico, en el que el modelo que se usa para la estimación va cambiando de acuerdo a una cadena de Markov. El algoritmo IMM es subóptimo, ya que la solución óptima del estimador de múltiples modelos dinámico aumenta su complejidad exponencialmente con el tiempo, y por tanto su uso no es práctico en sistemas en tiempo real.

Otras técnicas para objetivos que maniobran son algoritmos adaptativos como:

1. Ajustar/ adaptar el nivel de ruido del proceso (“continuous noise level adjustment”), si la innovación excede un umbral, suponemos que el objetivo ha empezado a maniobrar, y aumentamos (“fudge factor”) el ruido del proceso ( $Q$ ), de forma que la innovación vuelva a niveles por debajo del umbral seleccionado.

2. Niveles discretos de ruido de proceso ("Process noise with several discrete levels"): Otra opción similar a la anterior, es tener dos niveles de ruido, uno alto y otro bajo, y usar uno u otro dependiendo de si detectamos que el objetivo está maniobrando o no, para pasar de un nivel a otro se monitoriza que la innovación no pase de un umbral determinado, o se puede utilizar una media móvil exponencial de la innovación ("fading memory average").
3. Modelando la entrada del filtro ("input estimation"), si tenemos dos filtros corriendo en paralelo uno para objetivos que no maniobran y otro con objetivos que maniobran, seremos capaces de estimar la entrada del filtro para blancos que maniobran observando las innovaciones del otro filtro Pero ninguna de estas técnicas consigue mantener el error por debajo del error de la medida cuando las maniobras comienzan, por lo que recurriremos a las técnicas de múltiples modelos.

### **Detección de maniobra y de fin de maniobra**

Para cambiar de un modelo a otro, necesitamos detectar cuando el blanco empieza a maniobrar, y cuando el blanco termina de maniobrar. La detección del comienzo de la maniobra es más sencilla, ya que el modelo del blanco que no maniobra tiene menor covarianza de la innovación, por lo que una innovación más grande de lo normal destacará más. Sin embargo, una vez que ya ha comenzado la maniobra y estamos utilizando el modelo del blanco que maniobra, la matriz de covarianza de la innovación será mucho mayor (aunque solo sea por el hecho de añadir nuevas variables de estado, e.g. aceleraciones, al vector de estado), por lo que será más difícil detectar una anomalía en la innovación.

Una de las ventajas del algoritmo IMM es que está libre de decisión, ya que como veremos más adelante, el algoritmo no necesita decidir o determinar cuando la maniobra empieza o acaba.

- Estimador de múltiples modelos estático: el estimador cuenta con varios modelos del sistema, cada uno con una probabilidad  $\mu_j$  de ser el modelo correcto. Finalmente obtiene la estimación como una combinación lineal de las estimaciones de cada modelo ponderadas con la probabilidad de cada modelo de ser el correcto, por lo que no existe un salto de un modelo a otro. Este filtro acaba convergiendo a la estimación del modelo que es realmente correcto o en su defecto al que más se parece, por lo que se utiliza cuando se cumplen las siguientes hipótesis:
  - El modelo correcto se encuentra dentro del conjunto de modelos
  - El mismo modelo está en uso en el sistema desde el principio (es decir, por ejemplo, si la aeronave comienza maniobrando, debe estar maniobrado siempre)
- Estimador de múltiples modelos dinámico: este filtro permite cambios en el modo, por tanto hay dos “estados” a estimar, el estado continuo discretizado  $x(k)$  (denominado ahora estado básico), y el estado discreto  $M(k)$  o estado modal.

El filtro IMM es una técnica de filtrado que se basa en el paradigma de los múltiples modelos.

Para cada instante  $k$ , el estado estimado se determina usando cada uno de los “ $r$ ” modelos posibles, y en cada filtro se utilizan varias combinaciones diferentes de los vectores de estado estimados previos condicionados a cada modelo. Es decir, utilizando el teorema de probabilidad total, la estimación sería:

$$P(x(k)|Z^k) = \sum_{j=1}^r p(x(k)|M_j(k), Z^k) \cdot P\{M_j(k)|Z^k\}$$

$$P\{M_j(k)|Z^k\} = \mu_j(k)$$

$$P(x(k)|Z^k) = \sum_{j=1}^r p(x(k)|M_j(k), z(k), Z^{k-1}) \cdot \mu_j(k)$$

Para llevar a cabo esta implementación, el algoritmo consta de los siguientes pasos:

### 1- Interacción:

En inglés “Interaction” o “Mixing”, consiste en determinar las probabilidades mezcladas, es decir, para nuestro caso la probabilidad de que el modo  $i$  fuera el verdadero en  $k-1$ , dado que el modo  $j$  es el verdadero en el instante  $k$ , y condicionado por el  $\sigma$ -álgebra  $Z^{k-1}$ , es decir, todas las medidas anteriores al instante  $k$ , como se propuso en [35].

N.B.  $\sigma$ -álgebra sobre un conjunto  $X$  es una colección  $\Sigma$  no vacía de subconjuntos de  $X$ .

$$\mu_{i|j}(k-1|k-1) = P(M_i(k-1)|M_j(k), Z^{k-1})$$

Aplicando el teorema de Bayes:

$$\mu_{i|j}(k-1|k-1) = \frac{1}{c_j} \cdot P(M_j(k)|M_i(k-1), Z^{k-1}) \cdot P(M_i(k-1)|Z^{k-1})$$

Por definición sabemos que:

$$P(M_j(k)|M_i(k-1), Z^{k-1}) = p_{ij}$$

$$P(M_i(k-1)|Z^{k-1}) = \mu_i(k-1)$$

Donde  $p_{ij}$  es la probabilidad de pasar del modo  $i$  al  $j$ , según la matriz de transición de la cadena de Markov. Y  $\mu_i(k-1)$  es la probabilidad de que en el instante  $k-1$  el modo de nuestro sistema fuera el modo  $i$ .

$$\mu_{i|j}(k-1|k-1) = \frac{1}{c_j} \cdot p_{ij} \cdot \mu_i(k-1)$$

Donde  $\bar{c}_j$  es la constante de normalización, que según el teorema de Bayes debería ser la probabilidad marginal de la hipótesis (hipótesis: el sistema está en modo j), es decir, para nuestro algoritmo que cuenta con dos modos sería la probabilidad de estar en el modo 1, que es la probabilidad de pasar el modo 1, estando anteriormente en el modo 2, más la probabilidad de permanecer en el modo 1, estando anteriormente en el modo 1:

$$\begin{aligned}\bar{c}_1 &= P(M_1(k) | M_2(k-1)) + P(M_1(k) | M_1(k-1)) \\ &= p_{21} \cdot \mu_2(k-1) + p_{22} \cdot \mu_2(k-1)\end{aligned}$$

$$\bar{c}_j = \sum_{i=1}^r p_{ij} \cdot \mu_i(k-1)$$

Por tanto para nuestro algoritmo que consta de dos modos (aeronave sin maniobrar, y aeronave maniobrando), las ecuaciones que determinan las probabilidades mezcladas son:

$$\mu_{1|1}(k-1|k-1) = \frac{p_{11} \cdot \mu_1(k-1)}{p_{11} \cdot \mu_1(k-1) + p_{21} \cdot \mu_2(k-1)}$$

$$\mu_{1|2}(k-1|k-1) = \frac{p_{12} \cdot \mu_1(k-1)}{p_{12} \cdot \mu_1(k-1) + p_{22} \cdot \mu_2(k-1)}$$

$$\mu_{2|1}(k-1|k-1) = \frac{p_{21} \cdot \mu_2(k-1)}{p_{11} \cdot \mu_1(k-1) + p_{21} \cdot \mu_2(k-1)}$$

$$\mu_{2|2}(k-1|k-1) = \frac{p_{22} \cdot \mu_2(k-1)}{p_{12} \cdot \mu_1(k-1) + p_{22} \cdot \mu_2(k-1)}$$

Una vez que tenemos las probabilidades mixtas, el siguiente paso es calcular las “condiciones iniciales mezcladas”, que no es más que una media ponderada de las estimaciones del filtro 1 y 2

$$x^{0j}(k-1, k-1) = \sum_{i=1}^r \hat{x}^i(k-1, k-1) \cdot \mu_{ij}(k-1, k-1)$$

Por tanto en nuestro caso:

$$x^{01}(k-1) = \hat{x}^1(k-1) \cdot \mu_{1|1}(k-1) + \hat{x}^2(k-1) \cdot \mu_{2|1}(k-1)$$

$$x^{02}(k-1) = \hat{x}^1(k-1) \cdot \mu_{1|2}(k-1) + \hat{x}^2(k-1) \cdot \mu_{2|2}(k-1)$$

Y sus matrices de covarianza es:

$$P^{0j}(k-1) = \sum_{i=1}^r \mu_{i|j}(k-1) \{P^i(k-1) + [x^i(k-1) - x^{0j}(k-1)] \cdot [x^i(k-1) - x^{0j}(k-1)]'\}$$

## 2 – Filtrado por modo:

En esta fase se ejecutan los filtros de cada modo, en nuestro caso son dos filtros, por lo que obtenemos los estados estimados de cada filtro, junto con sus covarianzas; y la verosimilitud de cada hipótesis (“likelihood”) de que el sistema esté realmente en cada uno de los modos.

La probabilidad de hipótesis (“likelihood”) (ver diferencia entre “probability” y “likelihood” en el capítulo 3), nos proporciona una indicación de que la probabilidad de que la hipótesis sea cierta (es decir, si la aeronave está o no maniobrando), de acuerdo a las medidas observadas:

$$\Lambda_j(k) = p(z(k)|M_j(k), Z^{k-1})$$

La probabilidad de hipótesis se determinará analizando los residuos de cada filtro, a mayor residuo del filtro, menor es la probabilidad de que sea cierta la hipótesis de ese filtro; para ello será necesario comparar el residuo con la matriz de covarianza esperada de la innovación del filtro.

### 3- actualización de la probabilidad de modo:

Una vez que hemos determinado la probabilidad de que cada hipótesis sea cierta basándonos en las medidas observadas durante el ciclo actual, se deben actualizar las probabilidades de cada modo  $\mu_j(k)$ .

$$\mu_j(k) = P(M_j(k)|Z^k)$$

$$\mu_j(k) = \frac{1}{c} \cdot \Lambda_j(k) \bar{c}_j$$

Donde c es la constante de normalización:

$$c = \sum_{j=1}^r \Lambda_j(k) \cdot \bar{c}_j$$

$$\mu_1(k) = \frac{\Lambda_1(k) \cdot [p_{11} \cdot \mu_1(k-1) + p_{21} \cdot \mu_2(k-1)]}{\Lambda_1(k) \bar{c}_1 + \Lambda_2(k) \bar{c}_2}$$

$$\mu_2(k) = \frac{\Lambda_2(k) \cdot [p_{12} \cdot \mu_1(k-1) + p_{22} \cdot \mu_2(k-1)]}{\Lambda_1(k) \bar{c}_1 + \Lambda_2(k) \bar{c}_2}$$

### 4 – Combinación de las estimaciones y covarianzas:

Finalmente se combinan las estimaciones y covarianzas condicionadas por cada uno de los modelos, ponderándolas con la probabilidad de cada modelo.

$$\hat{x}(k, k) = \sum_{j=1}^r \hat{x}^j(k, k) \cdot \mu_j(k)$$

$$P(k, k) = \sum_{j=1}^r \mu_j(k) \cdot [P^j(k, k) + [\hat{x}^j(k, k) - \hat{x}(k, k)] \cdot [\hat{x}^j(k, k) - \hat{x}(k, k)]]$$

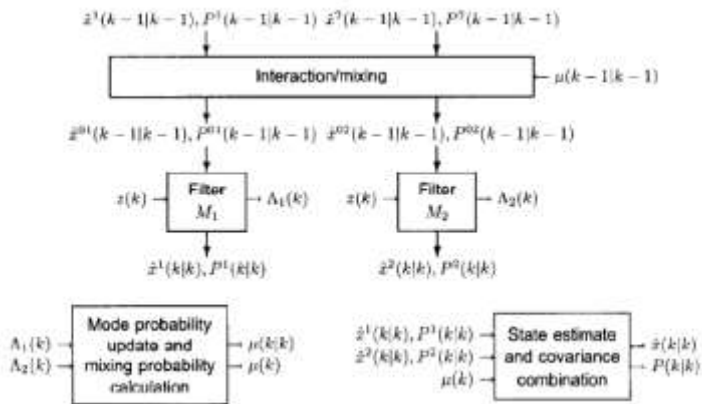


Ilustración 19. Un ciclo del filtro IMM.

Como bien sabemos el movimiento vertical esta desacoplado del lateral en una aeronave, por lo que el filtro IMM a diseñar deberá realizar el seguimiento solo de la posición horizontal. Respecto al movimiento horizontal vamos a considerar dos tipos o modos de movimiento, por lo que el filtro IMM requerirá de dos filtros:

- Movimiento uniforme: o modelo NCV (Nearly Constant Velocity) que representa vuelo a velocidad y rumbo constante. Este modelo se implementará como un modelo cinemático de segundo orden, donde el ruido blanco representa la aceleración.
- Maniobra: en concreto un viraje coordinado, por lo que implementaremos un modelo NCT (Nearly Coordinated Turn)

A modo de resumen, el flujo del algoritmo IMM está representado en el siguiente diagrama de flujo:

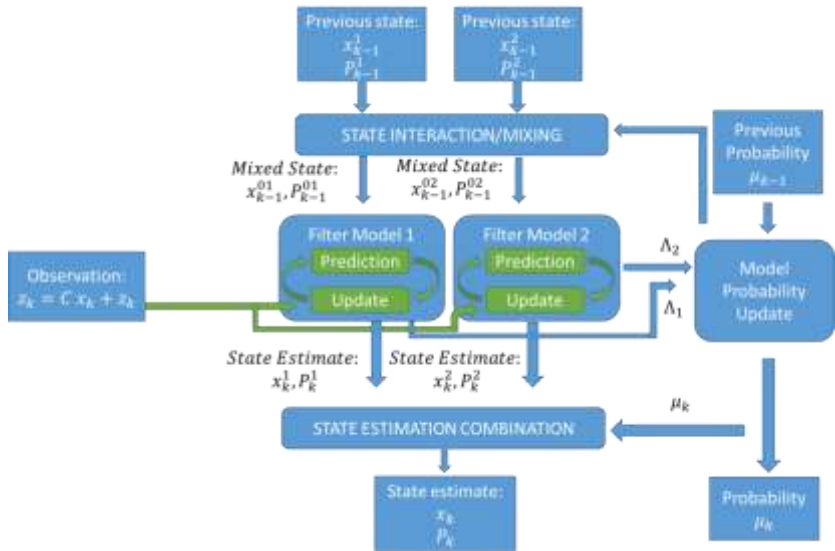


Ilustración 20. Diagrama de flujo del filtro IMM

Uno de los problemas de depender solo de la verosimilitud o “likelihood” para determinar las probabilidades de cada modo se da en escenarios de alta incertidumbre, ya que cuando la covarianza de la innovación es muy grande, las “likelihoods” de cada modo no variarán mucho entre ellas, y por tanto la convergencia del modo no está garantizada. Por esta razón un método adaptativo es necesario para ajustar las probabilidades de transición. Este método define un modo primario o fuerte y otro secundario o débil basándose en la distancia Mahalanobis ( $d_M$ ) para facilitar así la transición al modo fuerte, y por tanto mejorando el comportamiento durante los periodos transitorios. La función distancia utilizada es:

$$d_M = \sqrt{(\hat{z}_j(k|k-1) - z_j(k))^T \cdot S^{-1} \cdot (\hat{z}_j(k|k-1) - z_j(k))}$$

## 5.1 Sistemas estocásticos híbridos

Los sistemas estocásticos híbridos o SHS (Stochastic Hybrid Systems) permiten modelar las interacciones entre una dinámica continua, una dinámica discreta y la incertidumbre probabilística.

Generalmente se componen de un conjunto de variables de estado continuas que evolucionan de acuerdo a una ecuación diferencial estocástica o SDE (“Stochastic Differential Equation”); generalmente esa ecuación, que es continua, se suele discretizar, por lo que las variables de estado serán valores continuos discretizados (que no es lo mismo que una variable discreta).

Por otra parte, también se componen por un conjunto de variables discretas, denominada modo, que evolucionan según una cadena de Markov, y que tienen influencia en la dinámica del sistema, es decir, en las variables continuas.

De igual forma que si el ruido del proceso no era blanco, el proceso no se podía definir como un proceso de Markov y por tanto no podíamos aplicar el filtro de Kalman, y teníamos que recurrir al filtro aumentado, si el proceso no solo depende del vector de estado anterior, el proceso tampoco se puede definir como un proceso de Markov, por lo que tenemos que redefinir el vector de estado del sistema. En este caso debemos añadir el modo (variable discreta) al vector de estado.

Un ejemplo de un sistema estocástico híbrido se da en el siguiente caso de uso: si un coche autónomo está siguiendo el movimiento de un peatón despistado que cruza un semáforo en rojo, la dinámica del peatón será distinta si el peatón sigue cruzando la carretera despistado o de si se percata que hay un coche a punto de atropellarle, en el segundo caso, el peatón entrará en confusión y se detendrá o retrocederá para volver a la acera, por lo que su dinámica será diferente a la de movimiento rectilíneo uniforme. En este caso la variable discreta o modo es si el peatón está despistado o no.

Otro claro ejemplo es la trayectoria de una aeronave, cuya dinámica se modela a través de las ecuaciones de la trayectoria que resuelven la posición y la velocidad (variables continuas del sistema), y una variable discreta, denominada modo, que define el modo de operación de la aeronave (e.g. crucero, aterrizaje, o viraje), y el modo de operación de la aeronave cambia las ecuaciones de su dinámica.

Hay tres aspectos a tratar en los sistemas estocásticos híbridos, por una parte el análisis probabilístico de las transiciones discretas, y por otra el análisis estocástico de la evolución de las variables continuas, y por último las interacciones entre ambas dinámicas [36].

## 5.2 Cadenas de Markov

Para modelar las transiciones discretas se utilizan las cadenas de Markov [37], es decir, se asume que el proceso que rige el comportamiento del modo del sistema estocástico híbrido es un proceso de Markov.

Los procesos estocásticos que cumplen la propiedad de Markov se pueden dividir según el espacio de estados en el que trabajen (espacio continuo o discreto numerable), y según el tiempo en que se ejecutan continuo o discreto. En este caso ya que se ejecuta en tiempo discreto y sobre un conjunto de estados discreto numerable (modos), se puede modelar con una cadena de Markov DTMC (“Discrete Time Markov Chain”).

Una cadena de Markov es un proceso estocástico, donde la probabilidad de que ocurra un evento depende solamente del evento inmediatamente anterior, a esta propiedad principal que tienen estos procesos de falta de memoria se la conoce como la propiedad de Markov. Por tanto, toda cadena de Markov debe cumplir dos requisitos:

- Propiedad de Markov: falta de memoria

- Proceso estocástico: es decir, aleatorio  
Por otro lado, se pueden clasificar según las siguientes propiedades:
- Homogeneidad: Si las probabilidades de transición de un estado a otro no varían con el tiempo se dice que la cadena de Markov es homogénea, en caso contrario es no homogénea.
- Periodicidad: una cadena de Markov es aperiódica si todos sus estados son aperiódicos, y un estado es aperiódico si estando en ese estado, se puede volver a él en cualquier momento. Si el estado es periódico solo es posible volver a él en un número de etapas que sea múltiplo de un número entero (denominado periodo) mayor que uno.
- Irreducibilidad: Se dice que una cadena de Markov es irreducible si desde cualquier estado  $S$  se puede acceder a cualquier otro, es decir, todos sus estados se comunican entre sí, para ello la cadena debe ser regular.
- Regular: la cadena es regular si la  $n$ -ésima potencia de la matriz de transición, tiene sus elementos positivos distintos de cero.
- Recurrencia: una cadena es recurrente si todos sus estados son recurrentes, y un estado es recurrente si la probabilidad de regresar a él es la unidad, si es menor que la unidad el estado es transitorio.
- Absorbente: una cadena es absorbente si presenta un estado absorbente  $j$  ( $p_{jj} = 1$ ), es decir un estado del que no se puede salir; y de cualquier estado no absorbente se puede acceder al estado absorbente.

Para el modelado del comportamiento de conmutación (“switching logic”) del modo de nuestro sistema, utilizaremos en concreto una cadena de Markov con probabilidades de transición conocidas, invariantes en el tiempo, e independientes del vector de estado básico  $x(k)$ , por lo que estamos ante una cadena de Markov homogénea, en la que el modo actual depende únicamente del modo anterior. Además, esta cadena deberá ser no absorbente, aperiódica, regular e irreducible. Por tanto, la cadena de Markov será ergódica.

N.B. Algunos algoritmos utilizan cadenas de Markov no homogéneas, donde las probabilidades de transición varían en el tiempo, según el tiempo de permanencia en el modo (i.e. “Mean sojourn time”).

N.B. la propiedad de ergodicidad es una característica de algunos procesos estocásticos, donde los estadísticos de conjunto son iguales a los de tiempo; es decir, para este caso concreto el promedio de muchas aeronaves con una cadena de Markov ergódica sería igual al promedio de una sola aeronave a lo largo del tiempo con esa misma matriz ergódica.

La cadena de Markov se puede representar con la matriz de transición, que contiene en sus elementos las probabilidades de transición. La matriz de transición es una matriz cuadrada de tamaño igual a el número de estados, en nuestro caso el sistema tendrá dos estados (blanco maniobrando y sin maniobrar).

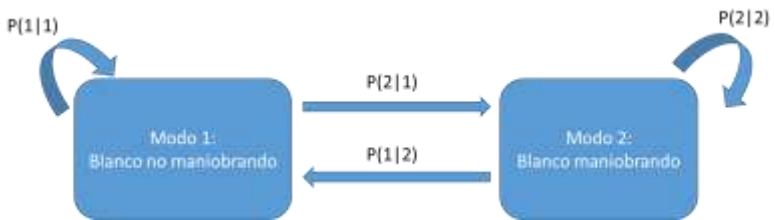


Ilustración 21. Cadena de Markov para el modo de una aeronave.

$$\begin{pmatrix} P_{k+1}(1) \\ P_{k+1}(2) \end{pmatrix} = \begin{pmatrix} P(1|1) & P(1|2) \\ P(2|1) & P(2|2) \end{pmatrix} \cdot \begin{pmatrix} P_k(1) \\ P_k(2) \end{pmatrix}$$

$$\begin{pmatrix} P_{k+1}(1) \\ P_{k+1}(2) \end{pmatrix} = \begin{pmatrix} p^{11} & p^{21} \\ p^{12} & p^{22} \end{pmatrix} \cdot \begin{pmatrix} P_k(1) \\ P_k(2) \end{pmatrix}$$

Donde  $P(j|i) = P^{ij}$  es la probabilidad de pasar del estado  $i$  al estado  $j$ . Por tanto las columnas de la matriz de transición deben sumar la unidad.

$$\sum_{j=1}^{j=n} P(j|1) = 1$$

Para el diseño de nuestro algoritmo se utilizarán las siguientes probabilidades de transición:

$$\begin{pmatrix} P_{k+1}(\text{no maniobrar}) \\ P_{k+1}(\text{maniobrar}) \end{pmatrix} = \begin{pmatrix} 0.85 & 0.2 \\ 0.15 & 0.8 \end{pmatrix} \cdot \begin{pmatrix} P_k(\text{no maniobrar}) \\ P_k(\text{maniobrar}) \end{pmatrix}$$

N.B. las probabilidades de transición son parámetros de diseño del estimador, que se deben seleccionar durante la fase de diseño.

### Ecuación de Chapman-Kolmogorov

Es una ecuación que se utiliza para determinar la distribución de probabilidad conjunta de un conjunto de coordenadas de un proceso estocástico, que aplicado a cadenas de Markov en el transcurso del tiempo se obtiene el siguiente corolario:

$$p^{ij}(n) = (p^{ij})^n$$

Donde  $p^{ij}(n)$  es la probabilidad de pasar del estado  $i$  al  $j$  en  $n$  pasos.

N.B. esta ley solo se puede expresar de forma matricial si la cadena de Markov es discreta (espacio de estados discreto) y homogénea.

Por lo que de esta forma podemos ver como convergería la cadena de Markov propuesta anteriormente.

$$\begin{pmatrix} 0.85 & 0.2 \\ 0.15 & 0.8 \end{pmatrix}^{\infty} = \begin{pmatrix} 0.571 & 0.571 \\ 0.429 & 0.429 \end{pmatrix}$$

Es importante analizar que todas las probabilidades de transición se mantienen positivas no nulas, para garantizar que la cadena es no absorbente, regular y aperiódica.

## 5.3 Modelos

Como vimos en el capítulo anterior el modelo de la dinámica de las aeronaves usado por ACAS X es un movimiento lineal de velocidad constante. Que será el mismo que utilizaremos para cuando el blanco no está maniobrando.

### Modelo 1: Blanco no maniobrando

Modelo de velocidad constante o NCV (“Nearly Constant Velocity”):

Las ecuaciones de un objeto con movimiento rectilíneo uniforme son:

$$x = x_0 + v_x \cdot t$$

$$y = y_0 + v_y \cdot t$$

$$v_x = v_{x0}$$

$$v_y = v_{y0}$$

Que al discretizarlas se pueden expresar de forma matricial de la siguiente manera:

$$\begin{pmatrix} x \\ v_x \\ y \\ v_y \end{pmatrix}_{k+1} = \begin{pmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ v_x \\ y \\ v_y \end{pmatrix}_k + \begin{pmatrix} \frac{1}{2}T^2 & 0 \\ T & 0 \\ 0 & \frac{1}{2}T^2 \\ 0 & T \end{pmatrix} v(k)$$

Donde T es el tiempo de muestreo.

### Modelo 2: Blanco maniobrando

Modelo de viraje coordinado o NCT (“Nearly coordinated turn”):

Cuando una aeronave realiza un viraje puede resbalar (“slipping”), derrapar (“skidding”), o hacerlo de forma coordinada. Todas las aeronaves intentan hacer virajes coordinados, ya que es la forma más eficiente de volar.

Basándonos en la cinemática del blanco las ecuaciones de un viraje coordinado son:

$$\dot{x}(t) = V(t) \cdot \cos(\phi(t))$$

$$\dot{y}(t) = V(t) \cdot \sin(\phi(t))$$

$$\dot{V}(t) = a_t(t)$$

$$\phi(t) = \frac{a_n(t)}{V(t)}$$

Donde:

(x, y) son las coordenadas cartesianas.

$V(t)$  es la velocidad.

$a_t(t)$  es la aceleración tangencial.

$a_n(t)$  es la aceleración normal o centrípeta.

$\phi(t)$ : es el rumbo.

Para el caso de viraje coordinado, la aceleración tangencial es nula (velocidad constante), la aceleración normal es no nula (movimiento curvilíneo) y la velocidad angular es constante:

$$\dot{\phi}(t) = \omega$$

Por otra parte, las componentes x e y de la velocidad son:

$$v_x = |V(t)| \cdot \cos(\phi)$$

$$v_y = |V(t)| \cdot \sin(\phi)$$

Derivando, obtenemos las dos componentes cartesianas de la aceleración:

$$a_x = -|V(t)| \cdot \omega \cdot \sin(\phi)$$

$$a_y = |V(t)| \cdot \omega \cdot \cos(\phi)$$

Si discretizamos para un tiempo de muestreo T:

$$x(k+1) = x(k) + \int_k^{k+T} v_x(\tau) \cdot d\tau$$

$$x(k+1) = x(k) + \int_k^{k+T} |V| \cdot \cos(\phi + \omega \cdot \tau) \cdot d\tau$$

$$x(k+1) = x(k) + |V|T \left[ \frac{\sin(\omega T)}{\omega T} \cos(\phi) - \frac{1 - \cos(\omega T)}{\omega T} \sin(\phi) \right]$$

$$x(k+1) = x(k) + T \left[ \frac{\sin(\omega T)}{\omega T} v_x - \frac{1 - \cos(\omega T)}{\omega T} v_y \right]$$

De la misma forma para la coordenada y:

$$y(k+1) = y(k) + \int_k^{k+T} v_y(\tau) \cdot d\tau$$

$$y(k+1) = y(k) + \int_k^{k+T} |V| \cdot \sin(\phi + \omega \cdot \tau) \cdot d\tau$$

$$y(k+1) = y(k) + |V|T \left[ \frac{\sin(\omega T)}{\omega T} \sin(\phi) + \frac{1 - \cos(\omega T)}{\omega T} \cos(\phi) \right]$$

$$y(k+1) = y(k) + T \left[ \frac{\sin(\omega T)}{\omega T} v_y + \frac{1 - \cos(\omega T)}{\omega T} v_x \right]$$

De la misma manera podemos discretizar las ecuaciones de la velocidad, y expresarlas en función de las otras variables de estado:

$$v_x(k+1) = |V| \cdot \cos(\phi + \omega T)$$

$$v_y(k+1) = |V| \cdot \sin(\phi + \omega T)$$

Haciendo uso de las identidades notables:

$$\cos(a+b) = \cos(a) \cdot \cos(b) - \sin(a) \cdot \sin(b)$$

$$\sin(a+b) = \sin(a) \cdot \cos(b) + \cos(a) \cdot \sin(b)$$

Obtenemos:

$$v_x(k+1) = |V| \cdot [\cos(\phi) \cdot \cos(\omega T) - \sin(\phi) \cdot \sin(\omega T)]$$

$$\mathbf{v}_x(\mathbf{k}+1) = \mathbf{v}_x(\mathbf{k}) \cdot \cos(\omega T) - \mathbf{v}_y(\mathbf{k}) \cdot \sin(\omega T)$$

$$v_y(k+1) = |V| \cdot [\sin(\phi) \cdot \cos(\omega T) + \cos(\phi) \cdot \sin(\omega T)]$$

$$\mathbf{v}_y(\mathbf{k}+1) = \mathbf{v}_y(\mathbf{k}) \cdot \cos(\omega T) + \mathbf{v}_x(\mathbf{k}) \cdot \sin(\omega T)$$

Finalmente, solo nos quedará expresar las ecuaciones de las variables de estado en forma matricial:

$$\begin{pmatrix} x \\ v_x \\ y \\ v_y \end{pmatrix}_{k+1} = \begin{pmatrix} 1 & \frac{\sin(\omega T)}{\omega} & 0 & -\frac{1 - \cos(\omega T)}{\omega} \\ 0 & \cos(\omega T) & 0 & -\sin(\omega T) \\ 0 & \frac{1 - \cos(\omega T)}{\omega} & 1 & \frac{\sin(\omega T)}{\omega} \\ 0 & \sin(\omega T) & 0 & \cos(\omega T) \end{pmatrix} \cdot \begin{pmatrix} x \\ v_x \\ y \\ v_y \end{pmatrix}_k$$

Como la velocidad angular del viraje, aunque sea constante, es desconocida, el vector de estado debe ser aumentado con esta variable.

$$\begin{pmatrix} x \\ v_x \\ y \\ v_y \\ \omega \end{pmatrix}_{k+1} = \begin{pmatrix} 1 & \frac{\sin(\omega T)}{\omega} & 0 & -\frac{1 - \cos(\omega T)}{\omega} \\ 0 & \cos(\omega T) & 0 & -\sin(\omega T) \\ 0 & \frac{1 - \cos(\omega T)}{\omega} & 1 & \frac{\sin(\omega T)}{\omega} \\ 0 & \sin(\omega T) & 0 & \cos(\omega T) \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ v_x \\ y \\ v_y \\ \omega \end{pmatrix}_k + \begin{pmatrix} \frac{1}{2}T^2 & 0 & 0 \\ T & 0 & 0 \\ 0 & \frac{1}{2}T^2 & 0 \\ 0 & T & 0 \\ 0 & 0 & T \end{pmatrix} v_k$$

Como se puede apreciar este modelo no es lineal, por lo que hay dos opciones, linealizarlo y utilizar un filtro EKF o continuar con las ecuaciones no lineales y utilizar un filtro UKF.

Para estimar la velocidad angular se utilizarán las siguientes formulas:

$$a_n = \frac{v^2}{r}$$

$$a_n = a \cdot \sin(\theta)$$

$$\omega = \frac{v}{r} = \frac{v^2}{v \cdot r} = \frac{a_n}{v} = \frac{a \cdot \sin(\theta)}{v} = \frac{v \cdot a \cdot \sin(\theta)}{v^2} = \frac{|v \times a|}{v^2}$$

$$\omega = \frac{|v \times a|}{v^2}$$

Es importante destacar que si la velocidad angular estimada es nula, la matriz de estado será singular.

### Ecuación de medida

Respecto a la ecuación de medida para ambos casos (blanco maniobrando o no), será la ecuación radar, ya que una aeronave recibe la información sobre la aeronave intrusa, a través del radar secundario (distancia oblicua, ángulo de azimut y altura)

$$z(k) = H \cdot x(k) + w(k)$$

$$r = \sqrt{x^2 + y^2}$$

$$\theta = \text{atan}\left(\frac{y}{x}\right)$$

Al ser las ecuaciones no lineales, no se pueden expresar como una matriz, aunque si las linealizamos se puede expresar en formato matricial a través de la matriz Jacobina:

$$\begin{pmatrix} r \\ \xi \end{pmatrix} = \begin{bmatrix} \frac{x}{\sqrt{x^2 + y^2}} & 0 & \frac{y}{\sqrt{x^2 + y^2}} & 0 \\ -\frac{y}{x^2 + y^2} & 0 & \frac{x}{x^2 + y^2} & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ v_x \\ y \\ v_y \end{bmatrix}$$

Para la implementación del filtro, se utilizará la técnica UKF, para mejorar la exactitud durante la propagación de la distribución a través de la ecuación de medida, y evitar evaluar la matriz Jacobiana para cada instante, lo que empeora el tiempo de ejecución.

## 6. MÉTRICAS DE COMPARACIÓN

### 6.1 Selección de encuentros

Se define encuentro como una aproximación entre dos aeronaves cooperativas (equipadas con TCAS/ACAS), por lo que básicamente es la geometría de las trayectorias junto con un perfil de velocidades, es decir cada punto de ruta tiene asociado un tiempo (trayectoria 4D). Para poder determinar el desempeño del filtro IMM es importante que entre las geometrías haya virajes de aeronaves, y los encuentros sean representativos del espacio aéreo, como la plataforma CAFE (Collision Avoidance Fast-time Evaluator) de Eurocontrol o en [38], y [39].

Durante los resultados de esta tesis se seleccionarán 4 encuentros, dónde se variará las trayectorias horizontales, ya que como se verá más adelante los perfiles verticales de las aeronaves no influyen sobre el error de estimación horizontal, ya que los algoritmos de estimación del movimiento horizontal y vertical están desacoplados.

Cabe destacar, que la configuración de la simulación de Monte Carlo no forma parte del encuentro, y es lo que se denomina el escenario, es decir, el escenario son los parámetros de los modelos de ruido vistos en el capítulo 4.2,

y por tanto un encuentro no se puede simular en el simulador de Monte Carlo, es el encuentro más el escenario, lo que denominamos el encuentro-escenario la entrada del simulador, por eso en la publicación [6], se habla de encuentros-escenarios.

## 6.2 Métricas

Las principales métricas que se detallan a continuación tratan de caracterizar el error de estimación. El error de medición, que ya ha sido modelado en el capítulo 4 para las simulaciones de Monte Carlo, indica la diferencia entre la trayectoria verdadera y la trayectoria medida por el radar secundario de la aeronave. El proceso de estimación de trata de reducir el error de medición, por lo que es tras el proceso de estimación donde está el valor añadido de esta tesis, y en consecuencia se estudiará el error de estimación.

### 6.2.1 Tipos de errores

El error de estimación se define como la diferencia entre el valor verdadero y el estimado:

$$\tilde{x}(k) = x(k) - \hat{x}(k)$$

Por lo que para cada punto estimado de la trayectoria (k) tenemos un error de estimación diferente. Y es interesante analizar tanto el valor del error como su tendencia a lo largo de la trayectoria.

El error es una variable aleatoria y por lo que si aplicamos el operador esperanza y dispersión obtendremos el error medio y su varianza:

$$E[\tilde{x}] = \frac{1}{n} \sum_{k=1}^n x(k) - \hat{x}(k)$$

$$D[\tilde{x}] = \text{var}(\tilde{x}) = \frac{1}{n} \sum_{k=1}^n (\tilde{x}(k) - E[\tilde{x}])^2$$

El problema de esta definición es que obtenemos errores positivos y negativos de forma aleatoria, por lo que se podría presenciar un gran error cuya esperanza (media) fuera nula, lo cual no nos da una buena indicación del error presente en la trayectoria estimada; en consecuencia, generalmente no importa la dirección del error, si no su valor absoluto, y con tal propósito se define el error absoluto:

$$\tilde{x}(k) = |x(k) - \hat{x}(k)|$$

El problema del error absoluto es que aplica la función valor absoluto, que no es una función suave (o infinitamente diferenciable), y no queremos tener una variable no derivable, ya que hay algunas operaciones en las que necesitamos diferenciar el error, como por ejemplo para minimizar el error, o para obtener la velocidad de convergencia.

Por tanto, la métrica generalmente aceptada es el error cuadrático medio (MSE: Mean Squared Error) o la raíz del error cuadrático medio (RMSE: Root Mean Squared Error), que se define como:

$$\tilde{x}_{MSE} = (x - \hat{x})^2$$

$$\tilde{x}_{RMS} = \sqrt{(x - \hat{x})^2}$$

Los tres tipos de errores vistos anteriormente son absolutos, esto quiere decir que no se expresan en relación a otra cantidad. Por lo cual los errores anteriores deben ir acompañados de unidades.

Los errores relativos se determinan como el error absoluto dividido por el valor verdadero, y por tanto no tiene unidades, aunque se puede expresar en tanto por ciento. Un tipo de error relativo muy usado para medir el desempeño de un algoritmo de estimación es el error de estimación

cuadrático normalizado (NEES: Normalized estimation error squared), que es muy útil especialmente si hay diferentes variables de estado con diferentes órdenes de magnitud

$$\epsilon_x = \tilde{x}' \cdot P^{-1} \cdot \tilde{x}$$

El error normalizado NEES sigue (bajo la hipótesis de que el error de estimación sigue una distribución gaussiana) una distribución chi cuadrado de  $n_x$  grados de libertad, siendo  $n_x$  el número de variables de estado, por lo que el test estadístico a utilizar sobre el NEES es el  $\chi^2$ .

$$\epsilon_x \sim \chi^2$$

Este tipo de errores que consideran el error del estado en un espacio de estados multidimensional (4 variables de estado), es necesario cuando se estima utilizando una representación de espacio de estados, como en [40], dónde se utiliza la distancia “cut off”.

Sobre los errores medios cuadráticos y el error normalizado estudiaremos sus distribuciones de probabilidad, los parámetros más característicos serán:

- Media: es una medida de tendencia central que se utiliza generalmente para distribuciones normales.
- Mediana: es una medida de tendencia central más representativa que la media cuando la distribución no es simétrica (distribución sesgada), como en el caso del NEES que sigue una distribución  $\chi^2$ .
- Desviación estándar: es una medida de la dispersión de la muestra.
- Percentil 75: es una medida que indica que por debajo de ese valor recae el 75% de la muestra.
- Percentil 25: es una medida que indica que por debajo de ese valor recae el 25% de la muestra.

Estos datos se pueden representar en un único gráfico denominado “whisker-box”, que consiste en una caja cuyo límite superior es el percentil 75, el límite inferior el percentil 25, y una línea central que indica la mediana, y además

contiene dos mechas, la superior indica el valor máximo, y la inferior el valor mínimo:

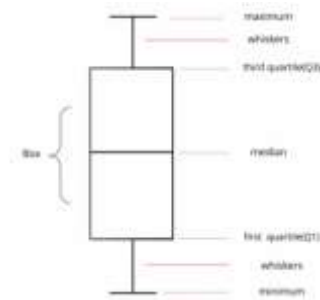


Ilustración 22. Gráfico "Whisker Box"

Además, también se mostrará las funciones de densidad de probabilidad obtenidas durante las simulaciones como en [41].

A continuación, se detallan los test de hipótesis elegidos como relevantes para comparar el rendimiento de dos algoritmos de estimación. Los test de hipótesis tienen una gran importancia a la hora de comparar algoritmos ya que nos dirán si las diferencias en el desempeño son estadísticamente significativas. En la literatura científica actual muchos artículos que intentan proponer nuevos algoritmos de estimación y los comparan con el estado del arte carecen de análisis de los test de hipótesis como en [7], [42], y [43].

Estos tests de hipótesis se aplicarán a cada uno de los errores a estudiar:

- Error de posición en x
- Error de posición en y
- Error de velocidad en x
- Error de velocidad en y
- Error normalizado NEES

## 6.2.2 Test de hipótesis de las regiones de probabilidad

Las regiones de probabilidad o regiones de aceptación se utilizan para comprobar la consistencia del filtro, si los test estadísticos caen dentro de la región de probabilidad se puede aceptar la hipótesis de que el filtro es consistente, en concreto que el error normalizado esté dentro de la región de probabilidad (sección 3.4.6 [44]).

La consistencia de un estimador sólo se puede analizar mediante el método de Monte Carlo, e implica que los errores de estimación reales están alineados o son consistentes con el error de estimación “predicho o estimado” y el teórico.

Cuando se realizan simulaciones de Monte Carlo, el error normalizado NEES será:

$$\bar{\varepsilon}(k) = \frac{1}{N} \sum_{i=1}^N \varepsilon^i(k)$$

N.B. el error se promedia para cada paso de la simulación (k).

En este caso  $N \cdot \bar{\varepsilon}(k)$  sigue una distribución chi cuadrado de  $N \cdot n_x$  grados de libertad, siendo N el número de ejecuciones de Mote Carlo y  $n_x$  el número de variables de estado.

Los errores de estimación serán consistentes con las covarianzas calculadas por el filtro, si pasa el test de hipótesis, en este caso el test  $\chi^2$ . Para ello el error deberá estar dentro de la región de aceptación ( [21] capítulo 3.7).

$$\bar{\varepsilon}(k) \in [r_1, r_2]$$

La región de aceptación se puede determinar con:

1. Región de probabilidad bilateral. Para el caso de la región de concentración de probabilidad del 95%, sería:

$$[\chi_{Nn_x}^2(0.025), \chi_{Nn_x}^2(0.975)]$$

2. Región de probabilidad unilateral. Para el caso de la región de concentración de probabilidad del 95% sería:

$$\chi_{Nn_x}^2(0.95)$$

### 6.2.3 Test de hipótesis de la diferencia de error

Aunque un test de hipótesis sencillo sería la comparación de medias del error de n ejecuciones de Monte Carlo, éste resultaría incompleto, ya que la incertidumbre asociada a cada media no se considera.

El test estadístico para comparar algoritmos de estimación

- Hipótesis nula (H0): algoritmo 1 no es mejor que algoritmo 2
- Hipótesis alternativa (HA): algoritmo 1 es mejor que algoritmo 2; en términos estadísticos, existe una diferencia significativa en la diferencia de las medias verdaderas (de la población) de cada algoritmo.

$$P\{\text{aceptar HA} | H0 \text{ es verdadera}\} = \alpha$$

Siendo  $\alpha$  el nivel de significancia de la hipótesis nula, que define el nivel de confianza:

$$\text{nivel de confianza} = 1 - \alpha$$

La función de coste a optimizar que será indicativa del desempeño de cada algoritmo que generalmente se usa es el error cuadrático

$$C = (\hat{x}_k - x_k)^2$$

Y el valor esperado de la función de coste, el error cuadrático medio, que es en la herramienta que nos basaremos para medir el desempeño.

$$J = E[C]$$

Sin embargo, al no ser cada ejecución de Monte Carlo de cada algoritmo independiente, ya que usan el mismo conjunto de números aleatorios, la función de coste será la diferencia del desempeño de cada ejecución de Monte carlo.

$$\Delta_i = C_i^{\{2\}} - C_i^{\{1\}}$$

La hipótesis alternativa se acepta si:

$$\mu = \frac{\bar{\Delta}}{\sigma_{\bar{\Delta}}} > \mu_0$$

Donde  $\bar{\Delta}$  es la media muestral de las diferencias y  $\sigma_{\bar{\Delta}}$  es la desviación típica de las diferencias del error.

$$\bar{\Delta} = \frac{1}{N} \sum_{i=1}^N \Delta_i$$

$$\sigma_{\bar{\Delta}} = \sqrt{\frac{1}{N^2} \sum_{i=1}^N (\Delta_i - \bar{\Delta})^2}$$

El umbral  $\mu_0$  se basa en el valor de una distribución normal que tiene una probabilidad de cola superior (cola derecha de la distribución) igual al nivel de significancia.

**Tabla 1. método de cálculo del test de hipótesis**

Ejecución de MC	Algoritmo 1	Algoritmo 2	Diferencia
Ejecución 1	$C_1^{\{1\}}$	$C_1^{\{2\}}$	$\Delta_1 = C_1^{\{2\}} - C_1^{\{1\}}$
Ejecución 2	$C_2^{\{1\}}$	$C_2^{\{2\}}$	$\Delta_2 = C_2^{\{2\}} - C_2^{\{1\}}$
Ejecución 3	$C_3^{\{1\}}$	$C_3^{\{2\}}$	$\Delta_3 = C_3^{\{2\}} - C_3^{\{1\}}$

...	...	...	...
<b>Ejecución N</b>	$C_N^{\{1\}}$	$C_N^{\{2\}}$	$\Delta_N = C_N^{\{2\}} - C_N^{\{1\}}$
<b>Media</b>	$\bar{C}^{\{1\}}$	$\bar{C}^{\{2\}}$	$\bar{\Delta} = \frac{1}{N} \sum_{i=1}^N \Delta_i$

Finalmente, si el ratio  $\bar{\Delta}/\sigma_{\bar{\Delta}}$  supera un umbral, entonces la hipótesis alternativa se acepta, es decir, existe una diferencia significativa entre las medias verdaderas del error de cada algoritmo. El umbral dependerá del nivel de confianza, para un nivel de confianza del 95%:

$$\frac{\bar{\Delta}}{\sigma_{\bar{\Delta}}} > N(0,95) = 1.65$$

N.B. Hemos definido la función de coste como la diferencia del error cuadrático, pero cada encuentro consta de muchos “errores cuadráticos”, puede ser la diferencia del error cuadrático en un waypoint determinado (de los muchos puntos de ruta de los que consta el encuentro), la diferencia en el error cuadrático medio de un tramo del encuentro (por ejemplo, donde se produce una maniobra que queremos analizar), o la diferencia en el error cuadrático medio del encuentro:

Diferencia del error cuadrático de un punto de la ruta k

$$C = \tilde{x}_k^{\{1\}^2} - \tilde{x}_k^{\{2\}^2}$$

Donde  $\tilde{x}_k^{\{1\}^2}$  y  $\tilde{x}_k^{\{2\}^2}$  son los errores cuadráticos en el punto k de los algoritmos 1 y 2 respectivamente.

Error cuadrático medio de un intervalo de [k, l] de la ruta:

$$C = \frac{1}{l - m + 1} \sum_{k=m}^l (\tilde{x}_k^{\{1\}^2} - \tilde{x}_k^{\{2\}^2})$$

Error cuadrático medio de todo el encuentro (que consta de n puntos de ruta):

$$C = \frac{1}{n} \sum_{k=1}^n (\tilde{x}_k^{\{1\}^2} - \tilde{x}_k^{\{2\}^2})$$

### 6.2.4 Test de hipótesis de dos muestras (2 sample t test)

Es un test de contraste de hipótesis que compara la diferencia de medias poblacionales correspondientes a dos muestras, el resultado del test es positivo si la diferencia entre las muestras poblacionales es estadísticamente significativa, y por tanto se rechaza la hipótesis nula.

**Hipótesis nula:** las medias poblacionales correspondientes a cada muestra son iguales.

**Hipótesis alternativa:** las medias poblacionales correspondientes a cada muestra son distintas.

El estadístico de esta prueba es la diferencia de medias normalizada con las desviaciones típicas y el tamaño de las muestras.

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\frac{s_1^2 + s_2^2}{2} \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

Donde  $s_1$  y  $s_2$  son las desviaciones típicas de las muestras 1 y 2 respectivamente, y  $n_1$  y  $n_2$  son el tamaño de las muestras.

De forma gráfica también se representarán los intervalos de confianza de la media (se calculan como se detalla en la sección 9.4 de [45]) de cada una de las muestras, para visualizar la diferencia en un gráfico.

### 6.2.5 Test de hipótesis Kolmogorov-Smirnov (test K-S):

Es un test de hipótesis estadístico o contraste de hipótesis que compara dos distribuciones de probabilidad unidimensionales, se puede utilizar tanto el test K-S de una muestra que compara una muestra contra una distribución de probabilidad de referencia; o el test K-S de dos muestras, que compara dos muestras, como se explica en la sección 5.1.1 de [46]; para el test K-S de dos muestras las hipótesis son las siguientes:

**Hipótesis nula:** Las dos muestras pertenecen a la misma población, es decir, provienen de la misma distribución de probabilidad.

**Hipótesis alternativa:** Las dos muestras pertenecen a poblaciones distintas, es decir, provienen de distintas distribuciones de probabilidad.

El estadístico de esta prueba es la distancia entre las funciones de distribución acumuladas de las dos muestras; y por tanto el test es sensible tanto a diferencias de localización de la función de probabilidad como a la forma.

$$D = \max(|F_1(x) - F_2(x)|)$$

Siendo  $F_1$  y  $F_2$  las funciones de distribución de las muestras 1 y 2 respectivamente.

### 6.2.6 Kernel Density Estimation (KDE)

Es un algoritmo no paramétrico para estimar la función de densidad de probabilidad de una variable aleatoria, y se utilizará durante las simulaciones para estimar la función de distribución de los errores de estimación.

El método tradicional sería determinar el histograma de frecuencias relativas para una anchura de los intervalos (barras del histograma) pequeña. Esto presenta algunos problemas como que se asume que un valor concreto de la variable aleatoria está en la mitad del intervalo correspondiente, mientras

que el algoritmo KDE no desplaza la posición de los valores, y simplemente posiciona una función delta de Dirac en cada realización de la variable aleatoria, y suma todas las funciones delta para estimar la distribución de probabilidad.

$$p(x) = \sum_{i=1}^N w_i \cdot k(x - x_i)$$

Donde  $w_i$  es un peso que se asigna a cada realización (valor observado de la variable aleatoria), tal que  $\sum_{i=1}^N w_i = 1$ ; y  $k(x - x_i)$  es la función kernel centrada en  $x_i$ , se pueden elegir varias funciones para la función kernel, por ejemplo, una función delta de Dirac, o función triángulo.

En el simulador desarrollado durante esta tesis  $w_i = 1/N$ , siendo N el número de simulaciones de Monte Carlo, la función kernel será una distribución normal, y  $x_i$  será el valor de la variable aleatoria en la i-ésima simulación de Monte Carlo.

## 7 RESULTADOS

En esta sección se pretende hacer simulaciones de Monte Carlo a 4 encuentros (5 si se tiene en cuenta que el encuentro número 1 se simula dos veces modificando la trayectoria vertical de la aeronave, para comprobar que la trayectoria vertical está desacoplada de la horizontal), cada encuentro se ejecutará durante 10000 ejecuciones de Monte Carlo (“Monte Carlo run”), y es importante destacar que se utilizará:

1. la misma secuencia de números aleatorios para ambos algoritmos.
2. La misma inicialización (condiciones iniciales).
3. Mismas variables dónde se modela el ruido.
4. Misma tasa de refresco de los sensores (1 segundo).

Para decidir el número de ejecuciones de Monte Carlo utilizaremos dos criterios:

1. Tiempo de la simulación: a mayor número de ejecuciones de Monte Carlo, mayor será el tiempo de simulación.

2. Exactitud de los resultados: a mayor número de ejecuciones de Monte Carlo, mayor será la exactitud de los resultados. El objetivo es obtener la mayor exactitud posible en un tiempo de simulación razonable.

Respecto al análisis del rendimiento o tiempo de ejecución de las simulaciones de Monte Carlo, se han realizado pruebas para diferente número de ejecuciones, y se ha medido el tiempo de ejecución, más tarde se ha realizado una regresión lineal que se aproxima bastante bien a los datos recogidos.



Ilustración 23. Regresión lineal del tiempo de ejecución según las ejecuciones de Monte Carlo.

Respecto a la exactitud de los resultados, como hemos publicado en [48], conviene analizar parámetros estadísticos de la población y no de la muestra, ya que la media muestral converge pronto, en torno a 500 ejecuciones de Monte Carlo, pero la exactitud de la media poblacional todavía no es suficiente, como se muestra en la siguiente tabla para el encuentro 1 y el error de estimación en la coordenada x; dónde la exactitud de la media poblacional se hace a través de su intervalo de confianza, que consta de una cota superior (UE: Upper End), y otra inferior (LE: Lower End), y la diferencia entre ambas es el margen de error (ME):

**Tabla 2. Resultados según el número de ejecuciones.**

Ejecuciones de MC	RMSE media (ft)	Cambio (%)	UE (ft)	LE (ft)	ME (ft)	Tiempo ejecución
1	680.7	-	-	-	-	44.2 s
5	525.8	-29.5	614.82	436.84	177.98	47.1 s
10	661.98	+25.9	781.24	542.72	238.52	49.8 s
50	671.95	+1.5	717.26	626.63	90.63	71 s
100	657.56	-2.1	685.38	629.75	55.63	73.2 s
500	660.3	+0.4	673.16	647.43	25.73	233.9 s
1000	666.3	+0.9	675.68	657.1	18.58	412 s
2500	660	-0.95	661.62	650.35	11.27	1102 s
5000	666.1	+0.92	670.58	662.41	8.17	2050 s
10000	664.29	-0.32	667	661.6	5.4	4112 s

Por otro lado, también conviene analizar el tiempo de ejecución de los algoritmos (“profiling”), se presentan los resultados en la siguiente tabla:

**Tabla 3. Análisis de rendimiento de los algoritmos**

	ACAS X	IMM
<b>Media</b>	0.001 s	0.0023 s
<b>Desviación típica</b>	0.0007 s	0.006 s

De forma secuencial el filtro IMM tarda algo más de tiempo en ejecutar su lógica, lo cual tiene sentido, ya que debe ejecutar los modelos de dos filtros. El lado positivo es que los dos filtros pueden ejecutarse en paralelo dada su independencia como se explica en el capítulo 6, por lo que podemos obtener rendimientos similares mediante paralelización.

## 7.1 Encuentro 1

### 7.1.1 Geometría

El encuentro número 1 se compone de dos aeronaves aproximándose en línea recta con un ángulo de 45 grados.

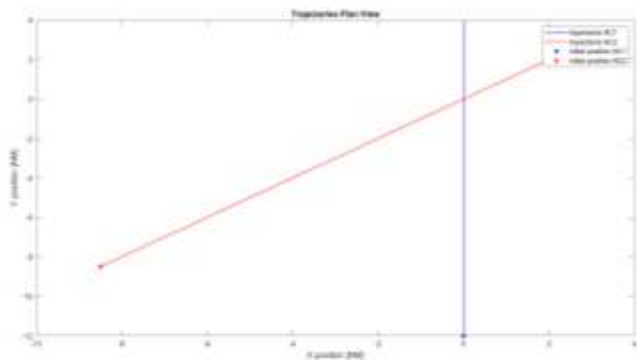


Ilustración 24. Vista horizontal del encuentro 1

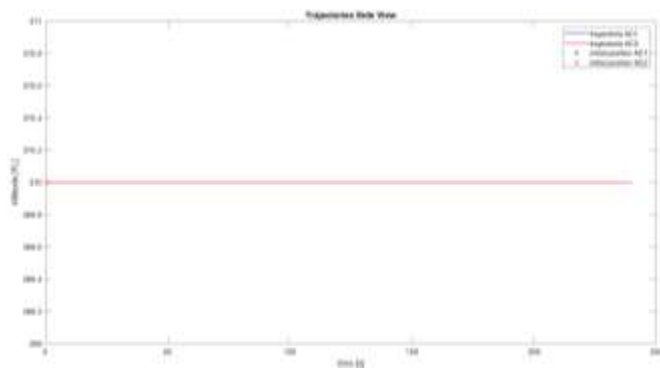


Ilustración 25. Vista de perfil del encuentro 1

Cabe destacar que los movimientos horizontal y vertical están desacoplados, por lo que si simulamos el mismo encuentro con otro perfil vertical distinto (ver siguiente figura), obtenemos los mismos resultados.

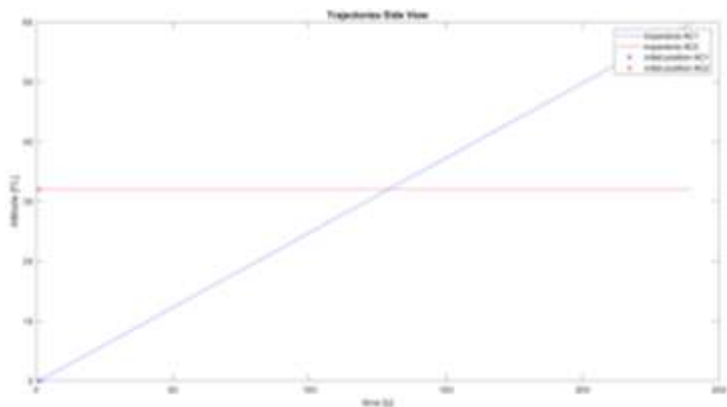


Ilustración 26. Modificación encuentro 1: vista de perfil

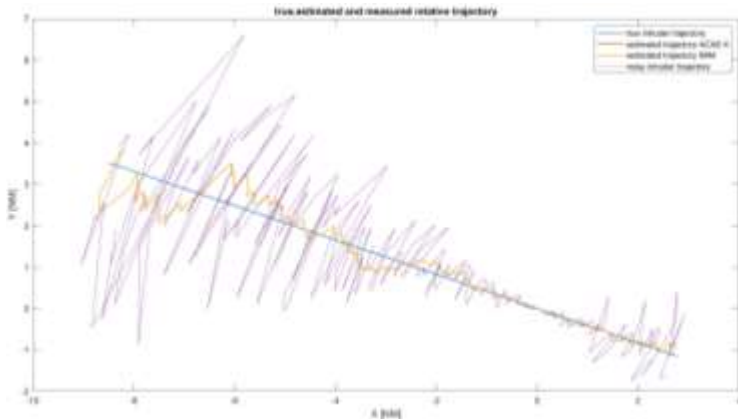


Ilustración 27. Trayectoria relativa del encuentro 1

## 7.1.2 Métricas

### *RMS X*

RMS X	ACAS X	IMM
-------	--------	-----

Media [ft]	664.29	664.19
Mediana [ft]	426.69	426.09
Desviación típica [ft]	777.45	778.03
Percentil 75 [ft]	894.09	894.14
Percentil 25 [ft]	175.00	174.36

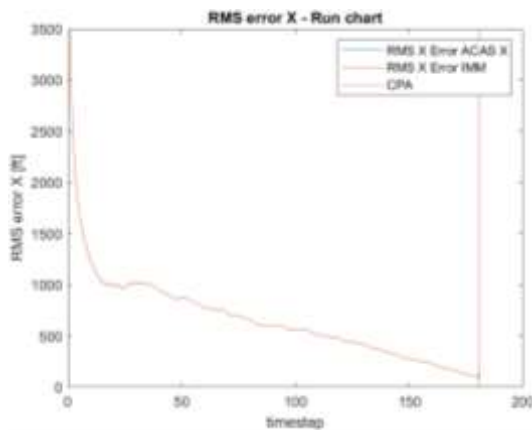


Ilustración 28. Secuencia del error RMS X

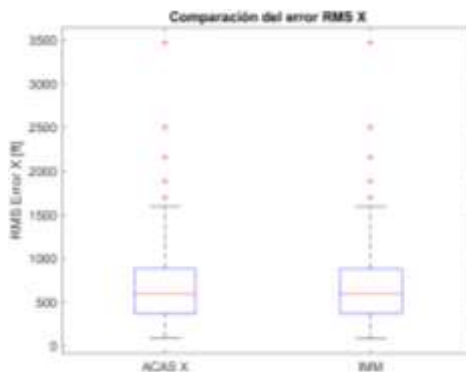


Ilustración 29. Gráfico "whisker box" del error RMS X

**Test de hipótesis “2 sample t test”:** no se rechaza la hipótesis nula (las medias de las poblaciones son iguales).

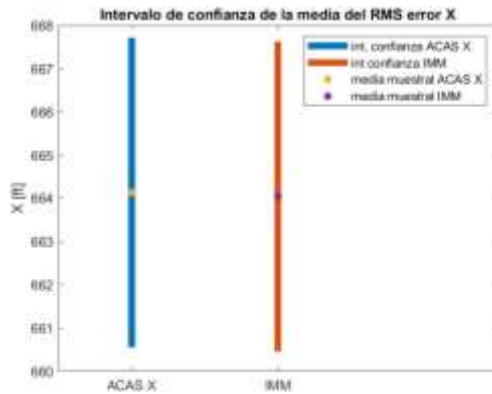


Ilustración 30. Intervalo de confianza de la media del error RMS X

**Test de hipótesis “Kolmogorov-Smirnov”:** no se rechaza la hipótesis nula (los datos provienen de la misma distribución)

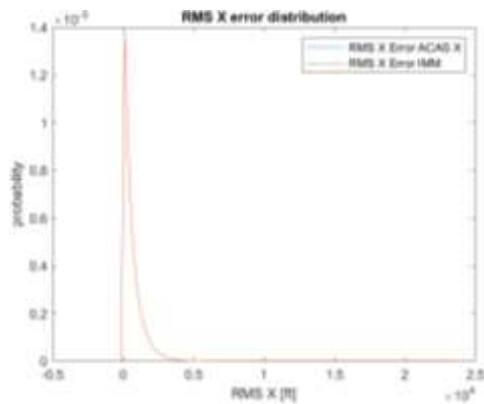


Ilustración 31. Distribución del error RMS X

**Test de hipótesis de la diferencia de error:** No se rechaza la hipótesis nula (Algoritmo IMM no es mejor que ACAS X)

## RMS Y

RMS Y	ACAS X	IMM
Media [ft]	1394.83	1394.54
Mediana [ft]	894.90	894.55
Desviación típica [ft]	1660.43	1660.60
Percentil 75 [ft]	1876.77	1876.11
Percentil 25 [ft]	345.79	345.74

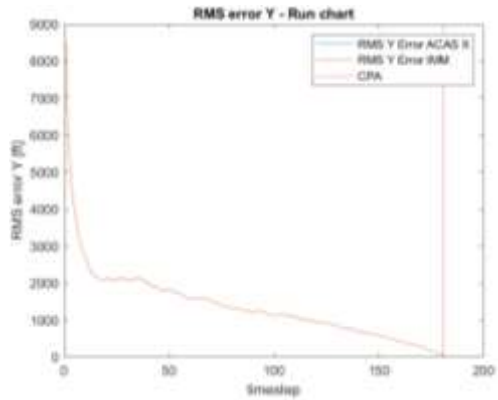


Ilustración 32. Secuencia del error RMS VX

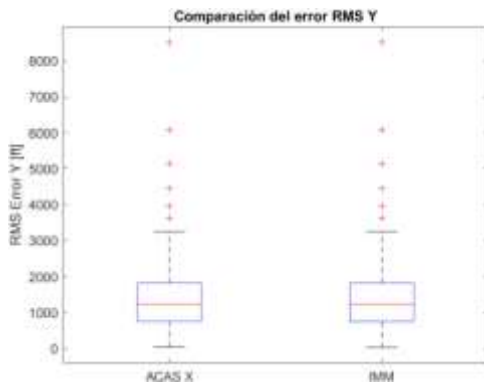


Ilustración 33. Gráfico "Whisker box" del error RMS VX

**Test de hipótesis "2 sample t test":** no se rechaza la hipótesis nula (las medias de las poblaciones son iguales).



Ilustración 34. Intervalo de confianza de la media del error RMS Y

**Test de hipótesis "Kolmogorov-Smirnov":** no se rechaza la hipótesis nula (los datos provienen de la misma distribución)

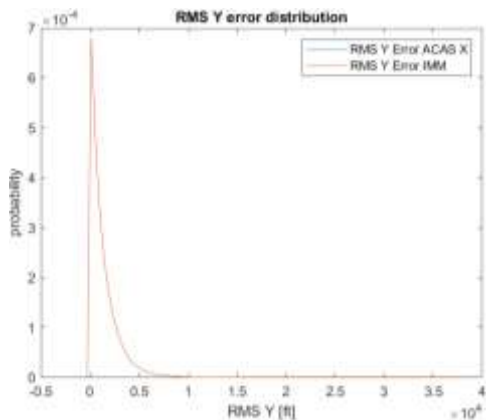


Ilustración 35. Distribución del error RMS Y

**Test de hipótesis de la diferencia de error:** No se rechaza la hipótesis nula (Algoritmo IMM no es mejor que ACAS X)

### *RMS VX*

El error cuadrático medio de la componente x de la velocidad, presenta las siguientes características:

<b>RMS VX</b>	<b>ACAS X</b>	<b>IMM</b>
Media [ft/s]	44.20	44.28
Mediana [ft/s]	26.56	26.62
Desviación típica [ft/s]	55.47	55.54
Percentil 75 [ft/s]	50.83	50.96
Percentil 25 [ft/s]	12.20	12.21

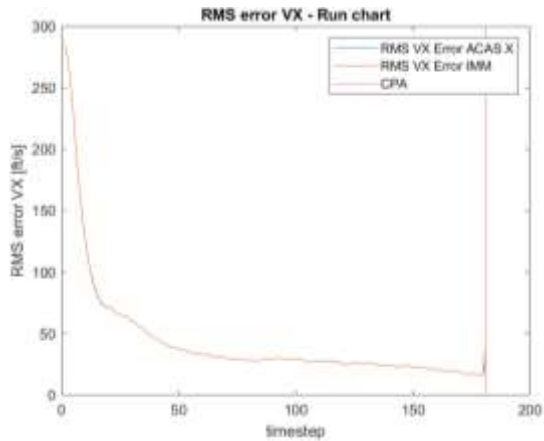


Ilustración 36. Secuencia del error RMS VX

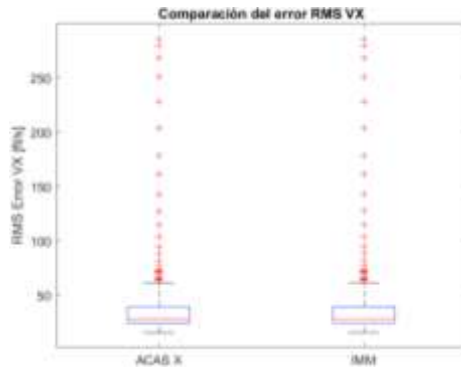


Ilustración 37. Gráfico "Whisker box" del error RMS VX

**Test de hipótesis "2 sample t test":** No se rechaza la hipótesis nula (las medias de las poblaciones son iguales), es decir las medias poblacionales de ACAS X e IMM son iguales.

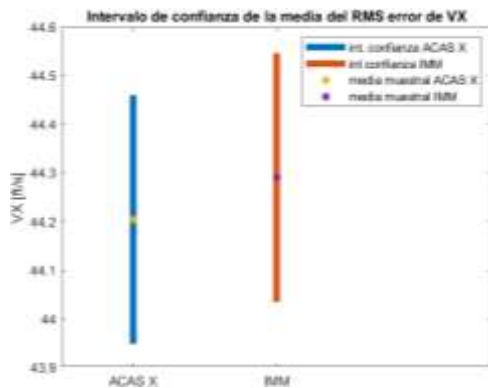


Ilustración 38. Intervalo de confianza de la media del error RMS VX

**Test de hipótesis “Kolmogorov-Smirnov”:** No se rechaza la hipótesis nula (los datos provienen de la misma distribución), por lo que las muestras de ACAS X e IMM pertenecen a la misma población.

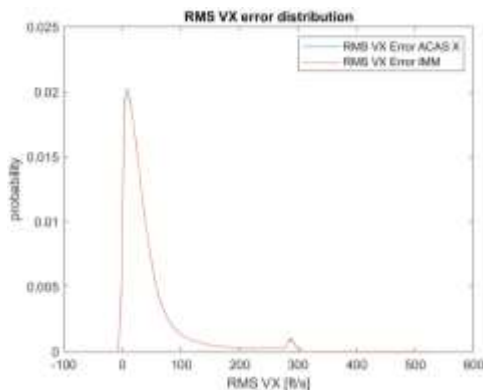


Ilustración 39. Distribución del error RMS VX

**Test de hipótesis de la diferencia de error:** No se rechaza la hipótesis nula (Algoritmo IMM no es mejor que ACAS X)

## RMS VY

El error cuadrático medio de la componente y y de la velocidad, presenta las siguientes características:

RMS VY	ACAS X	IMM
Media [ft/s]	65.14	65.14
Mediana [ft/s]	48.44	48.43
Desviación típica [ft/s]	61.77	61.82
Percentil 75 [ft/s]	89.81	89.78
Percentil 25 [ft/s]	22.22	22.21

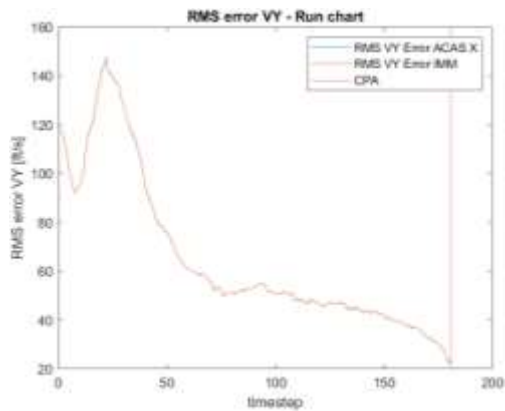


Ilustración 40. Secuencia del error RMS VY

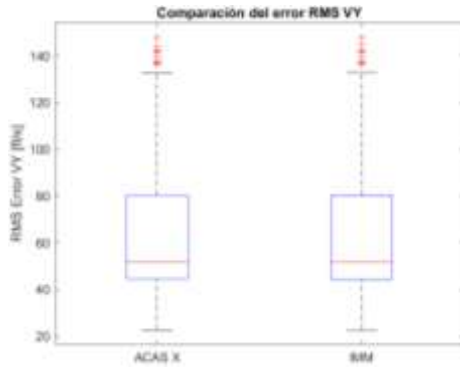


Ilustración 41. Gráfico “Whisker box” del error RMS VY

**Test de hipótesis “2 sample t test”:** No se rechaza la hipótesis nula (las medias de las poblaciones son iguales), por lo que las medias poblacionales de ACAS X e IMM son iguales.

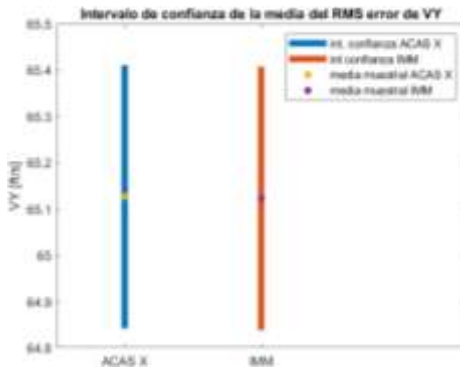


Ilustración 42. Intervalo de confianza de la media del error RMS VY

**Test de hipótesis “Kolmogorov-Smirnov”:** No se rechaza la hipótesis nula (los datos provienen de la misma distribución), por lo que las muestras de ACAS X e IMM pertenecen a la misma población.

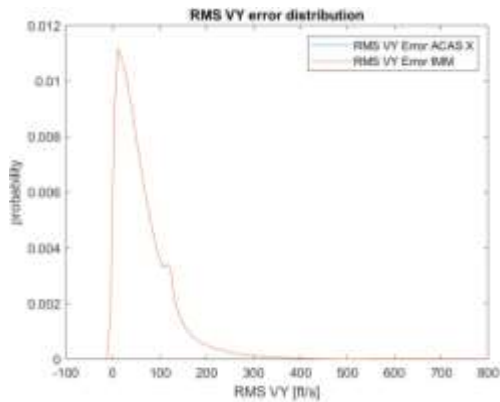


Ilustración 43. Distribución del error RMS VY

**Test de hipótesis de la diferencia de error:** No se rechaza la hipótesis nula (Algoritmo IMM no es mejor que ACAS X)

### NEES

El error cuadrático medio del error relativo NEES, presenta las siguientes características:

NEES	ACAS X	IMM
Media [-]	4.04	3.90
Mediana [-]	2.32	2.32
Desviación típica [-]	22.42	21.72
Percentil 75 [-]	4.38	4.38
Percentil 25 [-]	1.18	1.18

**Test de hipótesis de las regiones de probabilidad:** como se puede ver en la siguiente gráfica, los errores de estimación normalizados están dentro de la región de probabilidad del 95% la mayoría de la simulación, por lo que ambos filtros son consistentes.

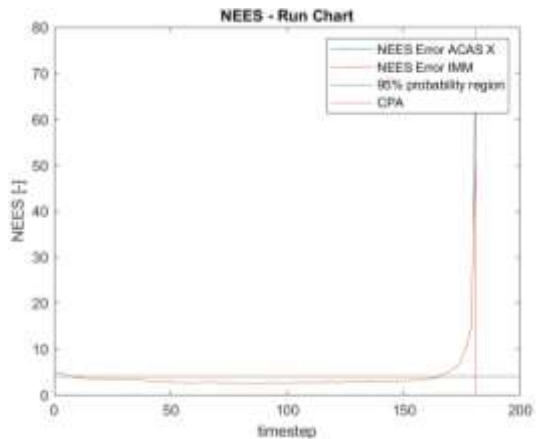


Ilustración 44. Secuencia del error NEES

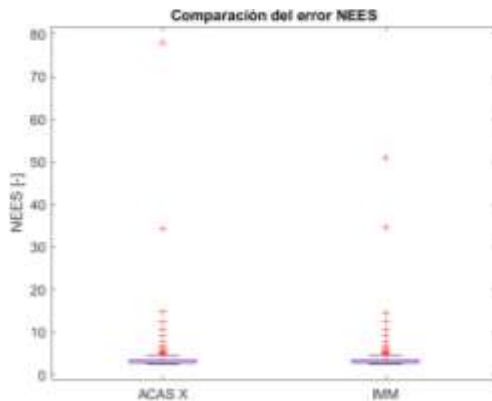


Ilustración 45. Gráfico "Whisker box" del error NEES

**Test de hipótesis "2 sample t test":** Se rechaza la hipótesis nula (las medias de las poblaciones son iguales), por lo que las medias poblacionales de ACAS X e IMM son distintas.

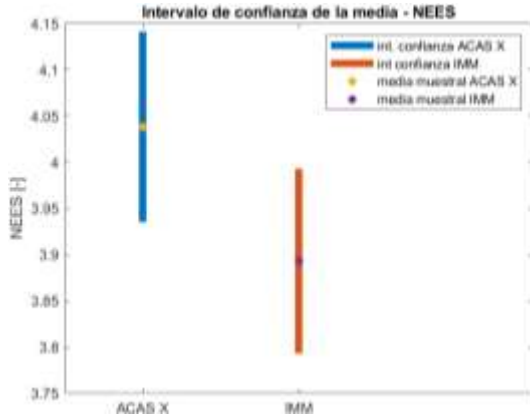


Ilustración 46. Intervalo de confianza de la media del error NEEs

**Test de hipótesis “Kolmogorov-Smirnov”:** No se rechaza la hipótesis nula (los datos provienen de la misma distribución), por lo que las muestras de ACAS X e IMM pertenecen a la misma población.

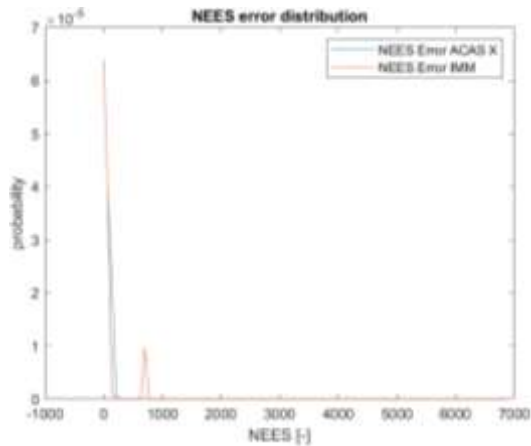


Ilustración 47. Distribución del error NEEs

**Test de hipótesis de la diferencia de error:** No se rechaza la hipótesis nula (Algoritmo IMM no es mejor que ACAS X)

## 7.2 Encuentro 2

### 7.2.1 Geometría

En el encuentro número 2 la aeronave 1 hace un viraje de 180° (desde 0° hasta 180°), mientras que la aeronave 2 vuela en movimiento rectilíneo uniforme rumbo 20°.

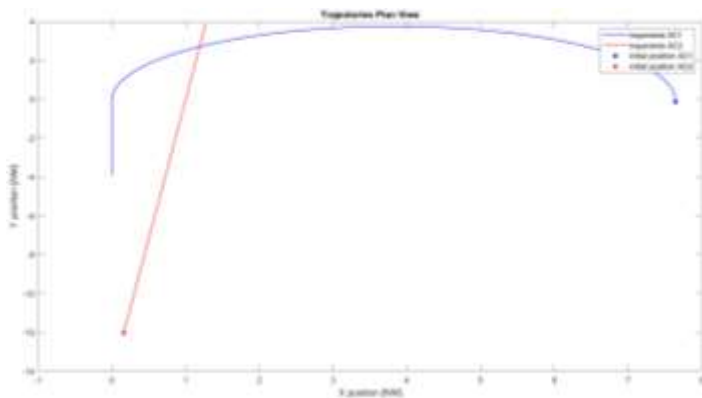


Ilustración 48. Vista horizontal del encuentro

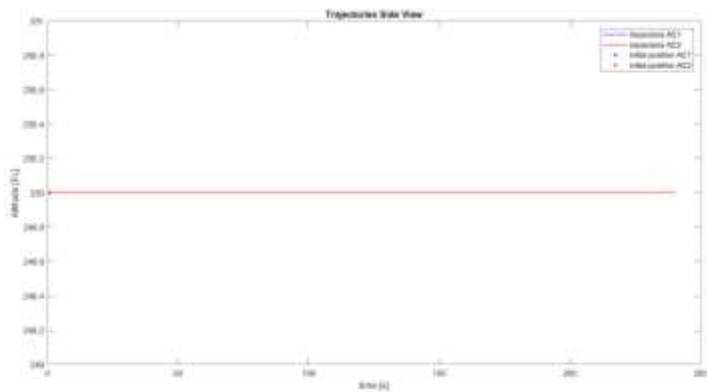


Ilustración 49. Vista de perfil del encuentro

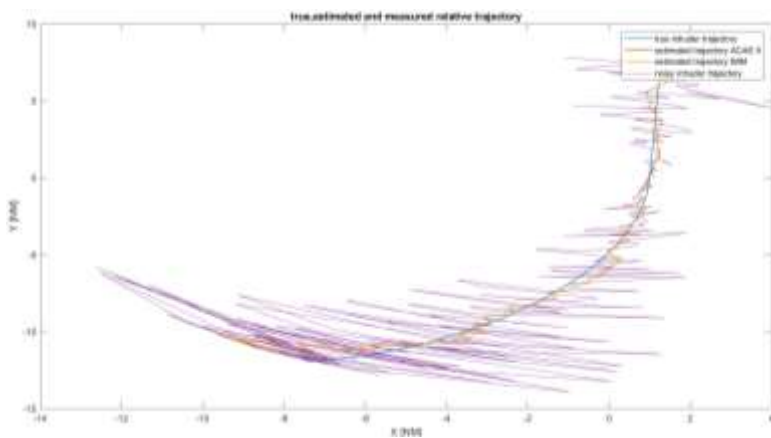


Ilustración 50. Trayectoria estimada

## 7.2.2 Métricas

### *RMS X*

RMS X	ACAS X	IMM
Media [ft]	2361.17	2360.78
Mediana [ft]	1687.39	1687.15

Desviación típica [ft]	2459.13	2458.85
Percentil 75 [ft]	3276.67	3276.35
Percentil 25 [ft]	687.17	686.52

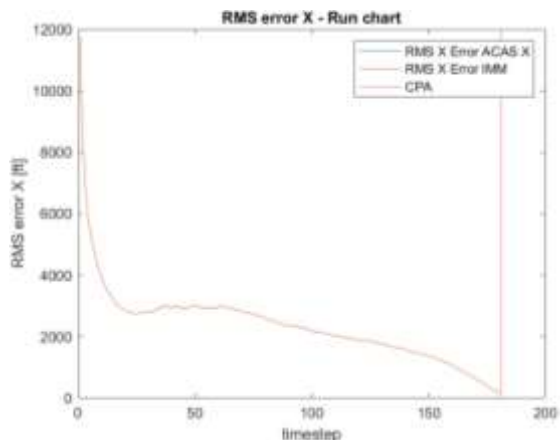


Ilustración 51. Secuencia del error RMS X

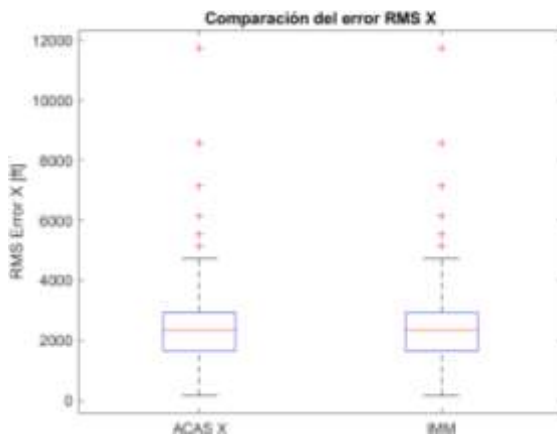


Ilustración 52. Gráfico "Whisker box" del error RMS X

**Test de hipótesis “2 sample t test”:** no se rechaza la hipótesis nula (las medias de las poblaciones son iguales).

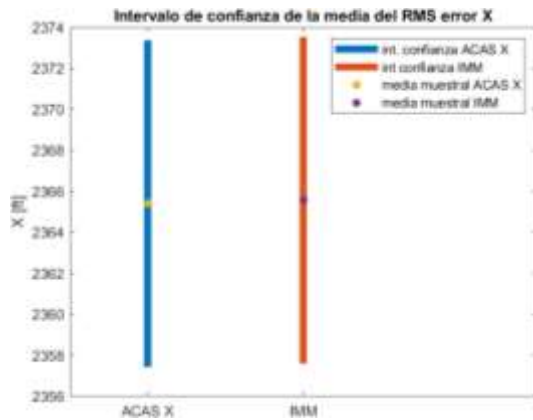


Ilustración 53. Intervalo de confianza de la media del error RMS X

**Test de hipótesis “Kolmogorov-Smirnov”:** no se rechaza la hipótesis nula (los datos provienen de la misma distribución)

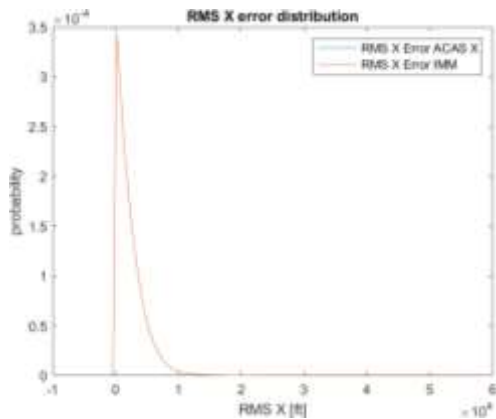


Ilustración 54. distribución del error RMS X

**Test de hipótesis de la diferencia de error:** No se rechaza la hipótesis nula (Algoritmo IMM no es mejor que ACAS X).

### *RMS Y*

El error cuadrático medio de la coordenada y, presenta las siguientes características:

<b>RMS Y</b>	<b>ACAS X</b>	<b>IMM</b>
Media [ft]	1184.85	1185.06
Mediana [ft]	665.96	665.93
Desviación típica [ft]	1545.67	1545.95
Percentil 75 [ft]	1529.49	1529.76
Percentil 25 [ft]	275.78	275.72

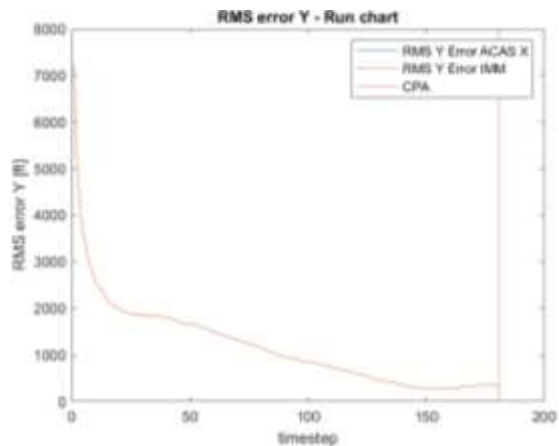


Ilustración 55. Secuencia del error RMS Y

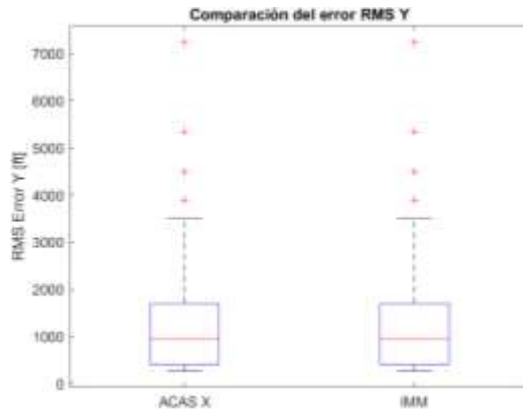


Ilustración 56. Gráfico “Whisker box” del error RMS Y

**Test de hipótesis “2 sample t test”:** no se rechaza la hipótesis nula (las medias de las poblaciones son iguales).



Ilustración 57. Intervalo de confianza de la media del error RMS Y

**Test de hipótesis “Kolmogorov-Smirnov”:** no se rechaza la hipótesis nula (los datos provienen de la misma distribución)

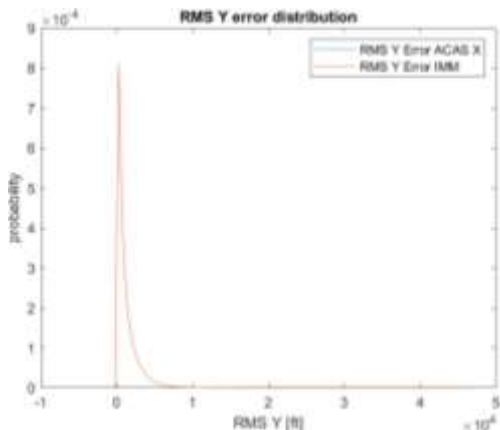


Ilustración 58. Distribución del error RMS Y

**Test de hipótesis de la diferencia de error:** No se rechaza la hipótesis nula (Algoritmo IMM no es mejor que ACAS X).

### *RMS VX*

El error cuadrático medio de la componente x de la velocidad, presenta las siguientes características:

<b>RMS VX</b>	<b>ACAS X</b>	<b>IMM</b>
Media [ft/s]	106.44	103.27
Mediana [ft/s]	88.50	83.38
Desviación típica [ft/s]	83.35	84.46
Percentil 75 [ft/s]	147.92	143.62
Percentil 25 [ft/s]	43.74	40.11

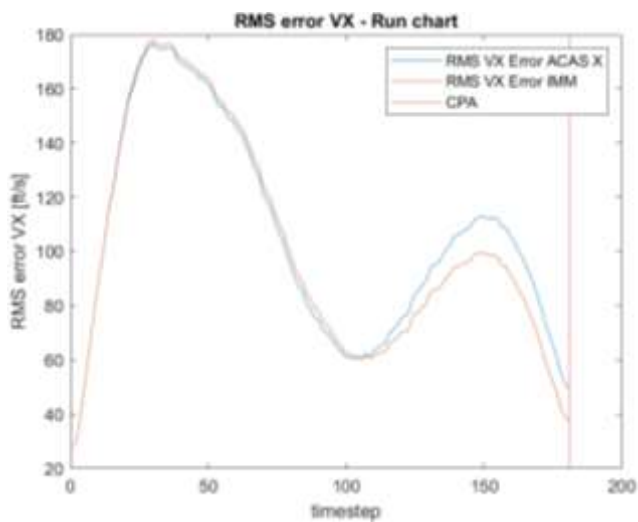


Ilustración 59. Secuencia del error RMS VX

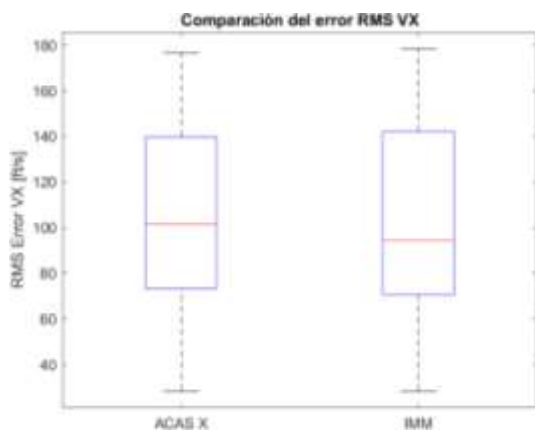


Ilustración 60. Gráfico "Whisker box" del error RMS VX

**Test de hipótesis “2 sample t test”:** se rechaza la hipótesis nula (las medias de las poblaciones son iguales), es decir las medias poblacionales de ACAS X e IMM no son iguales.

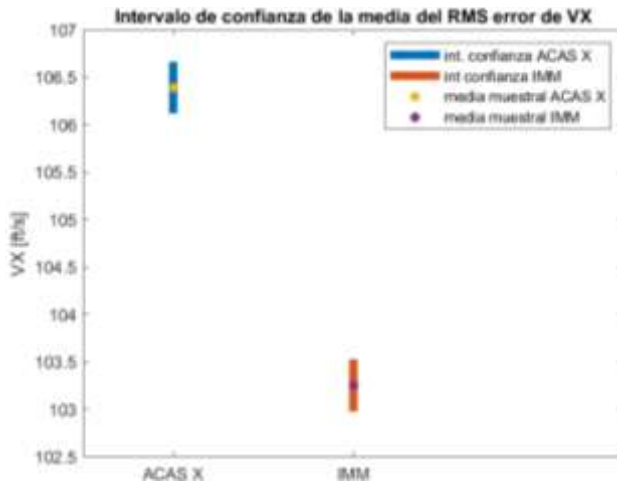


Ilustración 61. Intervalo de confianza de la media del error RMS VX

**Test de hipótesis “Kolmogorov-Smirnov”:** Se rechaza la hipótesis nula (los datos provienen de la misma distribución), por lo que las muestras de ACAS X e IMM pertenecen a poblaciones distintas.

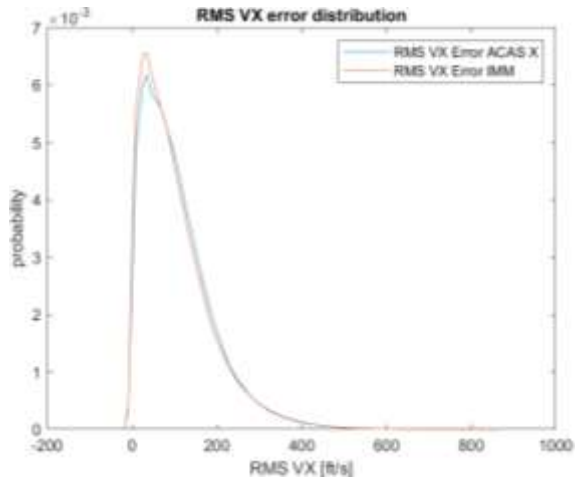


Ilustración 62. Distribución del error RMS VX

**Test de hipótesis de la diferencia de error:** Se rechaza la hipótesis nula (Algoritmo IMM no es mejor que ACAS X), por lo que el algoritmo IMM es mejor que ACAS X.

### *RMS VY*

El error cuadrático medio de la componente y de la velocidad, presenta las siguientes características:

RMS VY	ACAS X	IMM
Media [ft/s]	74.95	69.97
Mediana [ft/s]	69.95	64.31
Desviación típica [ft/s]	51.81	49.68
Percentil 75 [ft/s]	109.10	101.54
Percentil 25 [ft/s]	32.74	29.95

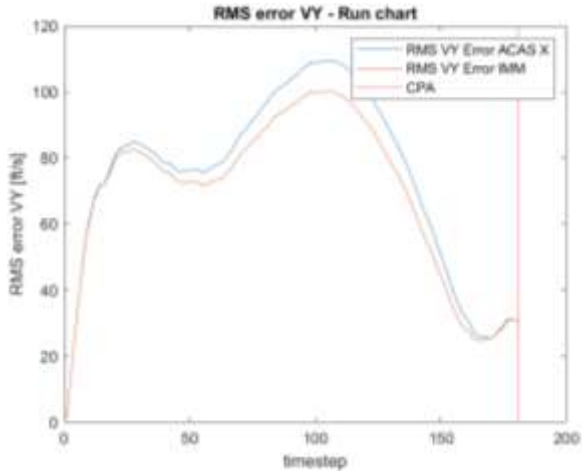


Ilustración 63. Secuencia del error RMS VY

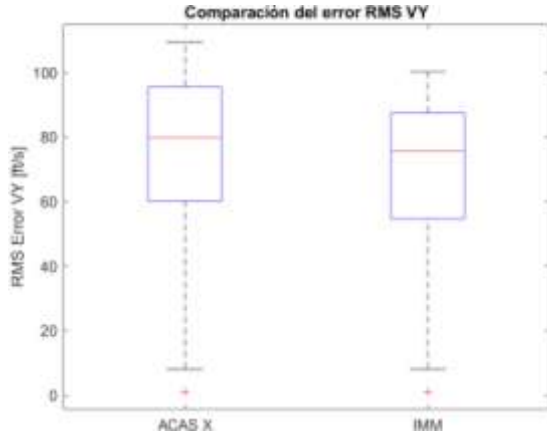


Ilustración 64. Gráfico "Whisker box" del error RMS VY

**Test de hipótesis "2 sample t test":** Se rechaza la hipótesis nula (las medias de las poblaciones son iguales), por lo que las medias poblacionales de ACAS X e IMM son distintas.

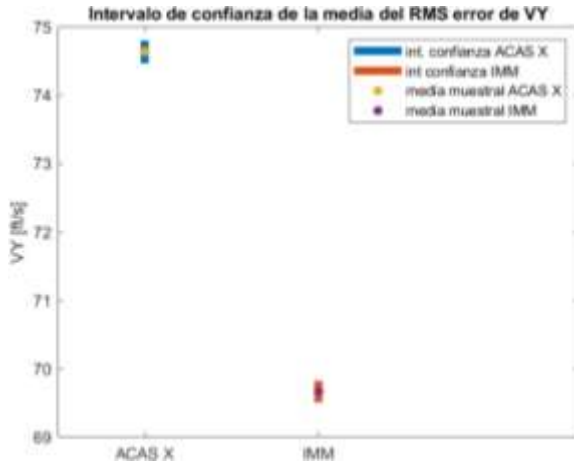


Ilustración 65. Intervalo de confianza de la media del error RMS VY

**Test de hipótesis “Kolmogorov-Smirnov”:** Se rechaza la hipótesis nula (los datos provienen de la misma distribución), por lo que las muestras de ACAS X e IMM pertenecen a poblaciones distintas.

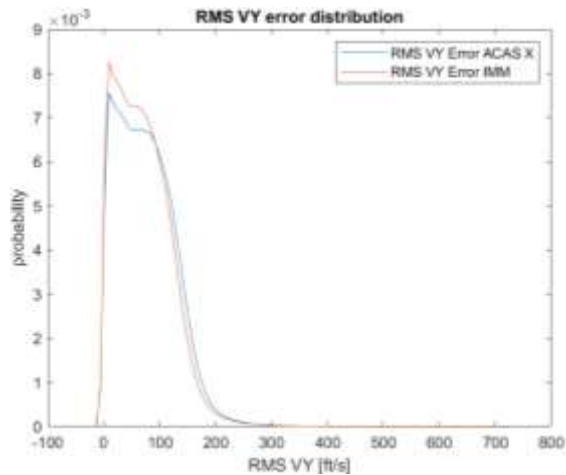


Ilustración 66. Distribución del error RMS VY

**Test de hipótesis de la diferencia de error:** Se rechaza la hipótesis nula (Algoritmo IMM no es mejor que ACAS X), por lo que el algoritmo IMM es mejor que ACAS X.

### NEES

El error cuadrático medio del error relativo NEES, presenta las siguientes características:

NEES	ACAS X	IMM
Media [-]	4.45	4.12
Mediana [-]	3.76	3.42
Desviación típica [-]	3.24	3.19
Percentil 75 [-]	5.58	5.19
Percentil 25 [-]	2.47	2.17

**Test de hipótesis de las regiones de probabilidad:** como se puede ver en la siguiente gráfica, el error de estimación normalizado de ACAS X es menos consistente que el filtro basado en múltiples modelos, ya que éste último pasa más tiempo dentro de la región de probabilidad del 95%.

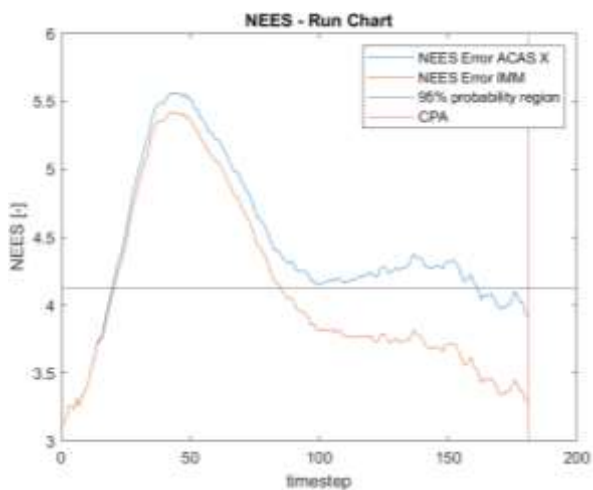


Ilustración 67. Secuencia del error NEES

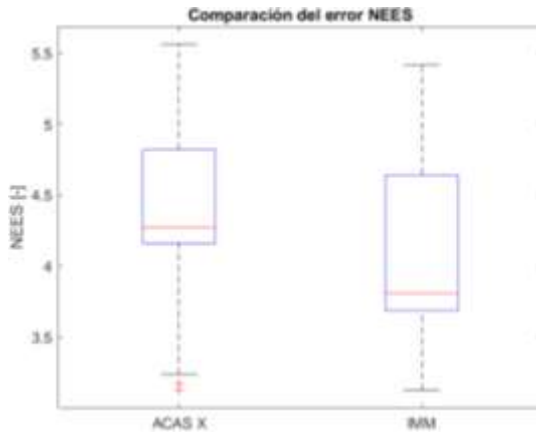


Ilustración 68. Gráfico “Whisker box” del error NEEES

**Test de hipótesis “2 sample t test”:** Se rechaza la hipótesis nula (las medias de las poblaciones son iguales), por lo que las medias poblacionales de ACAS X e IMM son distintas.



Ilustración 69. Intervalo de confianza de la media del error NEEES

**Test de hipótesis “Kolmogorov-Smirnov”:** Se rechaza la hipótesis nula (los datos provienen de la misma distribución), por lo que las muestras de ACAS X e IMM pertenecen a poblaciones distintas.

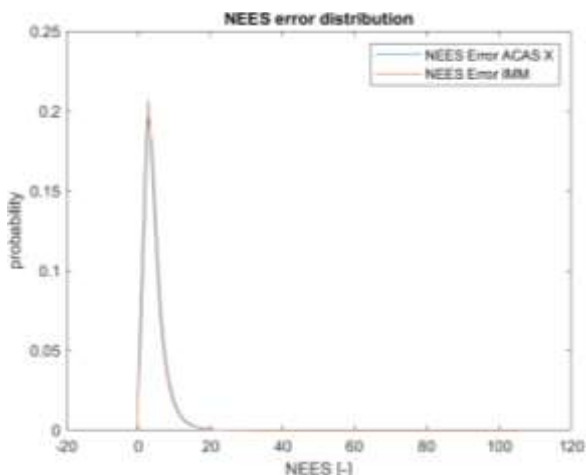


Ilustración 70. Distribución del error NEES

**Test de hipótesis de la diferencia de error:** Se rechaza la hipótesis nula (Algoritmo IMM no es mejor que ACAS X), por lo que el algoritmo IMM es mejor que ACAS X.

## 7.3 Encuentro 3

### 7.3.1 Geometría

En este caso, las dos aeronaves se encuentran haciendo un viraje en sentido contrario (aeronave 1 anti-horario, y aeronave 2 horario), por lo que la trayectoria relativa de la aeronave 2 respecto de la 1 será una trayectoria no lineal.

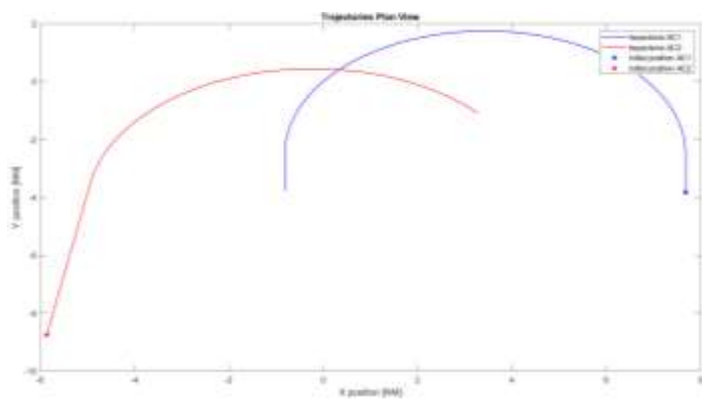


Ilustración 71. Vista horizontal del encuentro

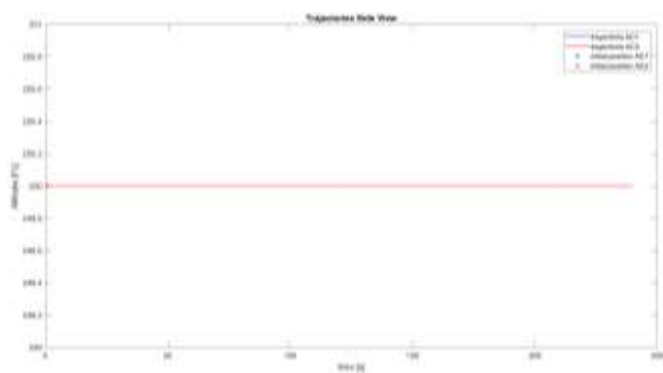


Ilustración 72. Vista de perfil del encuentro

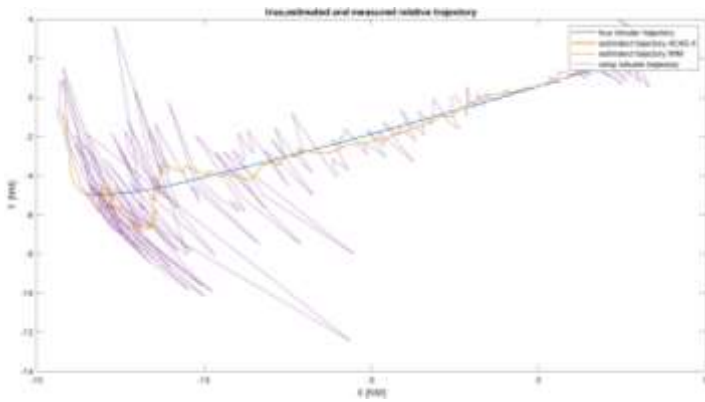


Ilustración 73. Trayectoria estimada

## 7.3.2 Métricas

### *RMS X*

<b>RMS X</b>	<b>ACAS X</b>	<b>IMM</b>
Media [ft]	2271.23	2271.48
Mediana [ft]	1669.05	1669.42
Desviación típica [ft]	2272.31	2272.37
Percentil 75 [ft]	3178.68	3180.21
Percentil 25 [ft]	672.79	672.99

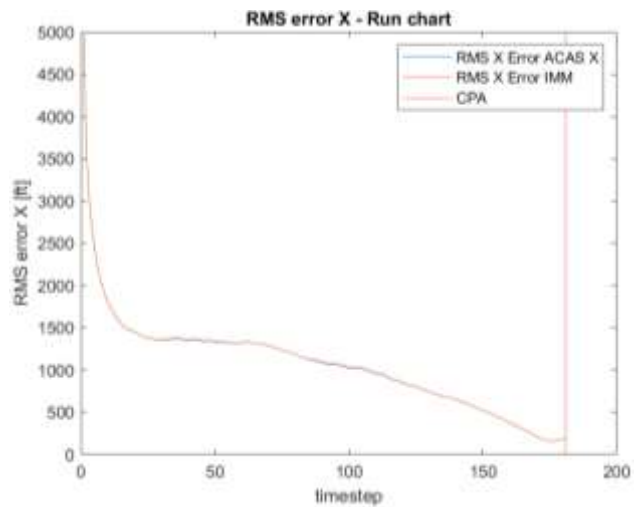


Ilustración 74. Secuencia del error RMS X

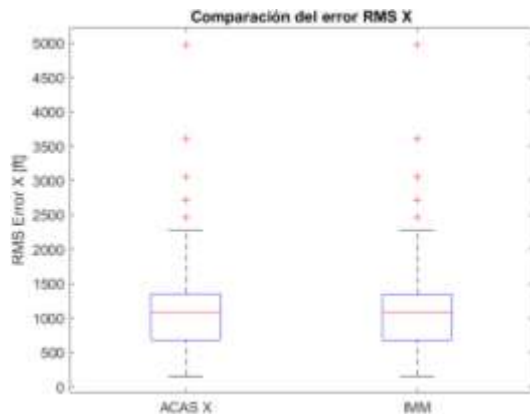


Ilustración 75. Gráfico "Whisker box" del error RMS X

**Test de hipótesis "2 sample t test":** no se rechaza la hipótesis nula (las medias de las poblaciones son iguales).

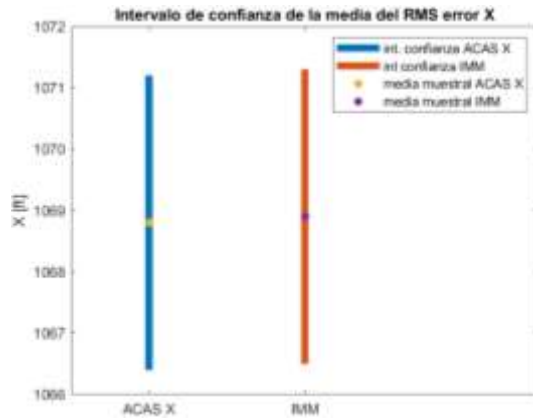


Ilustración 76. Intervalo de confianza de la media del error RMS X

**Test de hipótesis “Kolmogorov-Smirnov”:** no se rechaza la hipótesis nula (los datos provienen de la misma distribución)

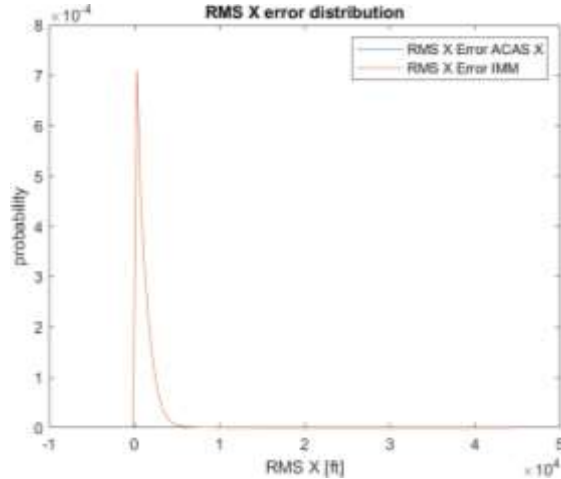


Ilustración 77. Distribución del error RMS X

**Test de hipótesis de la diferencia de error:** No se rechaza la hipótesis nula (Algoritmo IMM no es mejor que ACAS X)

### *RMS Y*

El error cuadrático medio de la coordenada y, presenta las siguientes características:

<b>RMS Y</b>	<b>ACAS X</b>	<b>IMM</b>
Media [ft]	1091.50	1093.00
Mediana [ft]	614.29	616.78
Desviación típica [ft]	1469.33	1468.80
Percentil 75 [ft]	1311.37	1314.32
Percentil 25 [ft]	271.50	272.70

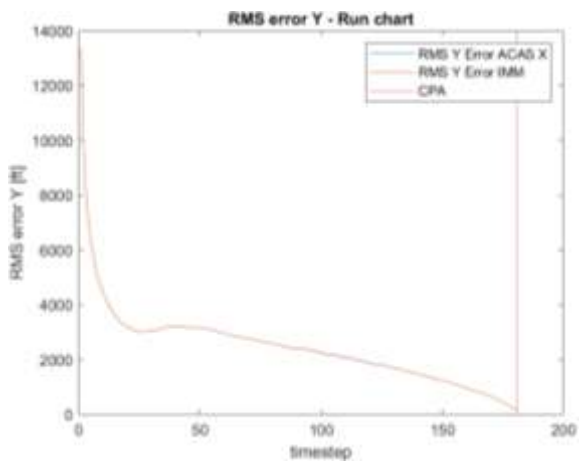


Ilustración 78. Secuencia del error RMS Y

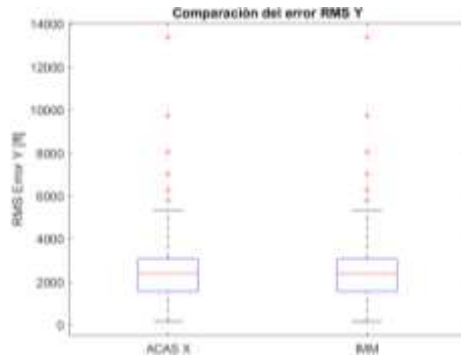


Ilustración 79. Gráfico “Whisker box” del error RMS Y

**Test de hipótesis “2 sample t test”:** no se rechaza la hipótesis nula (las medias de las poblaciones son iguales).



Ilustración 80. Intervalo de confianza de la media del error RMS Y

**Test de hipótesis “Kolmogorov-Smirnov”:** no se rechaza la hipótesis nula (los datos provienen de la misma distribución)

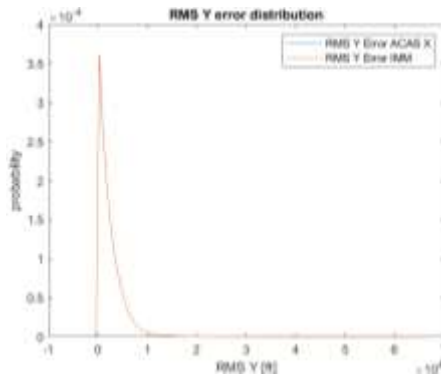


Ilustración 81. Distribución del error RMS Y

**Test de hipótesis de la diferencia de error:** No se rechaza la hipótesis nula (Algoritmo IMM no es mejor que ACAS X)

### **RMS VX**

El error cuadrático medio de la componente x de la velocidad, presenta las siguientes características:

Tabla 11. parámetros del error RMS VX para el encuentro 3

RMS VX	ACAS X	IMM
Media [ft/s]	144.48	135.88
Mediana [ft/s]	138.46	127.20
Desviación típica [ft/s]	89.55	89.97
Percentil 75 [ft/s]	198.17	187.42
Percentil 25 [ft/s]	76.21	66.64

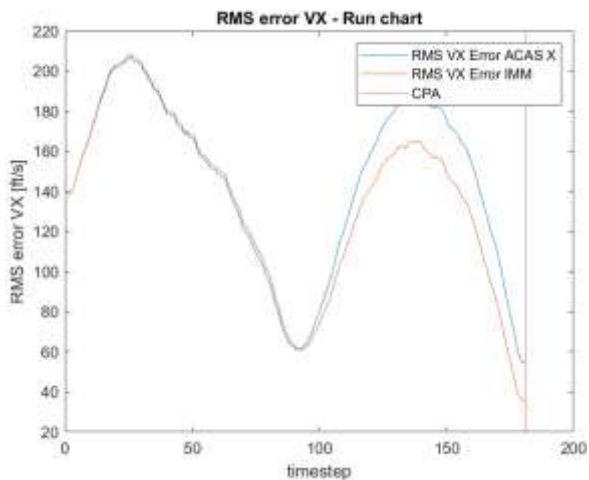


Ilustración 82. Secuencia del error RMS VX

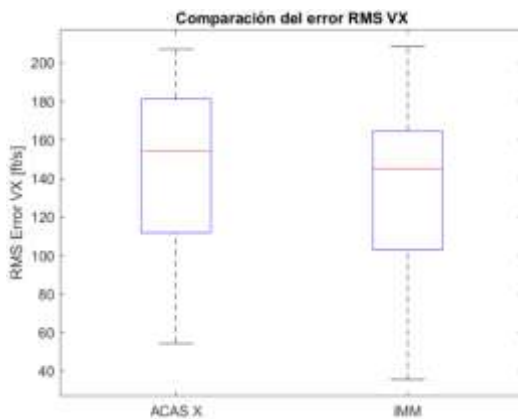


Ilustración 83. Gráfico "Whisker box" del error RMS VX

**Test de hipótesis "2 sample t test":** se rechaza la hipótesis nula (las medias de las poblaciones son iguales), es decir las medias poblacionales de ACAS X e IMM no son iguales.

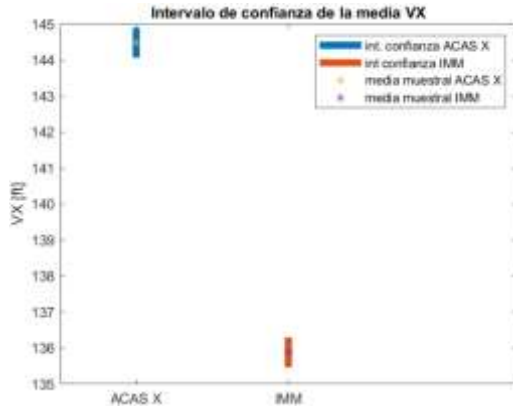


Ilustración 84. Intervalo de confianza de la media del error RMS VX

**Test de hipótesis “Kolmogorov-Smirnov”:** Se rechaza la hipótesis nula (los datos provienen de la misma distribución), por lo que las muestras de ACAS X e IMM pertenecen a poblaciones distintas.

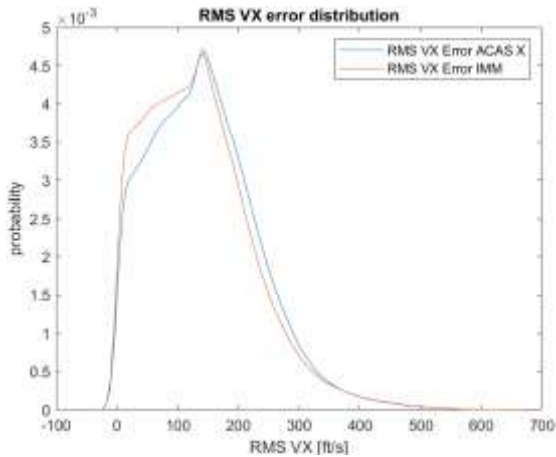


Ilustración 85. Distribución del error RMS VX

**Test de hipótesis de la diferencia de error:** Se rechaza la hipótesis nula (Algoritmo IMM no es mejor que ACAS X), por lo que el algoritmo IMM es mejor que ACAS X.

### *RMS VY*

El error cuadrático medio de la componente y de la velocidad, presenta las siguientes características:

<b>RMS VY</b>	<b>ACAS X</b>	<b>IMM</b>
Media [ft/s]	71.51	65.10
Mediana [ft/s]	64.51	57.65
Desviación típica [ft/s]	51.09	48.18
Percentil 75 [ft/s]	105.56	95.06
Percentil 25 [ft/s]	29.16	26.23

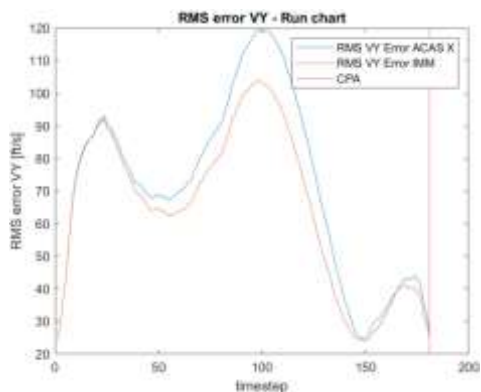


Ilustración 86. Secuencia del error RMS VY

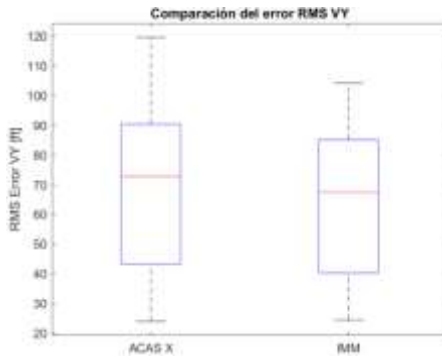


Ilustración 87. Gráfico “Whisker box” del error RMS VY

**Test de hipótesis “2 sample t test”:** Se rechaza la hipótesis nula (las medias de las poblaciones son iguales), por lo que las medias poblacionales de ACAS X e IMM son distintas.



Ilustración 88. Intervalo de confianza de la media del error RMS VY

**Test de hipótesis “Kolmogorov-Smirnov”:** Se rechaza la hipótesis nula (los datos provienen de la misma distribución), por lo que las muestras de ACAS X e IMM pertenecen a poblaciones distintas.

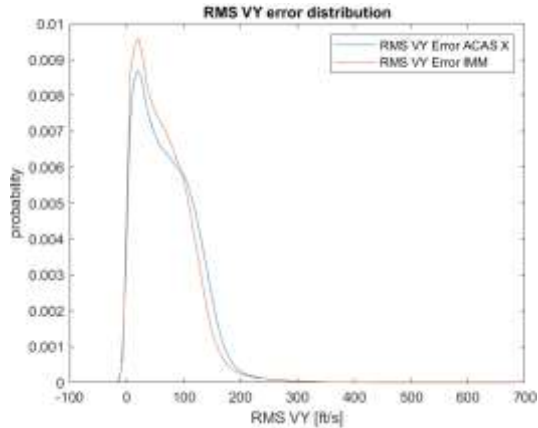


Ilustración 89. Distribución del error RMS VY

**Test de hipótesis de la diferencia de error:** Se rechaza la hipótesis nula (Algoritmo IMM no es mejor que ACAS X), por lo que el algoritmo IMM es mejor que ACAS X.

### NEES

El error cuadrático medio del error relativo NEES, presenta las siguientes características:

NEES	ACAS X	IMM
Media [-]	5.31	4.56
Mediana [-]	4.62	3.85
Desviación típica [-]	3.37	3.16
Percentil 75 [-]	6.81	5.82
Percentil 25 [-]	3.04	2.48

**Test de hipótesis de las regiones de probabilidad:** como se puede ver en la siguiente gráfica, el error de estimación normalizado de ACAS X es menos

consistente que el filtro basado en múltiples modelos, ya que éste último pasa más tiempo dentro de la región de probabilidad del 95%.

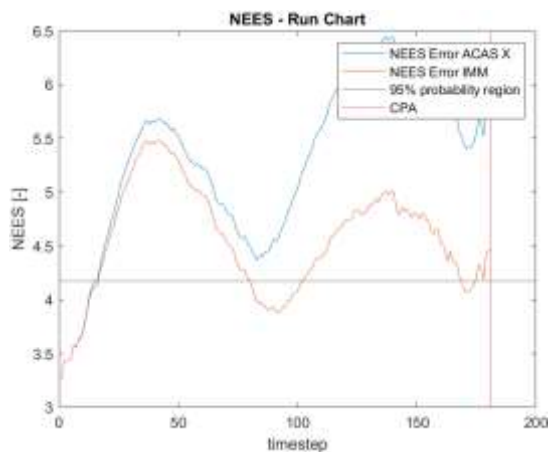


Ilustración 90. Secuencia del error NEES

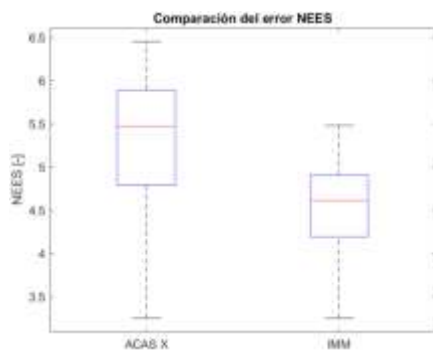


Ilustración 91. Gráfico "Whisker box" del error NEES

**Test de hipótesis “2 sample t test”:** Se rechaza la hipótesis nula (las medias de las poblaciones son iguales), por lo que las medias poblacionales de ACAS X e IMM son distintas.

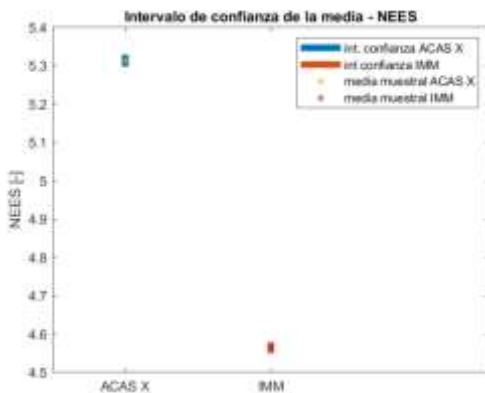


Ilustración 92. Intervalo de confianza de la media del error NEES

**Test de hipótesis “Kolmogorov-Smirnov”:** Se rechaza la hipótesis nula (los datos provienen de la misma distribución), por lo que las muestras de ACAS X e IMM pertenecen a poblaciones distintas.

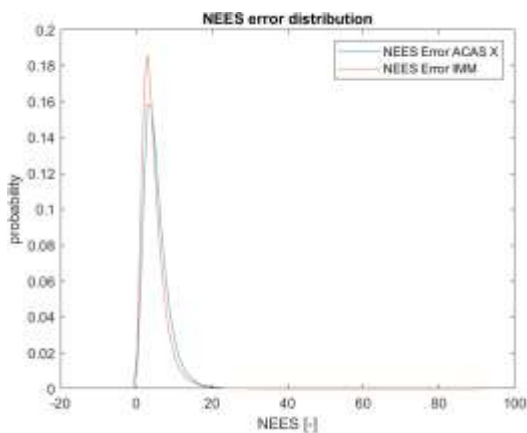


Ilustración 93. Distribución del error NEES

**Test de hipótesis de la diferencia de error:** Se rechaza la hipótesis nula (Algoritmo IMM no es mejor que ACAS X), por lo que el algoritmo IMM es mejor que ACAS X.

## 7.4 Encuentro 4

### 7.4.1 Geometría

En este caso, las dos aeronaves se encuentran haciendo un viraje en el mismo sentido (horario), por lo que la trayectoria relativa de la aeronave 2 respecto de la 1 será una trayectoria no lineal.

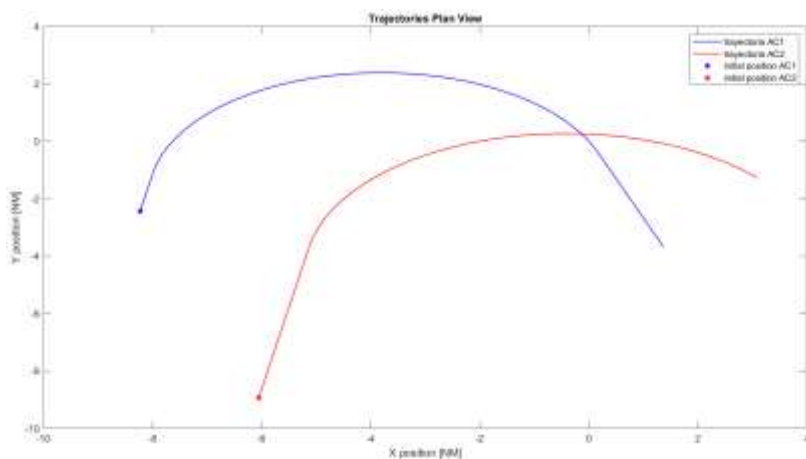


Ilustración 94. Vista horizontal del encuentro

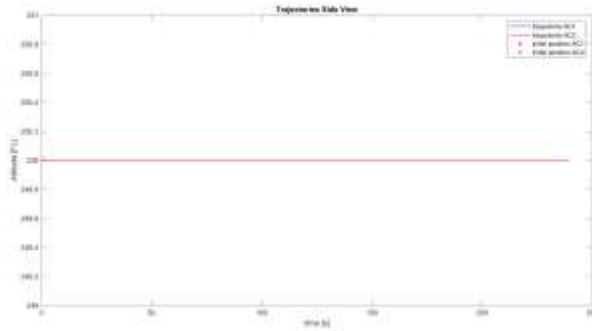


Ilustración 95. Vista de perfil del encuentro

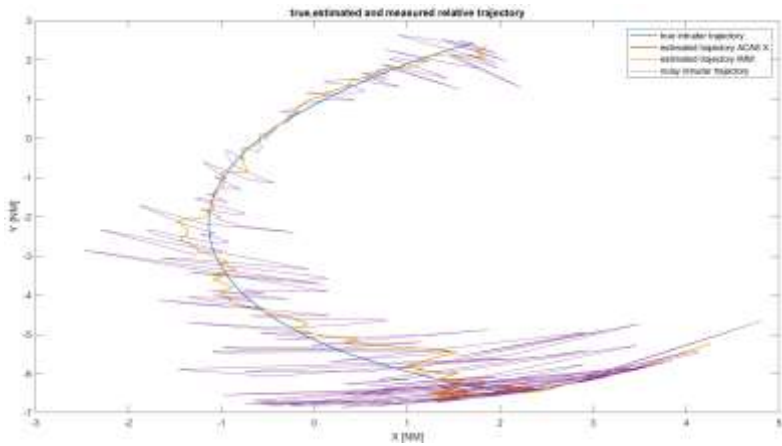


Ilustración 96. Trayectoria estimada

## 7.4.2 Métricas

### *RMS X*

<b>RMS X</b>	<b>ACAS X</b>	<b>IMM</b>
Media [ft]	1426.13	1397.70
Mediana [ft]	1006.36	986.42
Desviación típica [ft]	1455.87	1432.30
Percentil 75 [ft]	2033.09	1991.38

Percentil 25 [ft]	379.27	373.39
-------------------	--------	--------

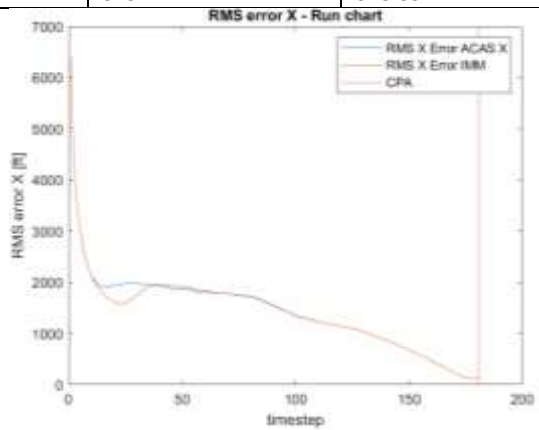


Ilustración 97. Secuencia del error RMS X

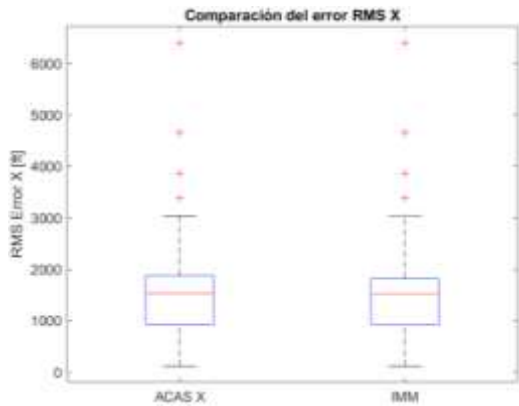


Ilustración 98. Gráfico “Whisker box” del error RMS X

**Test de hipótesis “2 sample t test”:** Se rechaza la hipótesis nula (las medias de las poblaciones son iguales), por lo que las medias poblacionales del error RMS X son distintas.

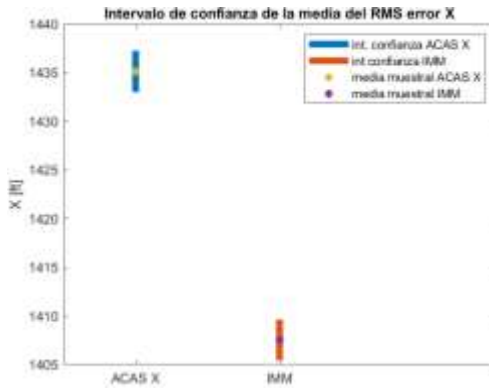


Ilustración 99. Intervale de confianza de la media del error RMS X

**Test de hipótesis “Kolmogorov-Smirnov”:** Se rechaza la hipótesis nula (los datos provienen de la misma distribución), por lo que provienen de distribuciones distintas.

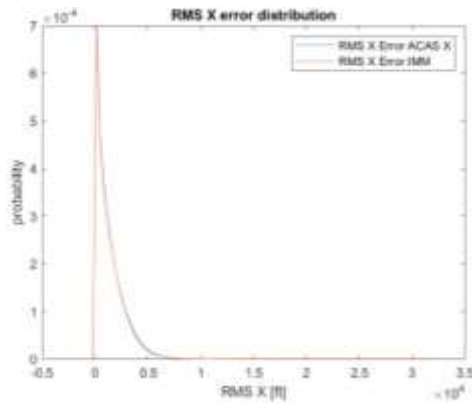


Ilustración 100. Distribución del error RMS X

**Test de hipótesis de la diferencia de error:** No se rechaza la hipótesis nula (Algoritmo IMM no es mejor que ACAS X).

## RMS Y

El error cuadrático medio de la coordenada y, presenta las siguientes características:

RMS Y	ACAS X	IMM
Media [ft]	520.48	509.41
Mediana [ft]	373.84	368.67
Desviación típica [ft]	557.18	543.69
Percentil 75 [ft]	686.45	674.12
Percentil 25 [ft]	172.24	170.35

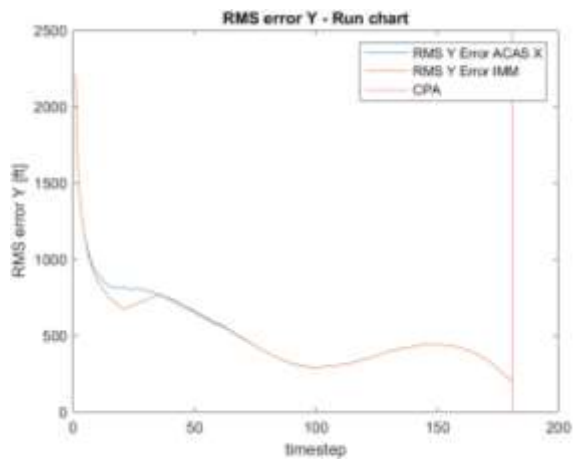


Ilustración 101. Secuencia del error RMS Y

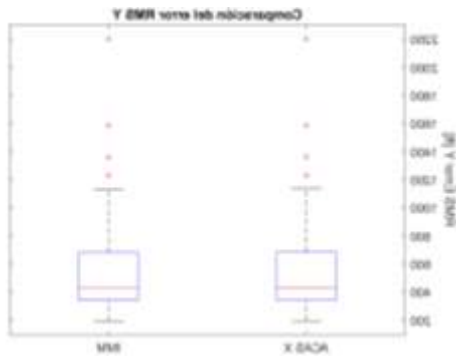


Ilustración 102. Gráfico “Whisker box” del error RMS Y

**Test de hipótesis “2 sample t test”:** Se rechaza la hipótesis nula (las medias de las poblaciones son iguales).



Ilustración 103. Intervalo de confianza de la media del error RMS Y

**Test de hipótesis “Kolmogorov-Smirnov”:** Se rechaza la hipótesis nula (los datos provienen de la misma distribución)

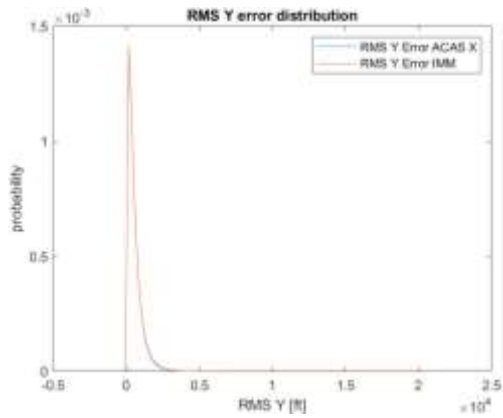


Ilustración 104. Distribución del error RMS Y

**Test de hipótesis de la diferencia de error:** No se rechaza la hipótesis nula (Algoritmo IMM no es mejor que ACAS X).

### *RMS VX*

El error cuadrático medio de la componente x de la velocidad, presenta las siguientes características:

<b>RMS VX</b>	<b>ACAS X</b>	<b>IMM</b>
Media [ft/s]	87.32	80.92
Mediana [ft/s]	73.52	64.84
Desviación típica [ft/s]	68.73	68.48
Percentil 75 [ft/s]	124.40	115.62
Percentil 25 [ft/s]	34.56	28.82

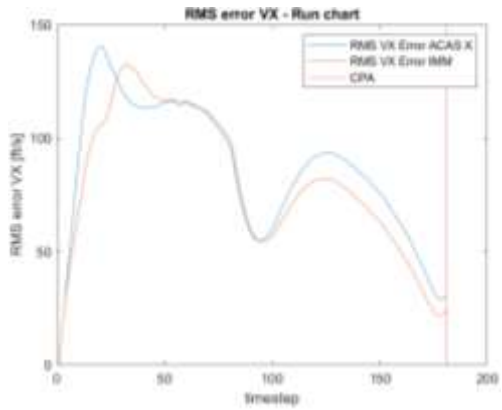


Ilustración 105. Secuencia del error RMS VX

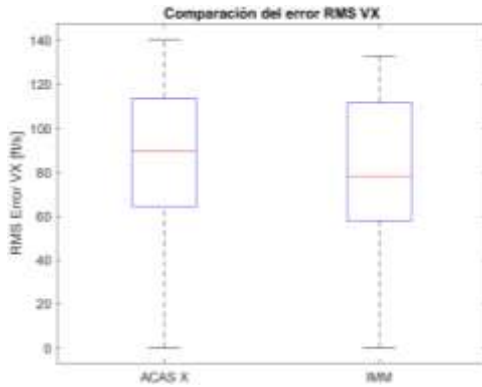


Ilustración 106. Gráfico "Whisker box" del error RMS VX

**Test de hipótesis "2 sample t test":** se rechaza la hipótesis nula (las medias de las poblaciones son iguales), es decir las medias poblacionales de ACAS X e IMM no son iguales.



Ilustración 107. Intervalo de confianza de la media del error RMS VX

**Test de hipótesis “Kolmogorov-Smirnov”:** Se rechaza la hipótesis nula (los datos provienen de la misma distribución), por lo que las muestras de ACAS X e IMM pertenecen a poblaciones distintas.

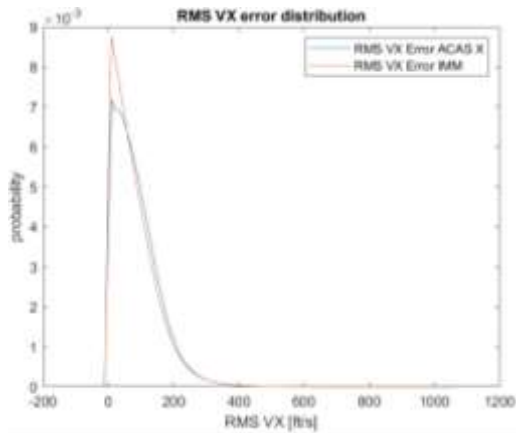


Ilustración 108. Distribución del error RMS VX

**Test de hipótesis de la diferencia de error:** Se rechaza la hipótesis nula (Algoritmo IMM no es mejor que ACAS X), por lo que el algoritmo IMM es mejor que ACAS X.

### *RMS VY*

El error cuadrático medio de la componente y de la velocidad, presenta las siguientes características:

<b>RMS VY</b>	<b>ACAS X</b>	<b>IMM</b>
Media [ft/s]	40.59	39.14
Mediana [ft/s]	35.47	33.58
Desviación típica [ft/s]	31.62	32.10
Percentil 75 [ft/s]	57.58	54.87
Percentil 25 [ft/s]	17.08	16.04

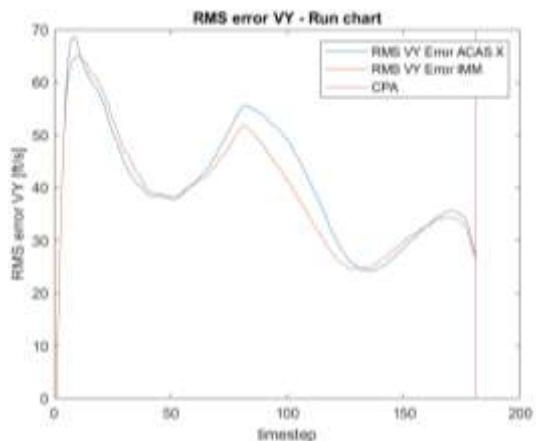


Ilustración 109. Secuencia del error RMS VY

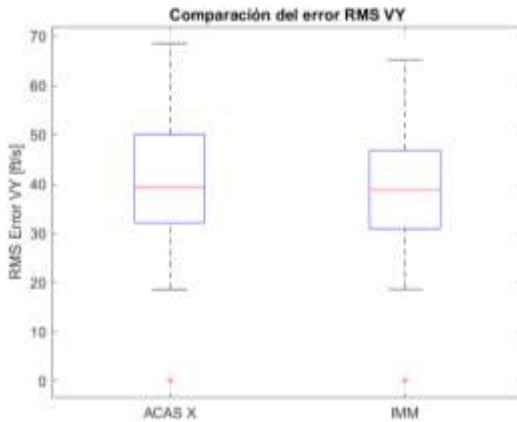


Ilustración 110. Gráfico “Whisker box” del error RMS VY

**Test de hipótesis “2 sample t test”:** Se rechaza la hipótesis nula (las medias de las poblaciones son iguales), por lo que las medias poblacionales de ACAS X e IMM son distintas.



Ilustración 111. Intervalo de confianza de la media del error RMS VY

**Test de hipótesis “Kolmogorov-Smirnov”:** Se rechaza la hipótesis nula (los datos provienen de la misma distribución), por lo que las muestras de ACAS X e IMM pertenecen a poblaciones distintas.

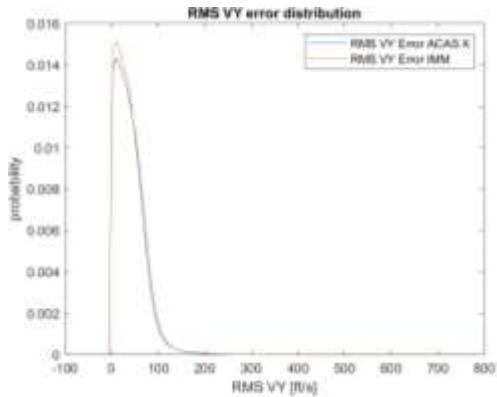


Ilustración 112. Distribución del error RMS VY

**Test de hipótesis de la diferencia de error:** Se rechaza la hipótesis nula (Algoritmo IMM no es mejor que ACAS X), por lo que el algoritmo IMM es mejor que ACAS X.

**NEES**

El error cuadrático medio del error relativo NEES, presenta las siguientes características:

NEES	ACAS X	IMM
Media [-]	3.89	3.47
Mediana [-]	3.07	2.69
Desviación típica [-]	3.33	3.08
Percentil 75 [-]	4.99	4.51
Percentil 25 [-]	1.80	1.50

**Test de hipótesis de las regiones de probabilidad:** como se puede ver en la siguiente gráfica, los errores de estimación normalizados están dentro de la

región de probabilidad del 95% la mayoría de la simulación, por lo que ambos filtros son consistentes.

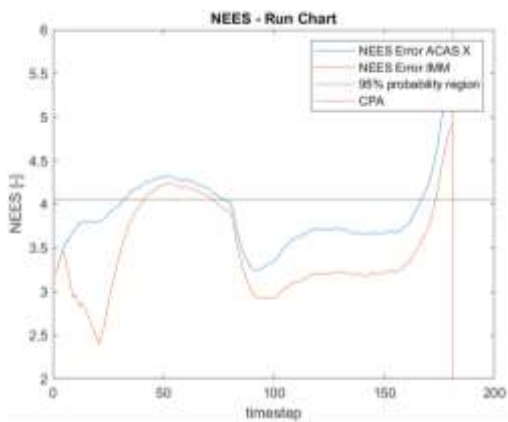


Ilustración 113. Secuencia del error NEES

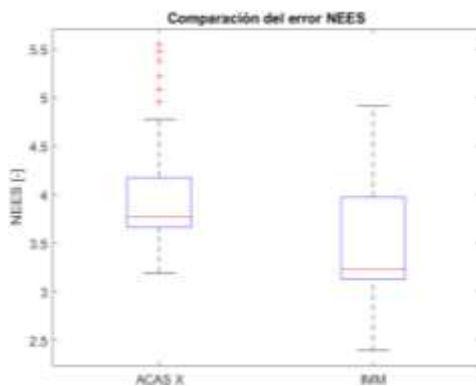


Ilustración 114. Gráfico "Whisker box" del error NEES

**Test de hipótesis "2 sample t test":** Se rechaza la hipótesis nula (las medias de las poblaciones son iguales), por lo que las medias poblacionales de ACAS X e IMM son distintas.

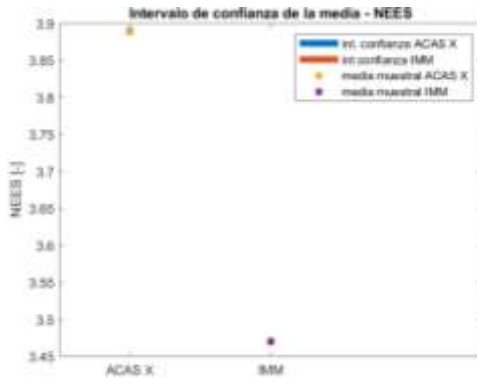


Ilustración 115. Intervalo de confianza de la media del error NEEB

**Test de hipótesis “Kolmogorov-Smirnov”:** Se rechaza la hipótesis nula (los datos provienen de la misma distribución), por lo que las muestras de ACAS X e IMM pertenecen a poblaciones distintas.

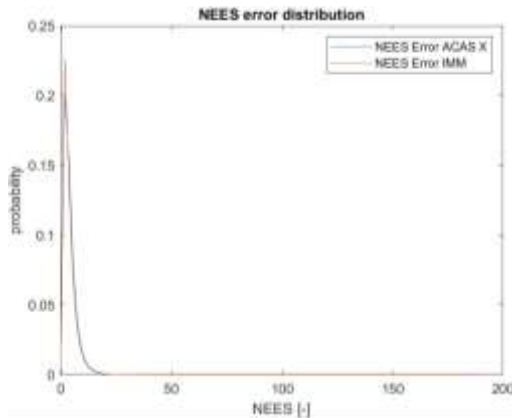


Ilustración 116. Distribución del error NEEB

**Test de hipótesis de la diferencia de error:** Se rechaza la hipótesis nula (Algoritmo IMM no es mejor que ACAS X), por lo que el algoritmo IMM es mejor que ACAS X.

En resumen, ambos algoritmos se comportan igual cuando las aeronaves no están maniobrando (movimiento rectilíneo uniforme) como ocurre en el encuentro 1; y cuando las aeronaves están maniobrando generalmente la estimación de la posición es igual para ambos algoritmos (solo en el encuentro 4 se observa una mejoría significativa, pero no en el resto de encuentros), y la estimación de la velocidad se mejora entre un 3 y 8 % en encuentros donde las aeronaves están maniobrando (encuentros 2, 3 y 4)

### **Impacto del número de ejecuciones de Monte Carlo en los resultados**

Una pregunta que se plantea a la hora de hacer simulaciones de Monte Carlo es ¿Cómo sabemos si el número de ejecuciones de MC es lo suficientemente grande?, es una pregunta que no tiene una fácil solución y es difícil encontrar en la literatura que número de ejecuciones de Monte Carlo es el correcto. Aunque ya hemos visto cómo solucionar este problema de forma experimental al principio del capítulo 7, ahora vamos a ver la aproximación teórica el planteamiento del problema es el siguiente, se quiere conocer el número de muestras necesarias para determinar la media poblacional a través de una muestra, de forma que el error de estimación sea menor que un valor “ $\delta$ ”, y con una confianza del 95%

$$N \geq \left( \frac{z_{1-\frac{\alpha}{2}}}{\delta} \right)^2 \cdot \sigma^2$$

Si la desviación estándar es desconocida, entonces debemos recurrir a los valores críticos de una distribución t-student

$$N \geq \left( \frac{t_{1-\frac{\alpha}{2}}}{\delta} \right)^2 \cdot s^2$$

Donde s, es la desviación típica de la muestra.

El problema es que hay que definir los grados de libertad de la distribución t-student, que es el tamaño muestral menos 1 (N-1), y el tamaño de la muestra es lo que estamos intentando determinar, por lo que hay que resolver la ecuación de forma iterativa.

## 8 CONCLUSIONES Y FUTURAS LÍNEAS DE TRABAJO

Se ha desarrollado una plataforma de simulación de Monte Carlo bastante extensa para analizar y estresar algoritmos de estimación, y se ha probado con los algoritmos anti-colisión ACAS X, con los que se obtiene resultados realistas entre encuentros de aeronaves, y en un tiempo razonable, tras un análisis del rendimiento del simulador se pueden hacer simulaciones de Monte Carlo de 10000 ejecuciones en poco más de una hora.

A continuación, se han definido las métricas y test de hipótesis que son clave para medir el rendimiento y comparar diferentes algoritmos de estimación, y finalmente se ha diseñado e implementado un algoritmo basado en el paradigma de múltiples modelos para intentar mejorar el rendimiento de ACAS X, y comprobar la utilidad de las métricas definidas.

Uno de los requisitos básicos del nuevo algoritmo es al menos tener el mismo rendimiento que ACAS X, respecto a este requisito el encuentro bajo test ha sido un encuentro entre aeronaves con movimiento rectilíneo uniforme, ya que ACAS X está diseñado para estas situaciones y es donde presenta mejor rendimiento.

Se ha comprobado que para encuentros de dos aeronaves aproximándose en movimiento rectilíneo uniforme, los algoritmos ACAS X e IMM presentan el mismo comportamiento.

Posteriormente se han probado encuentros dónde se producen virajes, ya que aquí se esperaba comprobar el mejor rendimiento del nuevo filtro. Para estos encuentros donde al menos una de las dos aeronaves está virando (movimiento circular), y por tanto la trayectoria relativa de una aeronave respecto de la otra es una trayectoria no lineal, se consigue mejorar el error

de estimación de la velocidad para todos los encuentros estudiados, y los test de hipótesis demuestran que esta diferencia es estadísticamente significativa.

Respecto a la posición, el error de estimación es similar, y el test de hipótesis no encuentra diferencias estadísticamente significativas, salvo en el encuentro 4, donde la mejora en el error de estimación tanto en la coordenada x, como la y es estadísticamente significativa.

Al reducir el error de estimación de la velocidad principalmente, se consigue reducir el error normalizado NEES, por lo que el filtro de múltiples modelos es más consistente cuando las trayectorias presentan no linealidades.

Por último, se ha analizado el tiempo de ejecución de los filtros, ya que al ser un algoritmo diseñado para trabajar en tiempo real, tiene una importancia capital. En la versión secuencial del filtro de múltiples modelos el tiempo de ejecución duplica el de ACAS X, pero al paralelizar la ejecución del filtro de múltiples modelos, se obtienen rendimientos similares.

Como futuras líneas de trabajo se pueden investigar sobre diversos temas como análisis de sensibilidad sobre los diferentes parámetros (probabilidades iniciales, matriz de transición de la cadena de Markov del filtro IMM, inicialización del filtro, tiempo de refresco de los sensores), elaboración de una interfaz gráfica que permita probar nuevos modelos de ruido y su impacto en los algoritmos, mejorar la generación de encuentros entre aeronaves mediante el uso de clotoides, añadir test de hipótesis estadísticos sobre las innovaciones del filtro, no solo sobre los errores de estimación, t analizar el impacto de variaciones en el error de estimación en la lógica de resolución del conflicto.

# Bibliografía

- [1] RTCA, Minimum Operational Performance Standards for Airborne Collision Avoidance System X (ACAS X). Volume 1:, 2018.
- [2] RTCA, DO-385, Minimum Operational Performance Standards for Airborne Collision Avoidance System X (ACAS X). Volume 2: Algorithm Design Description, 2018.
- [3] EUROCAE, ED-256, MOPS for ACAS Xa with ACAS Xo functionality, 2018.
- [4] RTCA, DO-386, Minimum Operational Performance Standards for Airborne Collision Avoidance System Xu (ACAS Xu), 2020.
- [5] EUROCAE, ED 275, MOPS for ACAS Xu, 2020.
- [6] S. Stroeve, H. Blom, C. Hernández Medel, C. García Daroca, Á. Arroyo Cebeira y S. Drozdowski, «Development of a Collision Avoidance Validation and Evaluation Tool (CAVEAT): Addressing the intrinsic uncertainty in TCAS II and ACAS X,» de *13th USA/Europe Air Traffic Management Research and Development Seminar*, Vienna, 2019.
- [7] C. Xuwei, Z. Zaitian, R. Haoyu, Q. Fengqi y C. Jianzhou, «Deep Integration Navigation Technique Based on Strong Tracking UKF Algorithm,» de *International Conference on Intelligent Computing*, 2021.
- [8] G. Hu, S. Gao, Y. Zhong, B. Gao y A. Subic, «Modified Strong Tracking unscented Kalman filter for nonlinear state estimation with process model uncertainty,» *International Journal of Adaptive Control and Signal Processing*, 2015.

- [9] H. A. Blom y Y. Bar-Shalom, «Bayesian filtering of stochastic hybrid systems in discrete-time and interacting multiple model,» de *The Oxford Handbook of Nonlinear Filtering*, Oxford, Oxford University Press, 2011, pp. 526-557.
- [10] H. A. Blom y E. A. Bloem, «Exact Bayesian and Particle Filtering of Stochastic Hybrid Systems,» *IEEE Transactions on Aerospace and Electronic Systems*, 2007.
- [11] Q. Xu, X. Li y C.-Y. Chan, «A Cost-Effective Vehicle Localization Solution Using an Interacting Multiple Model Unscented Kalman Filters (IMM-UKF) Algorithm and Grey Neural Network,» *SENSORS*, vol. 17, nº 06, p. 1431, 2017.
- [12] H. Zhiwen y B. Fanliang, «A small UAV tracking algorithm based on AIMM-UKF,» *AIRCRAFT ENGINEERING AND AEROSPACE TECHNOLOGY*, vol. 93, nº 579-592, p. 4, 2021.
- [13] L. Gao, J. Xing, Z. Ma, J. Sha y X. Meng, «Improved IMM algorithm for nonlinear maneuvering target tracking,» de *INTERNATIONAL WORKSHOP ON INFORMATION AND ELECTRONICS ENGINEERING*, 2012.
- [14] Eurocontrol, <https://skybrary.aero/articles/mid-air-collision>.
- [15] EUROCAE, ED-143, Minimum Operational Performance Standards For Traffic Alert And Collision Avoidance System II (TCAS II) Version 7.1, 2013.
- [16] OACI, ACAS Manual, 2006.
- [17] E. Wan y R. Van Der Merwe, «The unscented Kalman filter for nonlinear estimation,» *Proceedings of the IEEE 2000 Adaptive Systems for Signal*

*Processing, Communications, and Control Symposium*, pp. 153-158, 2000.

- [18] RTCA, «<https://www.rtca.org/standards/standards-guidance-materials/>».
- [19] S. Kumar Dash, *Hands-on Julia programming*, BPB Publications, 2021.
- [20] M. J. Kochenderfer, T. A. Wheeler y K. H. Wray, *Algorithms for decision making*, 2022.
- [21] Y. Bar-Shalom, X. Rong Li y T. Kirubarajan, *Estimation with Applications To Tracking and Navigation*, John Wiley & Sons, Inc, 2001.
- [22] J. E. Gentle, *Random number generation and Monte Carlo Methods*, Springer, 2003.
- [23] M. Matsumoto y T. Nishimura, «Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator,» *Transactions on Modeling and computer simulations: Special Issue on uniform Random Generation*, 1998.
- [24] Mathworks, <https://uk.mathworks.com/help/matlab/math/creating-and-controlling-a-random-number-stream.html>.
- [25] W. H. Press, S. A. Teukolsky, W. T. Vetterling y B. P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*, CAMBRIDGE UNIVERSITY PRESS, 2007.
- [26] A. J. Izenman, *Modern Multivariate statistical techniques: Regression, classification, and manifold learning*, Springer, 2008.

- [27] P. J. Brockwell y R. A. Davis, Introduction to time series and forecasting, Springer, 2002.
- [28] Eurocontrol, ACAS Guide, Edition 4.1, 2022.
- [29] OACI, Normas y métodos recomendados internacionales. Anexo 10: Telecomunicaciones aeronáuticas, 2001.
- [30] US Federal Government, «Electronic Code of Federal Regulations, Title 14 Aeronautics and Space,» 2018.
- [31] B. Raynaud y T. Arino, «ACAS safety analysis post-RVSM project; Edition 1.1,» Eurocontrol, 2006.
- [32] H. Hutchinson, «Update of altimetry-error assumptions. ASARP project; ASARP/WP3/16/D, Version 2.0,» 2006.
- [33] L. Sanz, G. Dean, V. Huck, I. Buleza, A. Price y R. Hayward, «VALR-ACASX-OPT1 Validation report for the evaluation of ACAS Xa optimized for Europe. SESAR Joint Undertaking; Project 04.08.01, Deliverable D92; Version 00.01.01,» 2016.
- [34] C. Mari y C. Mouland, «VALR-OT Validation Report on Optimising ACAS RA Thresholds. SESAR Joint Undertaking; Project 04.08.02, D10, Edition 00.01.01,» 2013.
- [35] H. A. P. Blom y Y. Bar-Shalom, «The Interacting Multiple Model Algorithm for Systems with Markovian Switching Coefficients,» *IEEE Transactions on automatic control*, vol. 33, nº 08, pp. 780-783, 1988.
- [36] M. L. Bujorianu, H. A. Blom y H. Hermanns, «Functional Abstractions of Stochastic Hybrid Systems,» de *Analysis and design of Hybrid Systems*,

Alghero, Italy, 2006.

- [37] J. Hu, J. Lygeros y S. Sastry, «Towards a Theory of Stochastic Hybrid Systems,» de *Hybrid systems: computation and control*, Berlin, 2002.
- [38] M. Kochenderfer, L. Espindle, J. Kuchar y J. Griffith , «Correlated encounter model for cooperative aircraft in the National Airspace System Version 1.0,» Lincoln Laboratory, 2008.
- [39] M. J. Kochenderfer, M. W. M. Edwards, L. P. Espindle, J. K. Kuchar y J. D. Griffith, «Airspace Encounter Models for Estimating Collision Risk,» *AIAA, Journal of Guidance, Control, and Dynamics*, vol. 33, nº 03, pp. 487-499, 2010.
- [40] B. Ristic, V. Ba-Ngu, D. Clark y V. Ba-Tuong, «A Metric for Performance Evaluation of Multi-Target Tracking Algorithms,» *IEEE Transactions on signal processing*, vol. 59, nº 07, pp. 3452-3457, 2011.
- [41] S. H. Stroeve, H. A. Blom, C. Hernandez Medel, C. Garcia Daroca, A. Arroyo Cebeira y S. Drozdowski, «Modeling and Simulation of Intrinsic Uncertainties in Validation of Collision Avoidance Systems,» *AIAA Journal of Air Transportation*, 2020.
- [42] W. Yi, L. Jongwoo y Y. Ming-Hsuan, «Object Tracking Benchmark,» *IEEE Transactions on pattern analysis and machine intelligence* , vol. 37, nº 09, pp. 1834-1848, 2015.
- [43] H. Qingshun y D. Bing, «Performance comparison of CKF and UKF in passive state noise of UAV,» de *International Conference on Machine Learning and Computer Application*, 2020.

- [44] Y. Bar-Shalom y X.-R. Li, *Multitarget MultiSensor Tracking: Principles and Techniques*, 1995.
- [45] R. E. Walpole, R. H. Myers, S. L. Myers y K. Ye, *Probability and Statistics for Engineers and Scientists*, Pearson, 2011.
- [46] R. R. Wilcox, *Introduction to Robust Estimation and Hypothesis Testing*, Elsevier, 2012.
- [47] L. Lian-qun y X. Wei-xin, «Bearings-only maneuvering target tracking based on fuzzy clustering in a cluttered environment,» *International Journal of Electronics and Communications*, vol. 68, nº 02, pp. 130-137, 2014.
- [48] A. Arroyo Cebeira y M. Asensio, «On the number of Monte Carlo runs for Stochastic Processes,» de *International Symposium on Unmanned Systems and the Defense Industry (ISUDEF)* , Madrid, 2022.
- [49] A. J. Haug, *Bayesian Estimation and Tracking: A Practical Guide*, John Wiley & Sons, 2012.
- [50] S. Julier y J. Uhlmann, «A New Extension of the Kalman Filter to Nonlinear Systems,» *Signal Processing, Sensor Fusion, and Target Recognition VI*, vol. 3068, 1997.
- [51] R. L. Harrison, «Introduction to Monte Carlo Simulation,» 2010.
- [52] R. Y. Rubinstein y D. P. Kroese, *Simulation and the Monte Carlo Method*, John Wiley & sons, 2016.

# Anexo A: Estimación Bayesiana

En este capítulo pretendemos detallar el marco teórico donde se desarrolla la investigación, también conocido como antecedentes.

Dentro de la teoría de estimación, esta tesis se desarrolla en el marco de la estimación Bayesiana, que es un tipo de inferencia estadística donde las observaciones se utilizan para inferir la probabilidad de que una hipótesis sea cierta; y con tal propósito se basa en el teorema de Bayes. Se diferencia de la inferencia clásica o frecuentista en que ésta solo se basa en la probabilidad frecuentista basada en la frecuencia relativa de un evento después de una secuencia de ensayos, sin tener en cuenta las evidencias o medidas que se toman pues según la inferencia clásica, estas evidencias no modifican la probabilidad.

## Teorema de Bayes

Para entender el teorema de Bayes, conviene que el lector este familiarizado con el concepto de probabilidad, para ello primero se debe distinguir entre probabilidad marginal, probabilidad condicionada y probabilidad conjunta.

La probabilidad condicional es la probabilidad de que ocurra un evento A, sabiendo a ciencia cierta que también sucede (sucedió o sucederá) otro evento B (no tiene por qué haber una relación temporal determinada entre A y B), por tanto el evento B es una evidencia a tener en cuenta.

$$P(A|B) = \frac{P(A, B)}{P(B)}$$

Lo mismo aplica en tiempo continuo para dos variables aleatorias “x” e “y”.

$$P(x|y) = \frac{P(x, y)}{P(y)}$$

Para eventos/variables independientes, entonces

$$P(A|B) = P(A); P(x|y) = P(x)$$

Donde  $P(A, B)$  es la probabilidad conjunta, y  $P(B)$  es la probabilidad marginal.

La probabilidad conjunta  $P(A, B)$  es una función que representa la distribución de probabilidad de la intersección de los eventos A y B, es decir que ambos eventos ocurran simultáneamente; en este caso sí que existe una relación temporal.

Por otra parte, la probabilidad marginal nace a partir del concepto de probabilidad conjunta, ya que representa la función de densidad de probabilidad de un subconjunto de variables aleatorias de un conjunto, por lo que dado el conjunto de variables/eventos  $\{A, B\}$ ,  $P(A, B)$  es su probabilidad conjunta y  $P(B)$  es la probabilidad marginal de B.

$$P(x) = \int_{-\infty}^{\infty} P(x, y) dy$$

Aplicando los conceptos ya vistos, podemos obtener el teorema de probabilidad total, que es fundamental para relacionar la probabilidad marginal con la probabilidad condicional:

$$P(x) = \int_{-\infty}^{\infty} P(x, y) dy = \int_{-\infty}^{\infty} P(x|y) \cdot p(y) dy$$

Además, se puede añadir una condición a todas las probabilidades de la ecuación por lo que el teorema de probabilidad total quedaría:

$$P(x|z) = \int_{-\infty}^{\infty} P(x, y|z) dy = \int_{-\infty}^{\infty} P(x|y, z) \cdot p(y|z) dy$$

Una vez vistos los conceptos de las distintas probabilidades, podemos entender el teorema de Bayes. El teorema de Bayes nos permite determinar la probabilidad condicionada de un evento aleatorio A, dado un evento B (Evidencia).

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Donde:

$P(A|B)$ : Probabilidad a posteriori o condicionada, puesto que es la probabilidad de que ocurra el evento A, sabiendo que el B es verdad.

$P(A)$ : Probabilidad a priori, probabilidad marginal del evento A, sin ningún tipo de condición.

$P(B|A)$ : conocida como “likelihood” o probabilidad condicionada del evento B, dada la evidencia A.

$P(B)$ : Probabilidad marginal del evento B, no depende del evento a estudiar A y simplemente se utiliza para normalizar la ecuación.

La forma en que se utiliza este teorema en los algoritmos de seguimiento de trayectoria es para determinar la probabilidad de que una aeronave este en una posición determinada (x), dado que el punto de ruta anterior es conocido ( $x_{k-1}$ ), es decir:

$$P(x|x_{k-1})$$

**Diferencia entre “Probability” y “Likelihood”**

Aunque en español se utiliza la palabra probabilidad en ambos casos, no es lo mismo y en inglés sí que se diferencian, pasa lo mismo con los conceptos de “safety” y “security” que no son lo mismo, pero en español utilizamos seguridad para ambos.

“Probability” tiene que ver con posibles resultados, es la probabilidad de que un posible resultado se dé, condicionado a una evidencia, que se sabe que es cierta (generalmente medidas observadas o una función de distribución de probabilidad que hemos obtenido a través de las mediciones). Por tanto la suma de todas las probabilidades debe ser igual a uno, ya que todos los potenciales resultados de un sistema para una evidencia dada son mutuamente excluyentes.

$$P(\text{potencial resultado} | \text{evidencia})$$

En estos casos el potencial resultado es variable, mientras que la evidencia (función de distribución de probabilidad es fija).

N.B. incluso cuando no hablamos de probabilidades condicionadas, estas siempre están condicionadas a una evidencia que quizás no se menciona por ser obvia, por ejemplo cuando hablamos de la probabilidad de que al lanzar un dado obtengamos el número 6, es la probabilidad de obtener un 6, dado que tenemos un dado con una función de distribución de probabilidad uniforme para los valores del 1 al 6, por lo que es una probabilidad condicionada por una función de distribución de probabilidad.

$$P(x = 6 | \text{distribución de probabilidad uniforme})$$

Para el caso de seguimiento de aeronaves se utilizará para determinar la función de densidad de probabilidad de la posición de la aeronave dada la trayectoria que hemos estado midiendo (puntos de ruta anteriores):

$$P(\text{posición aeronave} = X_k | \text{mediciones observadas})$$

“Likelihood” se refiere a como una medida observada soporta o apoya una hipótesis (generalmente en forma de función de distribución de probabilidad). Por lo que se podría nombrar como verosimilitud, probabilidad de hipótesis o probabilidad pasada, ya que se evalúa la probabilidad de una hipótesis a la luz de unos resultados (evidencia) que ya han pasado, por lo tanto se trata de encontrar la mejor función de distribución que se ajuste a los datos medidos.

$$P(\text{medida} | \text{hipótesis})$$

En el caso del dado anterior, usaríamos la probabilidad de hipótesis o “likelihood” en el siguiente escenario: Imagínese el lector que ya hemos medido la tirada de un dado 100 veces, y las cien veces ha salido 6, es decir, hemos obtenido de nuestra muestra una distribución de media 6 y desviación típica nula, y queremos evaluar la probabilidad de hipótesis de que el dado sea justo, es decir, siga una distribución uniforme de 1 al 6.

$$P(\mu_x = 6, \sigma_x = 0 | \text{dado justo})$$

Y se puede determinar para infinidad de hipótesis, tantas como funciones de distribución de probabilidad podamos asignar al dado, como por ejemplo que el dado esté trucado, y tenga un 100% de probabilidad de sacar un 6, para esa distribución de probabilidad, verosimilitud sería máxima en nuestro ejemplo anterior.

En el caso de seguimiento de aeronaves, evaluaremos la verosimilitud para determinar la veracidad de cada hipótesis de movimiento de las aeronaves dado las trayectorias que estamos midiendo (puntos de ruta anteriores).

$$P(\text{mediciones observadas} | \text{aeronave maniobrando})$$

Las hipótesis no son mutuamente excluyentes y por tanto las verosimilitudes no tienen por qué sumar uno (e.g. para caracterizar el movimiento de la

aeronave podemos tener las siguientes hipótesis: aeronave en movimiento rectilíneo uniforme, aeronave maniobrando, aeronave ascendiendo, aeronave en viraje coordinado, en las que varias pueden ser ciertas a la vez).

### **Diferencia entre técnicas de filtrado y fusión de datos**

Los filtros de estimación Bayesiana son ampliamente utilizados para muchas aplicaciones, pero hay dos en concreto donde son fundamentales:

1. técnicas de filtrado/estimación: se utilizan para estimar una variable desconocida mediante medidas procedentes de un único sensor. A este tipo de algoritmos se les denomina filtros o técnicas de filtrado porque la variable a estimar se ve interferida por ruidos y errores que hay que filtrar para obtener una variable estimada lo más exacta posible.
2. técnicas de fusión de datos: se utilizan cuando hay varios sensores que miden lo mismo, por ejemplo la posición de una aeronave se puede medir con el radar embarcado de otra aeronave, con un radar secundario basado en tierra (SSR), con un radar primario basado en tierra (PSR), o con ADS-B. Y su objetivo es fusionar los datos procedentes de diferentes sensores para mejorar la exactitud. Durante los inicios de las técnicas de procesamiento de datos provenientes de diferentes sensores (1970-1980) se utilizaban algoritmos de asociación y correlación, una vez que se asociaban los datos de diferentes sensores, se comprobaba cuál de las dos o varias fuentes de datos tenía mejor exactitud, y se eliminaban el resto de datos, de forma que siempre se procesaba únicamente los datos provenientes del sensor con mejor exactitud. Más tarde, en la década de los 90, se desarrolló otra técnica conocida como fusión de trazas o “track blending” donde sí que se fusionan o combinan las trazas provenientes de diferentes sensores, mejorando la exactitud del mejor de los sensores, mediante filtros de Kalman o similares.

Los estimadores lineales se dividen en dos tipos: estáticos y dinámicos.

Los estimadores lineales estáticos resuelven el problema clásico de estimación de parámetros, es decir, estiman un valor que es invariante en el tiempo (o al menos su variación es despreciable comparadas con sus variables de estado);

como el estimador de mínimos cuadrados o la regresión de mínimos cuadrados ponderada.

En nuestro caso nos centraremos en los estimadores lineales dinámicos, es decir, aquellos diseñados para resolver el problema de estimación de un estado, que varía en el tiempo, como por ejemplo:

- Filtro  $\alpha - \beta$
- Filtro  $\alpha - \beta - \gamma$
- Filtro de Kalman
- Filtro aumentado de Kalman
- Filtro extendido de Kalman
- Filtro Unscented de Kalman

A continuación, se presenta el compendio del conocimiento actual sobre filtros.

## 3.2 Filtro de Kalman

El filtro de Kalman desarrollado por Rudolf E. Kalman en 1960 sirve para estimar el estado de un sistema dinámico lineal en presencia de ruido gaussiano blanco. Es un algoritmo recursivo, es decir, se aplica continuamente, siendo la salida del filtro, su entrada para la próxima vez que se ejecute.

Está especialmente diseñado para trabajar en el espacio de estados, donde el algoritmo se modela a partir de las siguientes ecuaciones [49]:

Ecuación de la planta o ecuación de estado:

$$x_k = A \cdot x_{k-1} + B \cdot u_k + w_k$$

La matriz A o matriz de estado relaciona el estado actual con el estado anterior sin ninguna influencia externa, la matriz B, denominada matriz de entrada, refleja la influencia de las entradas externas.

Ecuación de medida:

$$z = H \cdot x_k + v_k$$

Donde la matriz H es la matriz de salida, que relaciona las variables de estado con las medidas, en el caso de que las variables de estado se pudieran medir, la matriz de salida sería la matriz identidad.

Como se puede apreciar ambas ecuaciones tienen ruido “v” y “w”, con matrices de covarianza “Q” y “R” respectivamente; por lo que estamos ante un sistema estocástico, en ausencia de ruido sería un sistema determinista.

$$p(v) \sim N(0, Q)$$

$$p(w) \sim N(0, R)$$

El filtro de Kalman es válido para un sistema que se pueda representar en el espacio de estados y que tenga las siguientes características:

1. dinámico (causal): la salida en el instante k es función de una entrada y/o salida pasada o presente.
2. Estocástico: durante el proceso influyen variables aleatorias que varían con el tiempo
3. lineal: el modelo matemático del sistema se basa en el uso de un operador lineal. Y el sistema se reducirá a n ecuaciones de primer grado.

Por otro lado, las propiedades de un sistema irrelevantes para el filtro de Kalman, ya que se podría aplicar independientemente de las características del sistema:

- Dependencia con el tiempo: el sistema puede ser variante o invariante en el tiempo, generalmente se asumen sistemas invariantes en el tiempo por simplicidad, en ese caso las matrices A, B y H del modelo de espacio de estados son constantes.
- Variable temporal: Según la variable temporal el sistema se puede caracterizar como continuo o discreto, y el filtro de Kalman se puede aplicar para ambos sistemas.

### Dualidad modelo funcional - estocástico

Para sistemas estocásticos existe una dualidad que queda evidente cuando aplicamos los operadores esperanza y dispersión a la ecuación del sistema.

$$y = A \cdot x + w$$

Siendo w, ruido gaussiano blanco.

Si aplicamos el operador esperanza al sistema obtendremos el modelo funcional del sistema, que es determinista,

$$E[y] = A \cdot x$$

Y si aplicamos el operador dispersión obtenemos el modelo estocástico del sistema, es decir, la matriz de covarianza.

$$D[y] = P = \begin{pmatrix} \sigma_{y_1}^2 & \sigma_{y_1 y_2} & \sigma_{y_1 y_3} \\ \sigma_{y_2 y_1} & \sigma_{y_2}^2 & \sigma_{y_2 y_3} \\ \sigma_{y_3 y_1} & \sigma_{y_3 y_2} & \sigma_{y_3}^2 \end{pmatrix}$$

La matriz de covarianza de un vector aleatorio X se define como la multiplicación de una matriz por su traspuesta (por lo que se podría expresar como una suma de cuadrados), lo que garantiza que es una matriz definida positiva.

$$P = E((X - E(X))^T \cdot (X - E(X)))$$

Una matriz es definida positiva si todos sus autovalores son positivos. Si la matriz de covarianza no es definida positiva se dice que es singular, lo que podría indicar que alguna de las variables es una combinación lineal de las otras. El filtro de Kalman es muy sensible a errores de truncamiento, y redondeo; esta falta de robustez numérica puede hacer que debido a errores numéricos se obtengan matrices de covarianza no definidas positivas, lo que puede afectar a la estabilidad del filtro. Si la matriz de covarianza se convierte por ejemplo en semi-definida positiva, quiere decir que uno de sus autovalores es cero, lo que indicaría que el filtro “cree” que ha estimado perfectamente algunas variables del estado, y por tanto la ganancia para esas variables será nula, lo cual no es deseable en un filtro.

Dado que de aquí en adelante vamos a trabajar con sistemas estocásticos conviene conocer más a fondo la matriz de covarianzas. Es una matriz cuadrada de tamaño igual al número de variables de estado, cuya diagonal contiene las varianzas de las variables de estado, la varianza de una variable indica la incertidumbre y es el cuadrado de la desviación típica.

$$\text{varianza} = \sigma^2$$

Siendo  $\sigma$  la desviación típica.

El resto de elementos de la matriz contienen las covarianzas, que indican como las variables de estado co-varian entre ellas, es decir, el grado de variación conjunta de dos variables aleatorias, y nos permite determinar si existe una dependencia entre las variables. Por ejemplo, imagine el lector un avión cuyo vector de estado es su posición horizontal “x” e “y”. Su matriz de covarianza, P, que será de tamaño 2x2 puede tener las siguientes formas:

- matriz diagonal con los mismos valores en la diagonal:

$$P = \begin{pmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{pmatrix}$$

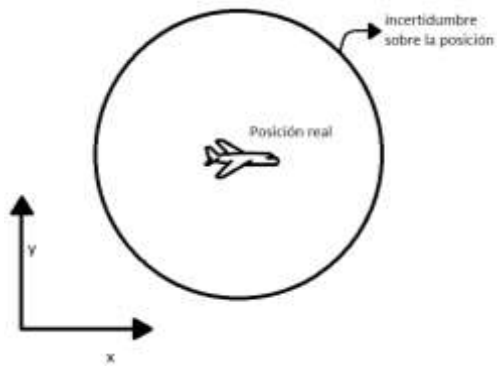


Ilustración 117. Incertidumbre según una matriz de covarianza diagonal con los mismos valores en la diagonal

- matriz diagonal con diferentes valores en la diagonal:

$$P = \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix}; \sigma_x^2 > \sigma_y^2$$

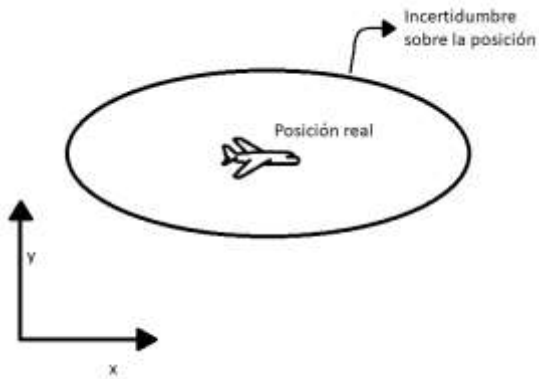


Ilustración 118. Incertidumbre según una matriz de covarianza diagonal

- matriz no diagonal, es decir, con covarianzas no nulas:

$$P = \begin{pmatrix} \sigma^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma^2 \end{pmatrix}$$

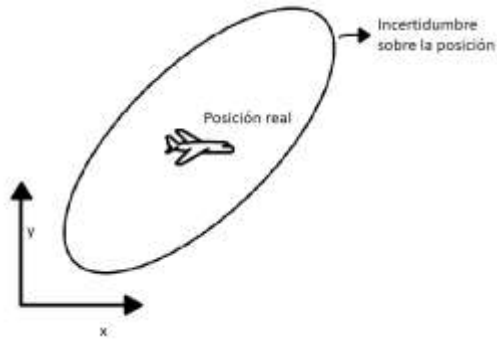


Ilustración 119. Incertidumbre según una matriz de covarianza no diagonal

Dado que la covarianza de la variable de estado 1, respecto de la 2 es igual a la covarianza de la variable de estado 2 respecto de la 1, la matriz de covarianzas es simétrica.

$$D[y] = \begin{pmatrix} \sigma_{y_1}^2 & \sigma_{y_1 y_2} & \sigma_{y_1 y_3} \\ \sigma_{y_1 y_2} & \sigma_{y_2}^2 & \sigma_{y_2 y_3} \\ \sigma_{y_1 y_3} & \sigma_{y_2 y_3} & \sigma_{y_3}^2 \end{pmatrix}$$

Y generalmente despreciables, para nuestro caso serán nulas:

$$D[y] = \begin{pmatrix} \sigma_{y_1}^2 & 0 & 0 \\ 0 & \sigma_{y_2}^2 & 0 \\ 0 & 0 & \sigma_{y_3}^2 \end{pmatrix}$$

## Descripción del algoritmo

El filtro de Kalman se basa en dos pasos:

1. Predicción: este paso también se conoce como estimación a priori, es decir, sin conocer la observación. En esta fase se predice el vector de estado para la siguiente época según la ecuación de estado, que modela nuestro sistema.

$$x_{kp} = A x_{k-1} + B u_k + w_k$$

Además, también se determina la matriz de covarianza del vector de estado predicho propagando la matriz de covarianza anterior según la ecuación de Lyapunov.

$$P_{kp} = A P_{k-1} A^T + Q_k$$

2. Corrección: Una vez que tenemos el vector de estado predicho y su incertidumbre (matriz de covarianza predicha), se procede a corregir la predicción con la medida/observación. Para ello se promedia la predicción y la medida según la ganancia de Kalman

$$K = \frac{\text{Incertidumbre estimación}}{\text{Incertidumbre estimación} + \text{Incertidumbre medición}}$$

$$K = \frac{P_{kp} \cdot H^T}{H \cdot P_{kp} \cdot H^T + R}$$

El vector de estado estimado es:

$$x_k = x_{kp} + K \cdot [z_k - H \cdot x_{kp}]$$

Que es la solución óptima del estimador MMSE, por lo que produce el mínimo error cuadrático medio.

Y la matriz de covarianza asociada a la estimación es:

$$P_k = (I - K \cdot H) \cdot P_{kp}$$

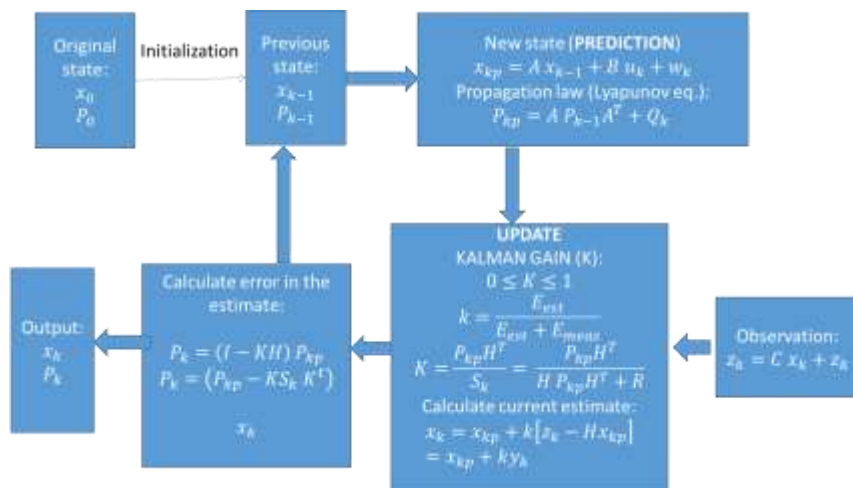


Ilustración 120. Diagrama de flujo del filtro de Kalman

Hipótesis:

- El ruido del proceso es blanco.
- El ruido de la medida es blanco.
- Las dos secuencias de ruido no están correladas.

Ventajas:

- Es la técnica estadística adecuada, ya que es el filtro de estimación óptimo: estimador insesgado y de varianza mínima.
- El filtro no solo proporciona una estimación, sino que también proporciona información sobre la calidad de la estimación (matriz de covarianza).
- Filtro recursivo de primer orden: solo es necesario retroalimentar al filtro con el último estado (otros filtros requieren estados más antiguos, aumentando la memoria necesaria)
- Ejecución en tiempo real
- Formato matricial
- Tiempo discreto

Las últimas cuatro características lo hacen ideal para aplicaciones embarcadas.

Desventajas:

- Solo aplicables a sistemas lineales
- Requiere condiciones iniciales: necesitan una estimación inicial (media y covarianza, ya que es un filtro recursivo)
- Estabilidad/convergencia: es necesario siempre analizar la estabilidad del filtro.

## Estabilidad

Para estudiar la estabilidad del filtro debemos conocer primero los conceptos de controlabilidad y observabilidad.

La estabilidad se define como la convergencia de la matriz de covarianza a una matriz estacionaria finita, es decir, el error se hace estacionario.

## Controlabilidad

Un sistema es completamente controlable, si dado un punto arbitrario en el espacio de estados, existe una función de entrada que puede llevar al sistema a un espacio determinado en un tiempo finito. Un sistema es controlable si su matriz de controlabilidad tiene rango máximo.

$$Q_c = [B \ AB \ \dots \ A^{n_x-1}B]$$

Siendo A la matriz de estado, B la matriz de entrada, y  $n_x$  el número de variables de estado.

La condición de controlabilidad evita que la matriz de covarianza converja a cero [Sección 5.25. de Bar Shalom]

## Observabilidad

Un sistema es observable si su estado inicial se puede determinar unívocamente, conociendo su salida por un intervalo de tiempo finito, y la entrada del sistema. Un sistema es observable si el rango de la matriz de observabilidad es máximo.

$$Q_o = \begin{bmatrix} C \\ CA \\ \dots \\ CA^{n_x-1} \end{bmatrix}$$

Siendo C la matriz de la ecuación de medida, A la matriz de estado, y  $n_x$  el número de variables de estado.

La condición de observabilidad garantiza que la incertidumbre no diverja, y que haya una matriz de covarianza estacionaria, definida o semi-definida positiva.

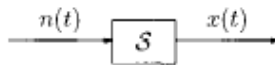
Nótese que la estabilidad del filtro no requiere que el sistema sea estable (solo requiere la condición de observabilidad y controlabilidad), esto es muy importante en los algoritmos de seguimiento de aeronaves, los modelos son inestables, ya que tienen un integrador (para obtener la posición a partir de la velocidad).

## 3.2 Filtro de Kalman Aumentado

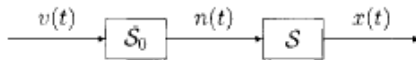
Cuando no podemos asumir un modelo de ruido blanco, pero se puede asumir un ruido estacionario (sus características permanecen constantes con el tiempo) no blanco, es decir, el ruido está autocorrelado (ruido de color),

entonces no se puede aplicar el filtro de Kalman, ya que el vector de estado no sigue un proceso de Markov.

Hace falta “pre-blanquear” (“pre-whitening”) el sistema, esto se hace añadiendo otro filtro imaginario,  $S_0$ , cuya entrada es ruido blanco ( $v(t)$ ), y cuya salida es el ruido de color ( $n(t)$ ), como se explica en sección 1.4 de [21].



**Ilustración 121. Sistema afectado por ruido no blanco  $n(t)$**



**Ilustración 122. Sistema aumentado gobernado por ruido blanco  $v(t)$**

Este nuevo ruido es otra variable de estado, de forma que el nuevo vector de estado si es un proceso de Markov.

$$\bar{x} = \begin{pmatrix} x(t) \\ n(t) \end{pmatrix}$$

Por lo que finalmente solo quedaría tener un modelo para la nueva variable de estado. Para un caso unidimensional donde el vector de estado sea la posición y la velocidad, entonces el vector de estado aumentado incluiría el ruido autocorrelado también, generalmente representado por la letra “ $\ddot{x}$ ”, por ser tratado como una aceleración en los modelos cinemáticos:

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} x \\ \dot{x} \\ \ddot{x} \end{pmatrix}$$

La ecuación del filtro que “blanquea” nuestro sistema es un modelo autoregresivo de primer orden con factor de autocorrelación  $\alpha$ :

$$x_3 = -\alpha \cdot x_3(t) + v(t)$$

Por lo que la ecuación de estado quedaría como:

$$\begin{pmatrix} \dot{x} \\ \ddot{x} \\ \ddot{\ddot{x}} \end{pmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\alpha \end{bmatrix} \begin{pmatrix} x \\ \dot{x} \\ \ddot{x} \end{pmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} v$$

### 3.3 Filtro de Kalman Extendido

Cuando el sistema no es lineal, el filtro de Kalman no se puede aplicar, ya que una de las hipótesis del filtro de Kalman era que al propagar una distribución gaussiana por la ecuación de estado o de medida, la salida seguiría siendo una distribución gaussiana representada por su media y su covarianza; sin embargo para ecuaciones no lineales esto no es así:

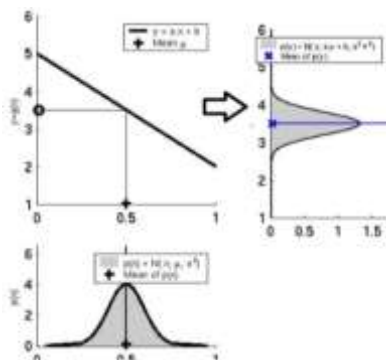
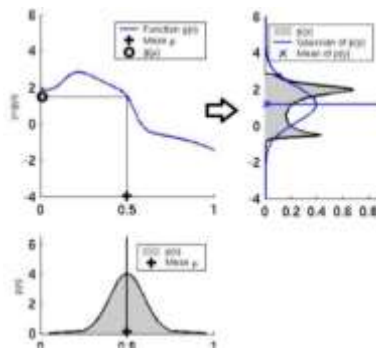


Ilustración 123. Propagación de una distribución gaussiana a través de una ecuación lineal



**Ilustración 124. Propagación de una distribución gaussiana a través de una ecuación no lineal**

N.B. la no linealidad puede estar presente en cualquiera de las ecuaciones del espacio de estados, i.e. ecuación del proceso o de la medida o ambas.

Por tanto debemos recurrir a un filtro para sistemas no lineales, en esta sección estudiaremos el filtro extendido de Kalman o EKF por sus siglas en inglés (Extended Kalman Filter)

El primer paso es identificar la o las ecuaciones que no son lineales, para proceder a su linealización en torno a un punto de trabajo a través de su desarrollo en serie de Taylor:

$$f = f(a) + \frac{f'(a)}{1!}(x - a) + \frac{f''(a)}{2!}(x - a)^2 + \frac{f'''(a)}{3!}(x - a)^3 + \dots$$

El punto de trabajo elegido es la media, y solo se utiliza el primer término del desarrollo de Taylor, ya que el segundo término es cuadrático (no lineal). Por tanto, la aproximación solo es válida para un pequeño rango alrededor de la media. Y como veremos más adelante esto es uno de los principales inconvenientes de este filtro.

En consecuencia, las nuevas matrices de estado y de observación serán las matrices jacobianas evaluadas en el punto donde hemos linealizado (la media de las variables de estado):

$$x_k = f(x_{k-1}, u_k) + w_k$$

$$z_k = g(x_k) + v_k$$

$$A = \left( \frac{\delta f}{\delta x} \right)_{\hat{x}_{k-1}, u_k}$$

$$H = \left( \frac{\delta g}{\delta x} \right)_{\hat{x}_k}$$

La mayoría de los problemas en ingeniería requieren trabajar con ángulos, por ejemplo si se procesan coordenadas polares, como es el caso de ecuaciones radar (distancia oblicua y azimut). Como veremos a continuación estas ecuaciones tienen funciones trigonométricas, que son no lineales, por lo que hay que linealizarlas.

Las ecuaciones radar, son las ecuaciones que transforman las coordenadas polares a cartesianas:

$$x = r \cdot \cos(\theta)$$

$$y = r \cdot \sin(\theta)$$

Sin embargo, la ecuación de medida (según el modelo de espacio de estados) expresa las medidas (coordenadas polares) en función de las variables de estado (coordenadas cartesianas), por lo que necesitamos las ecuaciones inversas:

$$r = \sqrt{x^2 + y^2}$$

$$\theta = \arctan\left(\frac{y}{x}\right)$$

Finalmente, la matriz Jacobiana se obtiene calculando las derivadas parciales de las ecuaciones respecto de cada una de las variables de estado:

$$H = \begin{pmatrix} \frac{\delta r}{\delta x} & \frac{\delta r}{\delta y} \\ \frac{\delta \theta}{\delta x} & \frac{\delta \theta}{\delta y} \end{pmatrix} = \begin{pmatrix} \frac{x}{\sqrt{x^2 + y^2}} & \frac{y}{\sqrt{x^2 + y^2}} \\ -\frac{y}{x^2 + y^2} & \frac{x}{x^2 + y^2} \end{pmatrix}$$

### 3.4 Filtro de Kalman Unscented

Una vez que hemos visto el filtro de Kalman, y sus hipótesis de trabajo de que los datos siguen una distribución gaussiana, y como estas distribuciones se propagan a través de ecuaciones lineales; nos dimos cuenta que en el mundo real la mayoría de ecuaciones son no lineales, por lo que también hemos visto el filtro extendido de Kalman, que linealizaba las ecuaciones, a través de desarrollos en serie de Taylor, en torno a un punto (generalmente la media de la gaussiana).

En 1997 Julier y Uhlmann propusieron el filtro de Kalman Unscented o UKF (“Unscented Kalman Filter”) [50]. La causa de su aparición es su desempeño. En el filtro EKF al utilizar solo un punto para linealizar la ecuación, la gaussiana aproximada no se parece a la distribución real, mientras que el filtro UKF utiliza varios puntos. Por lo que el filtro UKF tiene un mejor desempeño que el EKF sobre todo para sistemas fuertemente no lineales. El filtro UKF tiene mejor exactitud que el EKF para un mismo nivel de complejidad en varios problemas de estimación (estimación en el espacio de estados, y estimación de parámetros) como se detalla en [17].

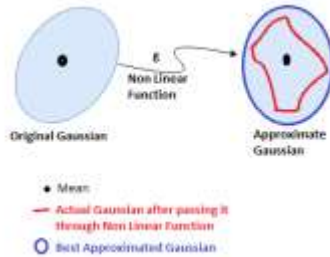


Ilustración 125 Gaussiana aproximada en filtro EKF alrededor de la media

En el filtro UKF se seleccionan un conjunto de puntos, denominados puntos sigma, entre los que se debe encontrar la media. La cantidad de puntos sigma depende de la exactitud de la aproximación que estemos buscando y el coste computacional disponible, a mayor número de puntos sigma, mayor exactitud.

Estos puntos sigma seleccionados de la distribución gaussiana original, se propagan a través de la ecuación no lineal, a esto se le conoce como transformación UT (“Unscented Transform”), y entonces se calcula la nueva media y varianza de la gaussiana transformada.

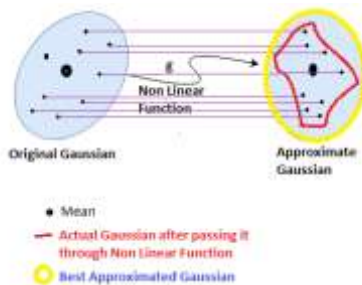


Ilustración 126 Gaussiana aproximada en filtro UKF alrededor de los puntos sigma

Para afinar aún más la aproximación, cada punto sigma recibe un peso ( $w$ ).

Por tanto, los pasos del filtro UKF para propagar una distribución gaussiana a través de una función no lineal son:

1. Selección de puntos sigma
2. Asignación de pesos a los puntos sigma
3. Propagar los puntos sigma a través de la ecuación no lineal
4. Determinar la gaussiana resultante a través de los puntos sigma propagados/transformados y sus pesos
5. Determinar la media y varianza de la nueva gaussiana

La selección de los puntos sigma (tanto en cantidad como en localización) es un paso muy importante, ya que estos deben ser representativos de la distribución gaussiana original.

Respecto al número total de puntos sigma a utilizar depende del número de dimensiones del sistema ( $n$ ):

$$\text{número de puntos sigma} = 2 \cdot n + 1$$

Y sus respectivas localizaciones, que dependen del número de dimensiones ( $n$ ), el factor de escala ( $\lambda$ ), la media ( $\mu$ ), y la matriz de covarianza ( $\Sigma$ ), viene dada por las siguientes formulas:

$$\chi_0 = \mu$$

$$\chi_i = \mu + \sqrt{(n + \lambda)\Sigma}; \text{ para } i = 1, \dots, n$$

$$\chi_i = \mu - \sqrt{(n + \lambda)\Sigma}; \text{ para } i = n + 1, \dots, 2n$$

La matriz de los puntos sigma ( $\chi$ ) recoge los puntos sigma para cada variable de estado en cada columna, por tanto sus dimensiones serán  $(2n+1) \times n$ .

columnas, una por variable de estado, y  $2n+1$  filas, que son la cantidad de puntos sigma por variable de estado.

Cabe destacar que en la fórmula de la matriz de puntos sigma aparece la raíz cuadrada de una matriz, para determinar la raíz cuadrada de una matriz se pueden utilizar distintos métodos algebraicos como la descomposición de Cholesky, sección 2.9 de [25], o mejor aún si la matriz es diagonal, su raíz cuadrada será otra matriz diagonal cuyos elementos en la diagonal son la raíz cuadrada de la diagonal anterior, como se explica en la sección 18.1 de [25].

$$A = B * B^T \rightarrow B = \sqrt{A}$$

El factor de escala ( $\lambda$ ) establece la lejanía de los puntos sigma de la media, como veremos más adelante en los MOPS de ACAS X a este valor se le denomina inflación de la covarianza inicial, y se determina para garantizar que las matrices del filtro sean semi-definidas positivas.

Respecto a los pesos de los puntos sigma (puntos sigma ponderados), se utilizan dos ecuaciones, una para el peso de la media, que será el punto sigma más importante, y otra para el resto. Obviamente la suma de todos los pesos es igual a 1.

$$w_0 = \frac{\lambda}{n + \lambda}$$

$$w_i = \frac{1}{2(n + \lambda)} ; \text{para } i = 1, \dots, 2n$$

Más tarde se realiza la reconstrucción paramétrica de la gaussiana después de la transformación no lineal, que se calcula con las siguientes ecuaciones:

$$\mu' = \sum_{i=0}^{2n} w_i \cdot g(x_i)$$

$$\Sigma' = \sum_{i=0}^{2n} w_i \cdot (g(\chi_i) - \mu') \cdot (g(\chi_i) - \mu')^T$$

Finalmente, al igual que en todo filtro de Kalman se aplican los pasos de predicción y corrección; y dependiendo si la ecuación no lineal es la de estado, o la de la observación, o las dos, se aplicarán los puntos sigma a una ecuación o la otra respectivamente.

Si la ecuación de estado no es lineal, a la hora de predecir el proceso, la distribución predicha tendrá los siguientes parámetros (mismas ecuaciones que las anteriores pero teniendo en cuenta el ruido del proceso Q, ya que siempre que se predice, la incertidumbre aumenta):

$$\mu' = \sum_{i=0}^{2n} w_i \cdot g(\chi_i)$$

$$\Sigma' = \sum_{i=0}^{2n} w_i \cdot (g(\chi_i) - \mu') \cdot (g(\chi_i) - \mu')^T + Q$$

Para el paso de corrección, se pueden determinar de nuevo los puntos sigma, o reutilizar los puntos sigma de la etapa de predicción (en el caso de que la ecuación de estado no sea lineal y hayamos tenido que determinarlos).

Una vez que tenemos los puntos sigma ( $\chi$ ), los transformamos a las coordenadas de la observación:

$$Z = h(\chi)$$

La media y covarianza de esos puntos en el espacio de estado de la medida será:

$$\hat{z} = \sum_{i=0}^{2n} w_i Z_i$$

$$S = \sum_{i=0}^{2n} w_i \cdot (Z_i - \hat{z}) \cdot (Z_i - \hat{z})^T + R$$

Donde:

H: función no lineal, que mapea los puntos sigma del espacio de estados al espacio de medidas.

$\chi$ : puntos sigma.

Z: puntos sigma transformados al espacio de medidas.

$\hat{z}$ : Media en el espacio de medidas

S: matriz de covarianza en el espacio de medidas

R: matriz de covarianza de la ecuación de medida

Para terminar, nos queda calcular la ganancia de Kalman, a diferencia con respecto el EKF, ahora no tenemos la matriz Jacobiana para determinar la ganancia de Kalman, por lo que primero tenemos que calcular el error en la predicción como la correlación cruzada entre los puntos sigma en el espacio de estados y los puntos sigma en el espacio de medida.

$$T = \sum_{i=0}^{2n} w_i \cdot (\chi_i - \mu') \cdot (Z_i - \hat{z})$$

Por tanto la ganancia de Kalman será:

$$K = T \cdot S^{-1}$$

## Anexo B: Método de Monte Carlo

El germen del método de Monte Carlo se remonta al año 1733, cuando el científico Buffon planteó el siguiente problema, si se lanza una aguja sobre un papel en el que se han trazado rectas paralelas distanciadas entre sí de manera uniforme una distancia igual a la longitud de la aguja ¿cuál es la probabilidad de que la aguja cruce alguna de las líneas?, la respuesta es  $2/\pi$ , su experimento se considera el primer experimento de una simulación de Monte Carlo, como se detalla en [51].



Ilustración 127. Experimento de la aguja de Buffon

Para sistemas sencillos (lineales, e invariantes en el tiempo) que se pueden analizar teóricamente, primero se suelen modelar con una ecuación diferencial (tiempo continuo) o una ecuación en diferencias (tiempo discreto), a esto se le conoce como modelo de caja blanca o “White box”, ya que conocemos las ecuaciones que gobiernan el sistema; y a partir de ahí se obtiene la función de transferencia, ya sea por la transformada de Laplace o la transformada Z.

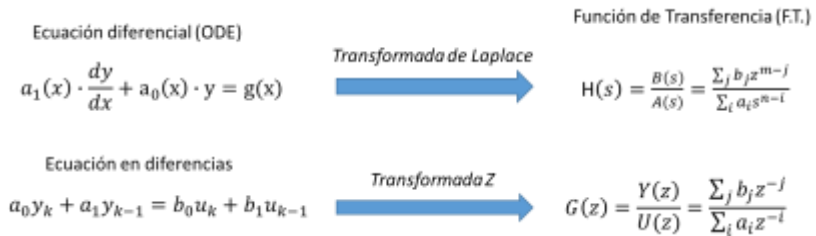


Ilustración 128. análisis teórico de un sistema

Sin embargo, para sistemas más complejos necesitamos analizarlos mediante el método de Monte Carlo.

El método de Monte Carlo es un método no determinista, usado para analizar expresiones matemáticas complejas que generalmente no se pueden evaluar teóricamente, para ello se genera una gran cantidad de números aleatorios, que en nuestro caso se utilizarán como medidas simuladas, y a partir de ahí se determinarán las probabilidades de que una determinada variable esté en una determinada región. Por lo que estudiando los resultados de la simulación de Monte Carlo podemos determinar la función de distribución de probabilidad o PDF (“Probability Density Function”) de la variable de interés. En definitiva, el método de Monte Carlo se utiliza para determinar probabilidades, por lo que no es otra cosa que un método de integración.

Por otro lado, la simulación es la mejor opción para evaluar el desempeño de los algoritmos de estimación, ya que necesitamos conocer el error entre las variables de estado estimadas frente al valor real, y eso solo se puede conocer mediante la simulación.

Finalmente, si aplicamos el método de Monte Carlo y los métodos de simulación obtendremos la simulación de Monte Carlo, que es una simulación no determinista (dos ejecuciones diferentes darán resultados diferentes), donde es importante definir:

- Variables en las que se modela el ruido (capítulo 4.2)
- Generador de números aleatorios utilizado (capítulo 4.1)
- Número de ejecuciones de Monte Carlo: cada encuentro entre aeronaves se simula varias veces; ya que, al ser un método aproximado no determinista, se necesitan varias iteraciones para mejorar la exactitud del resultado, y para que se cumpla el teorema del límite central visto en el capítulo 3. En el capítulo 7: Resultados, se define el número de ejecuciones de Monte Carlo por encuentro y se comprueba su impacto en el resultado final.
- Variables a estudiar: Las variables a analizar serán la trayectoria estimada frente a la trayectoria real (“groundtruth”).

### **Ventajas y Desventajas del método de Monte Carlo**

#### Ventajas:

1. Ofrece resultados gráficos
2. Funciona para sistemas no lineales
3. Análisis de sensibilidad: permite analizar que variables tienen un mayor impacto en el sistema.
4. Análisis en profundidad del escenario: si hay alguna ejecución de Monte Carlo que tiene valores que se diferencian de lo “normal”, el método de Monte Carlo permite hacer un análisis profundo, analizando las entradas concretas del algoritmo durante esa ejecución.
5. Correlación de variables de entrada: permite analizar las relaciones entre diferentes variables de entrada.

#### Desventajas:

1. Resuelve el problema mediante una aproximación
2. Diseño de la simulación complicado: definir las variables que se modelan, diseñar modelos de ruido, selección de generador de números aleatorios.
3. Recursos.
4. Tiempo de procesamiento/velocidad.

5. Cada simulación es única, ya que interviene el azar: necesidad de almacenamiento de datos para garantizar la reproducibilidad de los experimentos.