

UNIVERSIDAD NACIONAL DE CÓRDOBA
INSTITUTO UNIVERSITARIO AERONÁUTICO

TESIS DE MAESTRÍA

Sistema de Seguimiento y Generación de Pistas para Radar Track While Scan

Autor:
Santiago A. RODRIGUEZ
GONZALEZ

Director:
Ing. Jorge MUÑOZ

Co-Director:
Dr. Giorgio CARANTI

Maestría en Sistemas de Radar e Instrumentación

Facultad de Matemática, Astronomía y Física
Grupo Radar Córdoba

Abril de 2019



Sistema de Seguimiento y Generación de Pistas para Radar Track While Scan por Santiago Andrés Rodríguez González se distribuye bajo una Licencia Creative Commons Atribución-CompartirIgual 4.0 Internacional .

UNIVERSIDAD NACIONAL DE CÓRDOBA
INSTITUTO UNIVERSITARIO AERONÁUTICO

Resumen

Facultad de Matemática, Astronomía y Física
Grupo Radar Córdoba

Maestría en Sistemas de Radar e Instrumentación

Sistema de Seguimiento y Generación de Pistas para Radar Track While Scan

Santiago A. RODRIGUEZ GONZALEZ

En el siguiente trabajo se presenta el desarrollo de un sistema de software Generador de Pistas para un Radar de tipo *Track While Scan*, de tiempo real. El Generador de Pistas (*tracker*) correlaciona las detecciones de un radar (*plots*) en cada barrido para generar pistas (*tracks*) con información de identidad, velocidad, rumbo e historia de la trayectoria que siguen las aeronaves detectadas. La información de salida es presentada en el standard ASTERIX Eurocontrol. El sistema se encuentra actualmente operativo en parte de la flota de radares de la Fuerza Aérea Argentina (FAA).

Se hace una descripción de los radares sobre los que se trabajará, de los algoritmos involucrados, y se muestran resultados simulados y reales a partir de datos de dos tipos diferentes de radar. Uno de ellos es un radar de seguimiento que tiene Fuerza Aérea, el Vitro RIR 778C, el cual ha vuelto a funcionar recientemente. El otro es el Radar de vigilancia Bendix AN/FPS-113. Se muestra la implementación en hardware, con datos de performance de funcionamiento en operación.

Agradecimientos

En primer lugar, quiero agradecer al Comodoro Ing. Jorge Muñoz, quien me introdujo al tema y me permitió trabajar por primera vez en mi vida con radares, incorporándome al equipo de trabajo del Radar Vitro y al EDDR posteriormente.

Al Ing. Luis Murgio, quien siempre me acompañó y enseñó sobre como transitar este complicado mundo del desarrollo tecnológico en Argentina.

A la Fuerza Aérea Argentina en general, que me dio acceso a radares y datos reales, y confió en mi capacidad de poder administrar proyectos relacionados.

Al Ing. Ignacio Montamat, con quien siempre he trabajado como un hermano, en el proyecto del Combinador-Trackeador del EDDR.

Al equipo completo de trabajo del PIDDEF 24/14, relacionado al radar Vitro, por la dedicación y entusiasmo que han tenido a pesar de toda la adversidad, Ing. Pablo Morales, Ing. Mauro Gomez, Ing. Federico Prystupczuk, Ing. Fabio Nazzi.

A los docentes de la Maestría de Sistemas de Radar en Instrumentación, especialmente Dr. Raúl Comes por su constante interés y pasión para resolver problemas.

A mi viejo, que me inició e inspiró en el mundo de la ingeniería, y al día de hoy sigue siendo un pilar en lo personal y en lo profesional.

A mi familia, por acompañarme y apoyarme, aunque a veces les sea complicado entender los detalles de mi trabajo.

Índice general

Resumen	I
Agradecimientos	II
1. Introducción	1
1.1. Marco Histórico - Bendix FPS-113	1
1.1.1. Especificaciones	2
1.1.2. AN/FPS-113 en la Argentina	2
1.2. Antecedentes	3
1.3. Objetivos	5
2. Marco Teórico	6
2.1. Radar	6
2.1.1. Radar Primario	6
2.1.2. Radar Secundario	6
2.2. Sistemas de Seguimiento	7
2.2.1. Sistemas de Vigilancia	8
2.3. Tracking	8
2.3.1. Enfoque de Seguimiento Bayesiano	9
2.3.2. Filtro de Kalman Lineal	10
2.4. Asociación de Datos	11
2.4.1. Ventanas de Detección	14
2.4.2. Nearest Neighbour Standard Filter	16
2.4.3. Global Nearest Neighbour	17
Algoritmo Húngaro	17
3. Desarrollo	20
3.1. Extractor Digital de Datos de Radar-EDDR	20
3.1.1. Combinador	20
3.2. Trackeador y Generador de Pistas	21
3.2.1. Asociación de Detecciones a Pistas	21
3.2.2. Iniciación de Pistas	24
3.2.3. Filtrado de Pistas	24
3.2.4. Cierre de Pistas	28
3.2.5. Implementación del Algoritmo	28
Caso Tracking vuelta a vuelta	28
Caso Track While Scan	30
3.3. Simulador	30
3.3.1. Simulador por Hardware	30
3.3.2. Simulador por Software	32
3.4. Implementación en Hardware	35

4. Resultados	44
4.1. Filtro de Kalman	44
4.1.1. Trayectoria simulada sin maniobra en 2D	44
4.1.2. Trayectoria simulada con maniobra en 2D	49
4.1.3. Trayectoria real sin maniobra en 2D	49
4.1.4. Trayectoria real sin maniobra en 3D	54
4.1.5. Trayectoria real con maniobra en 3D	56
4.2. Asociación de Datos	62
4.2.1. Múltiples trayectorias simuladas en 2D	62
4.2.2. Trayectoria simulada en 2D con pérdida de datos	64
4.2.3. Trayectoria simulada en 2D con clutter	64
4.2.4. Trayectorias reales en 2D	66
4.3. Performance Computacional	67
4.3.1. Uso de Memoria	67
4.3.2. Uso de CPU	68
5. Conclusiones	72
Referencias	73

Índice de figuras

1.1. Radar Bendix AN/FPS-20	2
1.2. Centro de Vigilancia Aeroespacial Resistencia	3
1.3. Centro de Vigilancia Aeroespacial Posadas	4
1.4. Radar AN/FPS-113-Interior	4
2.1. Radar AN/FPS-35	7
2.2. Síntesis Proceso de Estimación	8
2.3. Síntesis Proceso de <i>Tracking</i>	9
2.4. Sistema de <i>Tracking</i>	9
2.5. Ecuaciones Filtro de Kalman	12
2.6. Radar Vitro RIR-778C	13
2.7. Radar INVAP RPA	14
2.8. Ejemplo Ventana de Detección	15
2.9. Ejemplo Ventana de Detección Compleja	15
2.10. Distancia Euclidiana y de Mahalanobis	16
2.11. NNSF vs GNN	17
3.1. Hardware Subrack EDDR	21
3.2. Hardware Rack EDDR	22
3.3. Síntesis Diagrama EDDR	23
3.4. Proceso de Combinación Dividido en Sectores	23
3.5. Ciclo de Tracking	24
3.6. Conversión Polares-Cartesianas	27
3.7. Algoritmo Tracking	29
3.8. Simulador por Hardware	31
3.9. Salida Simulador por Hardware	31
3.10. Simulador por Software	32
3.11. Trayectorias Simulador	33
3.12. Simulador con incertidumbre de medición	34
3.13. Trayectorias Muestreadas por Radar Simulado	34
3.14. Detecciones con simulador	35
3.15. Diagrama de bloques Simulador	36
3.16. Ruido de medición del radar en el simulador	37
3.17. Single Board Computer del EDDR	38
3.18. Diagrama de Clases del Software Combinador-Trackeador	39
3.19. Diagrama de Secuencia Recepción <i>Plot</i> Primario	41
3.20. Diagrama de Secuencia Recepción <i>Plot</i> Secundario	42
3.21. Diagrama de Secuencia Recepción TOP NORTE	43
4.1. Única trayectoria generada	45
4.2. Única trayectoria filtrada	46
4.3. Distancia Euclidiana entre detecciones y posición predicha	46
4.4. Distancia Estadística entre detecciones y posición predicha	47

4.5. Error de Estimación	48
4.6. Velocidad Estimada	48
4.7. Trayectoria Simulada con Maniobra	49
4.8. Trayectoria con Maniobra Filtrada	50
4.9. Error de seguimiento de trayectoria con maniobra	50
4.10. Ruta Seguida por Aerolínea Sao Pablo-Chile	51
4.11. Trayectoria Filtrada Boeing 737	52
4.12. Trayectoria Filtrada Boeing 737. Zoom Inicio	52
4.13. Trayectoria Filtrada Boeing 737. Zoom Final	53
4.14. Velocidad Estimada Boeing 737.	53
4.15. Trayectoria Boeing 737 en los planos X-Z	55
4.16. Trayectoria Boeing 737 en 3D	56
4.17. Trayectoria Boeing 737 sobre mapa	57
4.18. Detecciones Grob G-120 TP	58
4.19. Inicio trayectoria Grob G-120 TP en Cartesianas	58
4.20. Zona media trayectoria Grob G-120 TP en Cartesianas	59
4.21. Pista Filtrada en Esféricas del Grob G-120 TP	61
4.22. Componentes Esféricas Detecciones y Estimaciones Grob	61
4.23. Vuelo del Grob en 3 Dimensiones	62
4.24. Vuelo del Grob en 3 Dimensiones sobre el mapa	63
4.25. Trayectorias Simuladas y Pistas Generadas con Asociación de Datos	64
4.26. Trayectoria Simulada con Pérdida de Datos	65
4.27. Trayectoria Simulada con Clutter	65
4.28. Trayectorias Reales con poco Clutter	66
4.29. Trayectoria Real en TWS	67
4.30. Trayectorias Reales en situación de mucho <i>clutter</i>	68
4.31. Uso de Memoria de las pistas	69
4.32. Callstack Ingreso Pistas	69
4.33. Uso de Memoria de las detecciones	70
4.34. Callstack Ingreso Detección	70

Capítulo 1

Introducción

La Fuerza Aérea Argentina (FAA) cuenta con una flota heterogénea de radares de distinta antigüedad, tipo y procedencia. En su conjunto, integran el Grupo Vigilancia y Control del Espacio Aéreo, creado en 1952, dependiente del Comando de Adiestramiento y Alistamiento. Como su nombre lo indica, su objetivo es el control del espacio aéreo, principalmente en las fronteras del país.

Hasta tener una red de radares moderna y que cubra la totalidad del territorio nacional, la Fuerza Aérea cubre con los recursos disponibles, áreas denominadas estratégicas. Para ello, ha mantenido y actualizado sus antiguos radares para lograr mantenerlos operativos.

En el año 2007 el Ejército del Aire Español donó a la FAA cuatro radares de vigilancia Bendix AN/FPS-113 con radar de altura AN/FPS-90, de los cuales dos quedan finalmente operativos en las bases de Posadas y Resistencia entre 2010 y 2011. El mismo año, la Dirección General de Fabricaciones Militares encarga a la empresa INVAP el desarrollo y puesta en servicio de un Radar Primario 3D de Largo Alcance. En 2014 comienzan a entrar en operación los prototipos y modelos de serie del RPA (Radar Primario Argentino). Este tipo de radar, es el primero en el país con tecnología digital desde su etapa de diseño (González, 2014).

Para adaptarse a las necesidades actuales, fue necesaria una actualización a los viejos radares Bendix donados por España. Surge así el proyecto FAS-1132: "Extractor Digital de Datos Radar FPS-113/FPS-90" (PIDDEF 006/08/FAA), nombrado en adelante por sus siglas EDDR, desarrollado por FAA. El EDDR le permite a los viejos radares disponer de una salida digital comparable a radares modernos. Digitalizar la información radar facilita su transmisión a centros de control, permitiendo la operación desatendida de los sitios radar. También permite el procesamiento automático de la información, con la posibilidad de agregar inteligencia a la información radar.

El post-procesamiento de la información del EDDR es realizado por un módulo denominado CombTrack, acrónimo de sus submódulos, Combinador y Trackeador. CombTrack permite llevar viejos radares analógicos a la categoría de radar Track While Scan (TWS).

Un radar de tipo TWS proporciona en tiempo real información de identidad, trayectoria y capacidad de maniobra de los objetivos detectados.

El presente trabajo se centra en el desarrollo del submódulo Trackeador de CombTrack.

1.1. Marco Histórico - Bendix FPS-113

El radar Bendix AN/FPS-113 es un radar de vigilancia en Banda L derivado del radar AN/FPS-20 (Figura 1.1), desarrollado por EEUU durante la década de 1950, a su vez, derivado del AN/FPS-3. La producción del AN/FPS-20 comenzó en 1956 y fue el principal elemento de vigilancia de la región continental de los Estados Unidos durante la década de 1960 (Winkler, 1997). Durante los años posteriores se le hicieron diversas actualizaciones, incluyendo medidas *anti-jam* (denominados AN/FPS-66 y AN/FPS-67). El AN/FPS-113 agregó un módulo MTI (*Moving Target Indicator*) digital, contra-medidas electrónicas, y

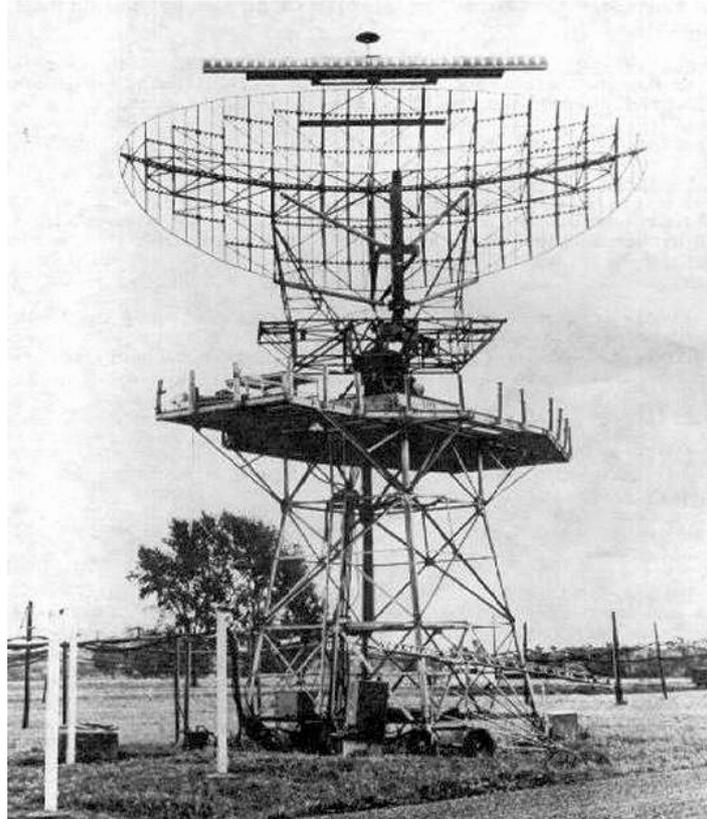


FIGURA 1.1: Radar Bendix AN/FPS-20 (Winkler, 1997)

diversas actualizaciones menores realizadas para el Ejército del Aire de España. La denominación FPS proviene de la *Joint Electronics Type Designation System* (JETDS), un sistema utilizado por el Departamento de Guerra de los Estados Unidos durante la Segunda Guerra Mundial para denominar al equipamiento electrónico. En 1957 JETDS se formalizó como **MIL-STD-196**. Las siglas indican, F - *Fixed Ground*, P - *Radar*, S - *Detecting or Range and Bearing, Search* (Defense, 2013). El prefijo AN originalmente indicaba *Aviation-Navy*, pero en la actualidad indica que el nombre pertenece al estándar JETDS.

1.1.1. Especificaciones

- Frecuencia: Banda L. 1250 a 1350 MHz.
- PRF: 350 Hz.
- Ancho de Pulso: 4 μ s
- Potencia: 2.5 MW pico
- Rango: 200 millas náuticas (370 Km)
- Ancho de Haz: 1 °

1.1.2. AN/FPS-113 en la Argentina

Los radares AN/FPS-113 que operan hoy en Argentina, provienen del Ejército del Aire de España. A mediados de 1957 el Ejército del Aire recibe sus primeros radares de vigilancia



FIGURA 1.2: Radar Bendix AN/FPS-113 en el Centro de Vigilancia Aeroespacial Resistencia

AN/FPS-20 a través del programa MAP (*US Military Assistance Program*), comenzando a operar en julio de 1958. En 1974 se actualizaron a la configuración AN/FPS-113 (Aeroplano, 2011). Prestaron servicio hasta el año 2007, cuando fueron reemplazados por los radares Lanza desarrollados por Indra. En ese mismo año el consejo de ministros de España autorizó la cesión de estos radares por el precio simbólico de un euro cada uno. En 2010, entra en funcionamiento en el Centro de Vigilancia Aeroespacial Resistencia (Figura 1.2), un edificio construido a medida para recibir el radar español. Un año después, se inaugura el Centro de Vigilancia Aeroespacial Posadas (Figura 1.3), con el segundo de los radares recientemente recibidos entrando en operación.

1.2. Antecedentes

Los antecedentes de este proyecto se remontan al año 2010, año en el que entran en operación en la Argentina los radares Bendix AN/FPS-113. Se hace necesario modernizar la forma en la que se muestran los datos, además de contar con datos radar en forma digital para que puedan ser accesibles desde el Centro de Vigilancia y Control Aeroespacial de la Fuerza Aérea Argentina, en la localidad de Merlo.

Con un objetivo concreto, se encarga el trabajo al Departamento de Radar del Centro de Investigaciones Aplicadas (CIA) de la Fuerza Aérea Argentina (FAA), grupo que ya había desarrollado trabajos similares para otros radares. El equipo desarrollado por FAA es denominado Extractor Digital de Datos de Radar (EDDR).

En el año 2013, ingenieros involucrados en la Maestría en Sistemas de Radar e Instrumentación, se incorporan al proyecto para actualizar parte del software que conforma al EDDR. Esta actualización se denomina CombTrack, y su primera parte (Combinador) ya ha sido publicada como Tesis de la mencionada maestría (Montamat, 2015).



FIGURA 1.3: Radar Bendix AN/FPS-113 en el Centro de Vigilancia Aeroespacial Posadas

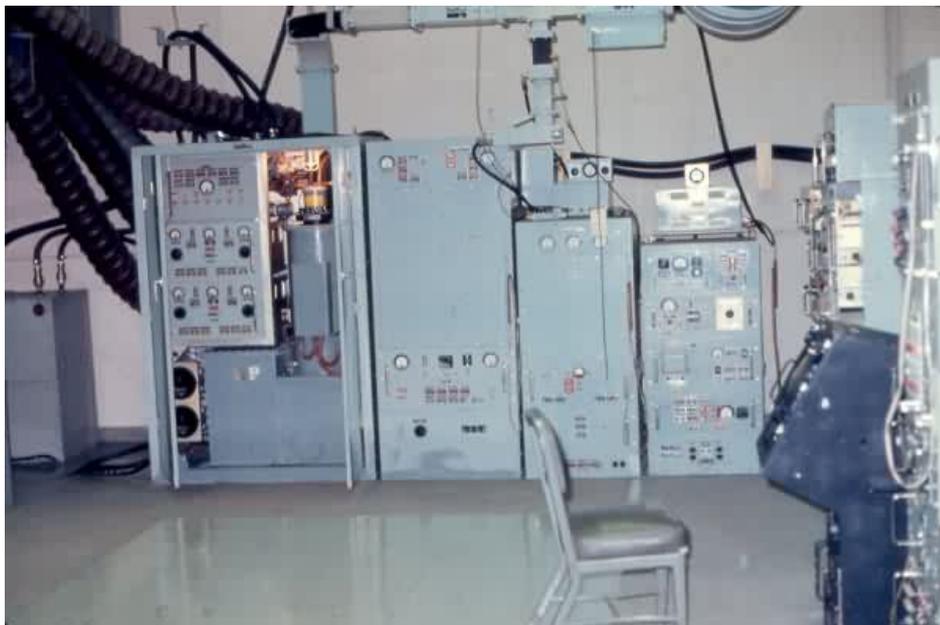


FIGURA 1.4: Transmisor y racks electrónicos del Radar Bendix AN/FPS-113

En el año 2014, este grupo de ingenieros comienza a asesorar al proyecto SINARAME (A. Rodríguez et al., 2017) (Sistema Nacional de Radares Meteorológicos), conformando el hoy denominado Grupo Radar Córdoba (Caranti et al., 2015).

1.3. Objetivos

El objetivo de este trabajo es desarrollar un medio para que los operadores de datos radar puedan contar con información adicional a los simples *plots* representados en las antiguas pantallas de fósforo.

Se especifica de esta manera el desarrollo de un sistema que tome detecciones combinadas de radar primario y secundario generando pistas que permitan,

- Identificar una secuencia de detecciones correspondiente a una misma trayectoria.
- Determinar la posición, velocidad y aceleración del objetivo detectado.
- Predecir la posición, velocidad y aceleración de la aeronave en momentos futuros.
- Discriminar detecciones de aeronaves de aquellas producidas por *clutter*.
- Continuar con un seguimiento ante detecciones faltantes.

El sistema se debe verificar desde el punto de vista algorítmico mediante simulaciones y datos de campo.

Debe poder funcionar en compatibilidad con los otros elementos de software y hardware del EDDR, debiéndose probar su desempeño en tiempo real. Esto implica funcionamiento de la versión funcional final en entorno linux, con memoria y procesador limitados, recepción y envío de datos por red, bajo standard Asterix Eurocontrol, y procesamiento de datos en modo *Track While Scan*.

Para ésto, es necesario en primera instancia realizar un análisis del sistema EDDR existente y del sistema de radar sobre el que se trabajará para conocer su desempeño y limitaciones.

Capítulo 2

Marco Teórico

2.1. Radar

2.1.1. Radar Primario

El término Radar, es generalmente utilizado para denominar a un radar del tipo primario. En un Radar Primario, un pulso de RF de alta potencia es transmitido desde una antena. Si este pulso alcanza un objetivo (ej, una aeronave), una pequeña parte de esta energía es reflejada nuevamente hacia la antena. La complejidad de un radar primario radica en la capacidad de detectar esta pequeña energía. Cuando una aeronave ha sido detectada, su rango puede ser determinado midiendo el tiempo transcurrido entre la transmisión y recepción del eco. El ángulo del objetivo es obtenido al conocerse el ángulo de apuntamiento de la antena.

La principal ventaja de un Radar Primario es que para su operación no requiere nada más que una superficie reflectante del objetivo. Las contras incluyen recepción de ecos no deseados provenientes de lluvia, tierra, etc, comúnmente llamados *clutter*. Existen muchas técnicas para mejorar el comportamiento en presencia de *clutter*, sin embargo, un radar primario, no puede determinar por si solo una aeronave de otra similar.

2.1.2. Radar Secundario

Los Radares Secundarios superan algunas de las desventajas de los radares primarios. Se denominan generalmente SSR-*Secondary Surveillance Radar*. Este tipo de radar, requiere que la aeronave a ubicar lleve a bordo un dispositivo receptor-transmisor denominado *Transponder*. Un SSR transmite pulsos desde una antena, de igual manera que un radar primario. Cuando el *Transponder* a bordo de la aeronave recibe el pulso de radar, emite una respuesta en una frecuencia diferente, esto permite diferenciar ecos que provienen de *clutter*.

Generalmente, la respuesta del *transponder* incluye la altura de la aeronave, además de su identificación. De esta manera, el radar debe calcular únicamente el rango y azimut del objetivo, simplificando de esta manera el diseño de la antena.

Los radares secundarios operan a menor potencia que los primarios, ya que solo se debe tener en cuenta la pérdida de energía del pulso entre el radar y el objetivo, y no su camino de regreso. También suelen disponer de antenas de menor tamaño, ya que necesitan resolución en elevación menor, como se puede observar en la Figura 2.1.

Los orígenes de este tipo de radar se remontan a la Segunda Guerra Mundial, cuando el objetivo era identificar aeronaves amigas y enemigas. Estos sistemas se denominaban IFF-*Identification Friend of Foe*, y consistían de pulsos en distinta frecuencia para la recepción y transmisión, sin ningún otro intercambio de información (Stevens, 1988).

Con el incremento del tráfico aéreo, y la incorporación de Radares Secundarios a la navegación civil, diversas evoluciones del sistema dieron lugar al denominado Modo S, antes



FIGURA 2.1: Radar AN/FPS-35 en Long Island, New York, EEUU. Se observa la gran antena correspondiente al Radar Primario, y una pequeña antena horizontal en la parte superior, correspondiente al Radar Secundario.

llamado DABS-*Discrete Address Beacon System*. Este modo permite a cada aeronave identificarse mediante un código de 24 bits, dando lugar a 16.777.215 posibilidades (Bouwman, 2009).

2.2. Sistemas de Seguimiento

Un sistema de seguimiento se utiliza para observar objetos en movimiento y proporcionar una secuencia ordenada de datos de posición. Este proceso implica estimar el estado de un objeto en movimiento a partir de mediciones remotas, indirectas e imprecisas. Las mediciones pueden originarse de uno o varios sensores, entre las que se encuentran, seguimiento por imágenes, seguimiento georeferenciado (GPS). En este trabajo, se profundizará sobre el seguimiento por radar.

El proceso de inferir el valor de una variable de interés a partir de una observación indirecta, imprecisa e incierta se llama **estimación**. Se denomina **decisión** a la selección de un elemento dentro de un conjunto de alternativas discretas. **Tracking** es la estimación del estado de un objeto en movimiento. Esto se logra usando uno o más sensores en posiciones fijas, o en plataformas móviles. Es un concepto más amplio que estimación, no sólo utiliza todas sus herramientas sino que también requiere un uso extenso de la estadística, especialmente en casos donde es necesaria la asociación de datos. **Filtrado** es la estimación del estado actual de un sistema dinámico a partir de datos ruidosos. **Suavizado** es la estimación del estado de un sistema en un tiempo previo al tiempo actual.

La variable de interés en un problema de estimación puede ser:

- Un parámetro (una cantidad invariante en el tiempo)
- El estado de un sistema dinámico (un vector)

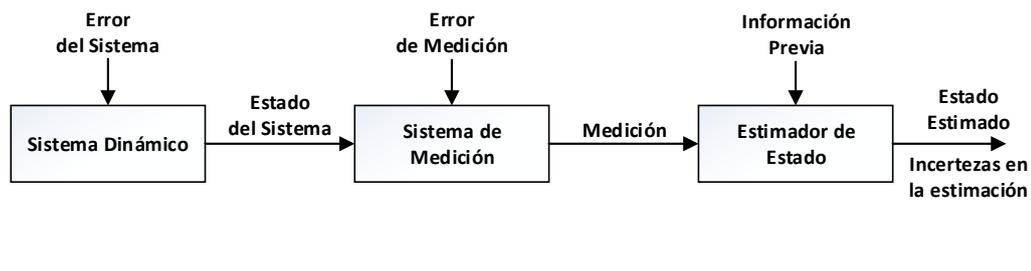


FIGURA 2.2: Proceso de Estimación.

La Figura 2.2 presenta un diagrama de bloques que ilustra la estimación de estados de un sistema dinámico. Los primeros dos bloques son cajas negras, no hay acceso a las variables en su interior. Las únicas variables disponibles para el estimador son las mediciones, que son afectadas por las fuentes de error en forma de ruido.

2.2.1. Sistemas de Vigilancia

El uso creciente de sistemas de vigilancia, tanto civiles como militares, ha generado un gran interés en algoritmos capaces de seguir gran cantidad de objetivos usando datos proporcionados por diversos tipos de sensores. La alta sensibilidad de los sensores modernos y la necesidad de trabajar en ambientes de baja SNR implica el manejo de gran cantidad de datos; la presencia de contra-medidas electrónicas puede incrementar la dificultad del *tracking*. El esfuerzo computacional y algorítmico para N objetivos puede ser sustancialmente mayor que N veces el esfuerzo para un único objetivo. Esto se debe al hecho de tener que establecer la correspondencia entre objetivos y observaciones.

En el seguimiento de objetivos puede haber incertidumbre adicional asociada a la medición que se suma a su inexactitud, generalmente modelada como ruido aditivo. Esta incertidumbre adicional está relacionada al origen de las mediciones. La medición usada en el algoritmo de *tracking* puede no haberse originado en el objetivo de interés. Esta situación puede ocurrir en un sistema de vigilancia cuando un radar opera en presencia de,

- *Clutter*
- Contra-medidas Electrónicas
- Falsas Alarmas
- Múltiples Objetivos

La incertidumbre en el origen de la medición también puede ocurrir cuando múltiples objetivos están cerca y, aún cuando se pueden separar las detecciones observadas, no se pueden asociar con seguridad a los objetivos. Una situación similar ocurre en el problema de formación de una pista, cuando hay múltiples objetivos pero su cantidad exacta es desconocida y algunas de las mediciones pueden ser espurias.

2.3. Tracking

Tracking o seguimiento, es el proceso aplicado a las mediciones obtenidas de un objetivo para mantener una estimación de su estado actual. El estado comprende en este caso componentes cinemáticos (posición, velocidad, aceleración). Las mediciones son observaciones

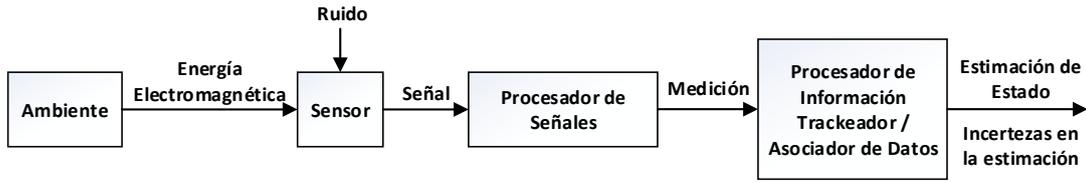


FIGURA 2.3: Componentes de un sistema de *Tracking*

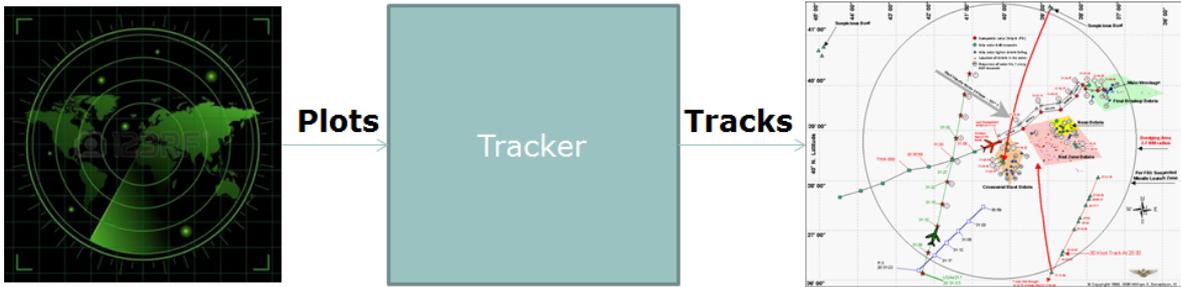


FIGURA 2.4: Proceso de *Tracking*. Se generan Pistas a partir de Detecciones.

contaminadas por ruido relacionadas al estado del objetivo en forma de, estimación directa de la posición (rango, azimut, elevación), o la medición del rango o azimut desde el sensor. Las mediciones de interés generalmente son el resultado de sistemas de procesamiento de señal y detección, como se ilustra en la Figura 2.3.

El proceso de *tracking* utiliza detecciones aisladas, denominados *plots*, para generar pistas (*tracks*) con mayor cantidad de información. Este proceso se ilustra en la Figura 2.4.

2.3.1. Enfoque de Seguimiento Bayesiano

Desde el enfoque Bayesiano, se asume que el estado es una variable aleatoria con función densidad de probabilidad (PDF) *a posteriori*, $p(x_k | z_{1:k})$, donde $z_{1:k} = \{z_i, i = 1, \dots, k\}$. Denominamos x_k al estado en el tiempo k , y z_k a las observaciones en ese instante de tiempo. La PDF $p(x_k | z_{1:k-1})$, es decir, el estado en el instante k , con la información de observaciones hasta el instante $k - 1$, es aproximada usando (1) una predicción y (2), una actualización.

1. En el paso de predicción, se calcula la PDF *a priori*, $p(x_k | z_{1:k-1})$ usando la ecuación de Chapman-Kolmogorov (Ross, 2014). Si $p(x_{k-1} | z_{1:k-1})$ está disponible, y el sistema se asume como una cadena de Márkov, la ecuación es,

$$p(x_k | z_{1:k-1}) = \int p(x_k | x_{k-1}) p(x_{k-1} | z_{1:k-1}) dx_{k-1} \quad (2.1)$$

2. En el tiempo k , una nueva medición z_k está disponible y la regla de Bayes puede ser usada para actualizar la PDF *a priori* a la probabilidad *a posteriori*, $p(x_k | z_{1:k})$, usando la probabilidad condicional $p(x_k | z_{1:k-1})$:

$$p(x_k | z_{1:k}) = \frac{p(z_k | x_k) p(x_k | z_{1:k-1})}{p(z_k | z_{1:k-1})} \quad (2.2)$$

CUADRO 2.1: Notación de Variables de Estado

Símbolo	Otras Notaciones	Definición
x		Vector de Estado
x_k		Componente k del vector x
x_k	$x[k]$	Elemento k de la secuencia de vectores $\dots, x_{k-1}, x_k, x_{k+1}, \dots$
\hat{x}	$E\langle x \rangle, \bar{x}$	Valor Estimado de x
$\hat{x}_{k k-1}$	$\hat{x}_k(-), \hat{x}_{k-}$	Estimación <i>a priori</i> de x_k , condicionado a todas las mediciones anteriores, excepto en el tiempo k .
$\hat{x}_{k k}$	$\hat{x}_k(+), \hat{x}_{k+}$	Estimación <i>a posteriori</i> de x_k , condicionado a todas las mediciones disponibles en el tiempo k .
\dot{x}	dx/dt	Derivada de x con respecto a t

Si este problema de calcular la PDF *a posteriori* recursivamente es resuelto de forma exacta, se obtiene la solución Bayesiana óptima. Sin embargo, esta solución óptima, solo existe en un conjunto muy reducido de casos, ya que implica la resolución de integrales complejas. Para el resto de los casos, se obtienen soluciones sub-óptimas.

En el Cuadro 2.1, se resume la notación utilizada en este trabajo, junto con otras notaciones comunes alternativas.

2.3.2. Filtro de Kalman Lineal

Con el Filtro de Kalman Lineal, se obtiene la solución Bayesiana óptima si la PDF *a posteriori* en todo paso de tiempo tiene distribución Gaussiana. Además, las funciones de avance del sistema y de medición tienen que ser conocidas y lineales. Su principal ventaja además de la solución óptima, es la baja carga computacional. Se hará una breve descripción del Filtro de Kalman Lineal. Para más información, referirse a (Bar-Shalom y Fortmann, 1988), o al paper original de Kalman, (Kalman, 1960).

Bajo las restricciones impuestas, se escribe la ecuación del sistema como,

$$x_k = F_k x_{k-1} + v_{k-1} \tag{2.3}$$

donde F_k es la función o matriz de avance del sistema, en el tiempo k . v_k , la función de perturbación del sistema, o ruido de sistema, se asume gaussiana, de media cero y covarianza Q_k . Por simplicidad de notación, se asumirá que la función de avance del sistema (F_k) es invariante en el tiempo (F), al igual que la matriz de covarianza ($Q_k = Q$), aunque el desarrollo es válido para el caso en que estas funciones sean variables.

La ecuación de medición se define como,

$$z_k = H_k x_k + w_k \tag{2.4}$$

donde H_k es la función o matriz de medición. w_k es la función de perturbación de la medición, o ruido de medición. Se asume gaussiana, de media cero y covarianza R_k . Las matrices de medición y de covarianza de la perturbación, también se asumirán invariantes en el tiempo; $H_k = H$ y $R_k = R$.

Las matrices F y H también son llamadas matrices de transición y de observación, respectivamente.

La predicción del estado, \hat{x}_k se calcula usando,

$$\hat{x}_{k|k-1} = F \hat{x}_{k-1|k-1} \tag{2.5}$$

donde el primer término representa la predicción del estado del sistema al tiempo k teniendo conocimiento del estado del mismo al tiempo $k - 1$.

La covarianza de la predicción es,

$$P_{k|k-1} = FP_{k-1|k-1}F^T + Q \quad (2.6)$$

donde el primer miembro representa la varianza de la predicción del estado del sistema al tiempo k con el conocimiento del estado del sistema al tiempo $k - 1$ y Q representa la contribución de las perturbaciones o ruido a la predicción del estado al tiempo k .

La predicción $\hat{z}_{k|k-1}$ de la medición z_k se calcula como,

$$\hat{z}_{k|k-1} = H\hat{x}_{k|k-1} \quad (2.7)$$

donde el primer miembro representa la predicción de la medición del estado del sistema al tiempo k teniendo el conocimiento del estado del mismo al tiempo $k - 1$.

La covarianza de la predicción de la medición es,

$$S_k = HP_{k|k-1}H^T + R \quad (2.8)$$

donde el primer miembro representa la covarianza de la predicción de la medición del estado del sistema al tiempo k y R representa la contribución del ruido o perturbación a dicha covarianza.

La ganancia de Kalman puede ser escrita como,

$$K_k = P_{k|k-1}H^T S_k^{-1} \quad (2.9)$$

siendo S_k^{-1} la matriz inversa de la covarianza de la estimación al tiempo k .

La innovación v_k , es la diferencia entre la predicción de la medición y la observación en ese instante de tiempo,

$$v_k = z_k - \hat{z}_k \quad (2.10)$$

El estado estimado puede ser calculado usando la ecuación de actualización de estado,

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k v_k \quad (2.11)$$

Finalmente, se calcula la Covarianza del Estado para el tiempo k como,

$$P_{k|k} = (1 - K_k H_k) P_{k|k-1} \quad (2.12)$$

Todos estos pasos se resumen en la Figura 2.5. Cada paso de tiempo el algoritmo avanza actualizando su estado. En la Eq. 2.11 se puede apreciar la influencia de la ganancia de Kalman en la actualización del estado. Una ganancia baja, le dará mas peso a la predicción, mientras que una ganancia alta hará que el sistema tome en cuenta la innovación. A su vez, la ganancia depende de las covarianzas Q y R . En el Cuadro 2.2 se resume la notación con los nombres de las variables utilizadas a lo largo del trabajo.

2.4. Asociación de Datos

Una pista es la estimación de una trayectoria a partir de un conjunto de mediciones (datos) que han sido asociadas con algún criterio.

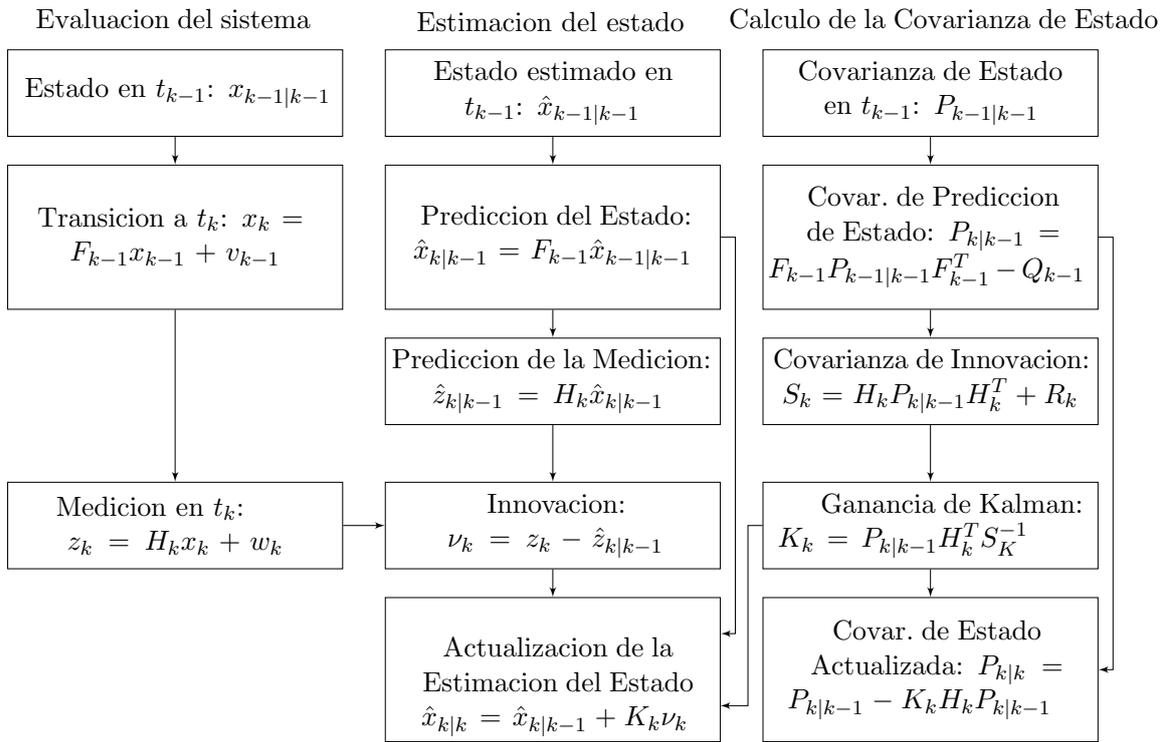


FIGURA 2.5: Un ciclo de la estimación de estado de un Sistema Lineal usando Filtro de Kalman. Se muestran las ecuaciones extendidas para los casos de F, H, Q y R variantes en el tiempo.

CUADRO 2.2: Notación de Matrices y Vectores

Símbolo	Nombre	Dimensión
x	Estado del Sistema	n
v	Ruido de Proceso	n
z	Medición	l
w	Ruido de Medición	l
F	Matriz de Transición de Estado Matriz de Avance de Sistema	$n \times n$
Q	Covarianza de Ruido de Proceso	$n \times n$
H	Matriz de Medición	$l \times n$
R	Covarianza de Ruido de Medición	$l \times l$
P	Covarianza del Estado	$n \times n$
S	Covarianza de la Predicción de la Medición	$l \times l$
K	Ganancia de Kalman	$n \times l$
ν	Innovación	l



FIGURA 2.6: Radar Vitro RIR-778C. Sistema de seguimiento para un único objetivo con capacidad de maniobra.

El proceso de Asociación de Datos conecta mediciones a pistas conocidas. El enfoque con el que se trata el problema varía si se trata de seguir a un único objetivo, a múltiples objetivos, sin capacidad, o con capacidad de maniobra, con aumento de complejidad a medida que aumenta el número de objetivos o la capacidad de variación de su trayectoria.

En la Figura 2.6 se muestra un radar de trayectografía denominado Vitro RIR-778C. Dicho radar, operado por la Fuerza Aérea Argentina, se lo utilizó para el desarrollo de cohetes de combustible sólido a fines de la década de 1980, particularmente en el proyecto Cóndor. Es un sistema de seguimiento de alta precisión, que enfoca el haz mediante la antena parabólica para apuntar a un único objetivo. Para este tipo de radares, la asociación que hay que hacer es seleccionar la mejor detección para la trayectoria en curso, pudiéndose obtener detecciones por ruido, o por interferencia de algún otro blanco.

La complejidad para el proceso de asociación de datos aumenta cuando es necesario el seguimiento de múltiples objetivos. En la Figura 2.7 se observa el Radar Primario Argentino 3D (RPA), desarrollado por Invap. Dicho radar puede realizar el seguimiento de múltiples objetivos en tiempo real, en un proceso denominado *Track While Scan*.

Un problema multi-objetivo consiste en llevar a cabo estas asociaciones (proceso de correlación) a mediciones con incertidumbre sobre su origen debido a,

- Falsas Alarmas aleatorias en el proceso de detección
- *Clutter* debido a otros reflectores o radiadores cerca del objetivo de interés
- Objetivos interfiriendo las detecciones
- Contra-medidas Electrónicas

La probabilidad de obtener una medición a partir del objetivo de interés es menor a la unidad. Los problemas de asociación de datos se pueden categorizar según

- Asociación medición a medición (formación de pistas)
- Asociación de medición a pista (mantenimiento o actualización de pistas)

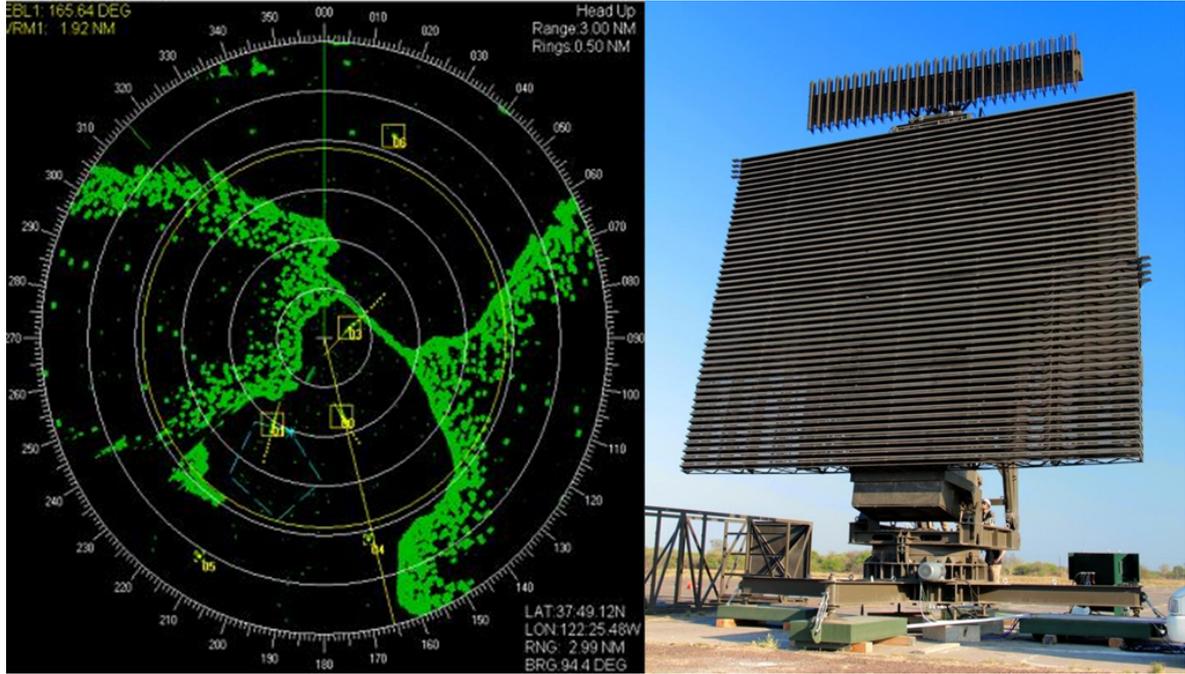


FIGURA 2.7: Radar Primario Argentino 3D de la firma Invap. Capaz de hacer seguimiento de múltiples objetivos.

- Asociación de pista a pista (fusión de pistas en sistemas multi-sensor)

Hay dos aproximaciones posibles para llevar a cabo el proceso de decisión: asociación de datos no bayesiana, utilizando criterios heurísticos, como distancia o herramientas estadísticas (máxima verosimilitud, por ejemplo); o asociación de datos probabilística (bayesiana) usando procesos de estimación (Bar-Shalom, Willett y Tian, 2011).

2.4.1. Ventanas de Detección

En este trabajo se hará foco en el método de asociación utilizando ventana o región de detección. Una región de detección es una zona en donde la próxima detección tiene gran probabilidad de aparecer. Para poder definir esta región, es necesario que el objetivo se encuentre en una pista, de manera que pueda existir una predicción de la medición ($\hat{z}_{k|k-1}$) y una matriz de covarianza (S_k) de esta predicción. Teniendo en cuenta la definición de innovación en la Eq. 2.10, se define la ventana de detección como,

$$\tilde{V}_k(\gamma) = \{z : v_k^T S_k^{-1} v_k \leq \gamma\} \quad (2.13)$$

Donde el parámetro γ se utiliza para obtener la probabilidad deseada. Las mediciones que caen dentro de la región de asociación se llaman mediciones validadas.

En la Figura 2.8 se ilustra una ventana de detección junto a 4 detecciones validadas. La forma del elipse está dada por S_k y su posición centrada en $\hat{z}_{k|k-1}$.

La cantidad de objetos a seguir puede ser mayor a uno, por lo que se tiene que decidir qué medición corresponde a qué objeto. La complejidad aumenta en situaciones donde la cantidad de objetivos es desconocida y variable. El algoritmo de asociación de datos debe decidir si una medición es correcta o incorrecta y si se ha de crear una nueva pista o actualizar una ya existente.

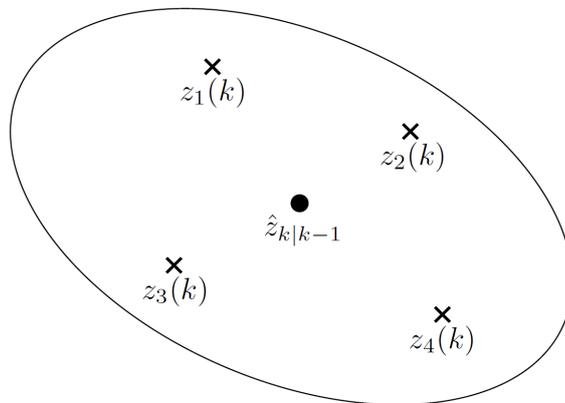


FIGURA 2.8: Problema de Asociación de Datos. La medición predicha de un único objetivo junto a cuatro mediciones validadas.

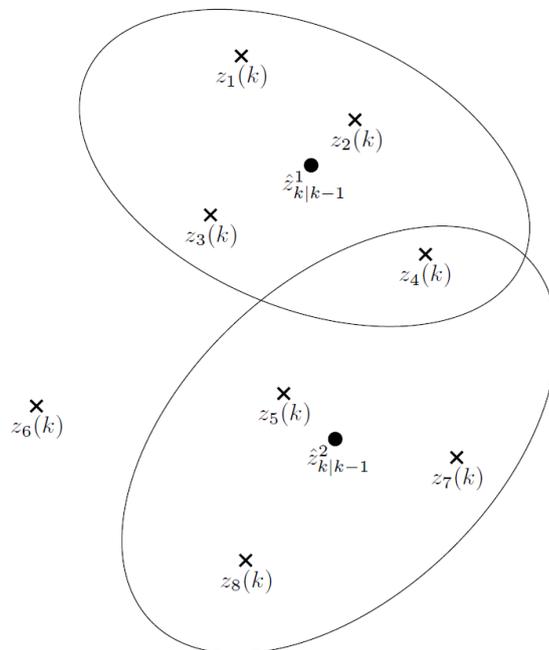


FIGURA 2.9: Problema de Asociación de Datos. Dos mediciones predichas junto a varias mediciones.

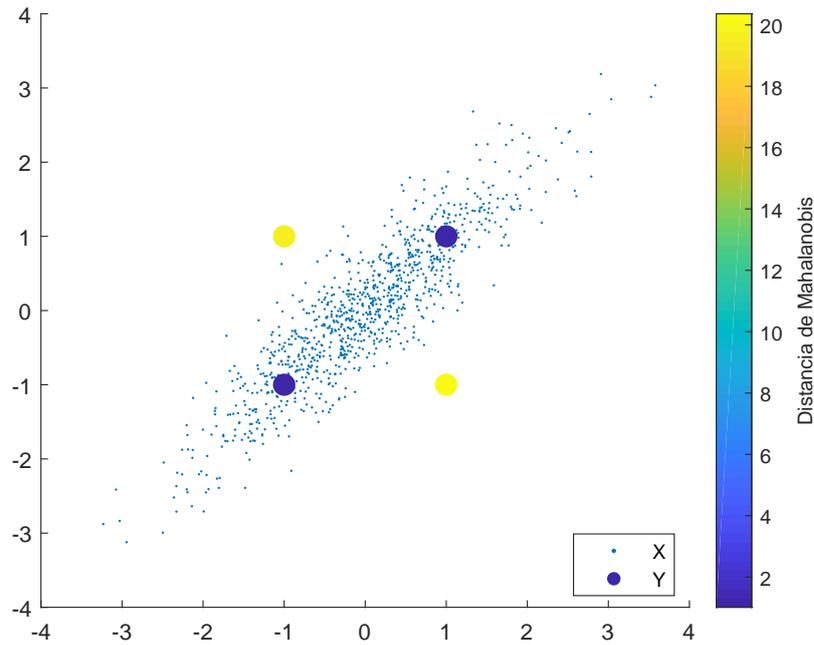


FIGURA 2.10: Comparación gráfica entre las distancias Euclidiana y de Mahalanobis.

Una situación más compleja se observa en la Figura 2.9, donde $\hat{z}_{k|k-1}^1$ y $\hat{z}_{k|k-1}^2$ son mediciones predichas para dos pistas diferentes. $z_1(k), z_2(k), z_3(k)$ y $z_4(k)$ son mediciones validadas para la primera pista y $z_4(k), z_5(k), z_7(k)$ y $z_8(k)$ correspondientes a la segunda. Todas las mediciones que no sean usadas para la actualización de una pista existente pueden deberse a *clutter* o a nuevos objetivos. También puede darse la situación de que ninguna de las mediciones corresponda a la pista y que el objeto haya desaparecido (Bar-Shalom y Fortmann, 1988).

2.4.2. Nearest Neighbour Standard Filter

El algoritmo de vecino más próximo (*Nearest Neighbour Standard Filter*-NNSF) es el más directo para resolver el problema de asociación de datos. El NNSF selecciona la medición validada más cercana a la medición predicha $\hat{z}_{k|k-1}$, de forma de minimizar la distancia entre el punto predicho y medido. Para esto, se utiliza el concepto de distancia de Mahalanobis o distancia estadística, utilizada en la Ecuación 2.13, que se define como,

$$d^2 = v_k^T S_k^{-1} v_k \quad (2.14)$$

La distancia de Mahalanobis tiene en cuenta la correlación entre las variables. En la Figura 2.10 se compara gráficamente la distancia euclidiana con la distancia estadística (Paluszek y Thomas, 2017). Las observaciones Y se encuentran a la misma distancia euclidiana del conjunto de datos X. Sin embargo, se observa que dos de ellas se encuentran a una menor distancia estadística de la media.

Utilizando este cálculo de distancia, en la Figura 2.9 la selección para actualizar las pistas sería z_2 y z_5 para la primer y segunda pista respectivamente.

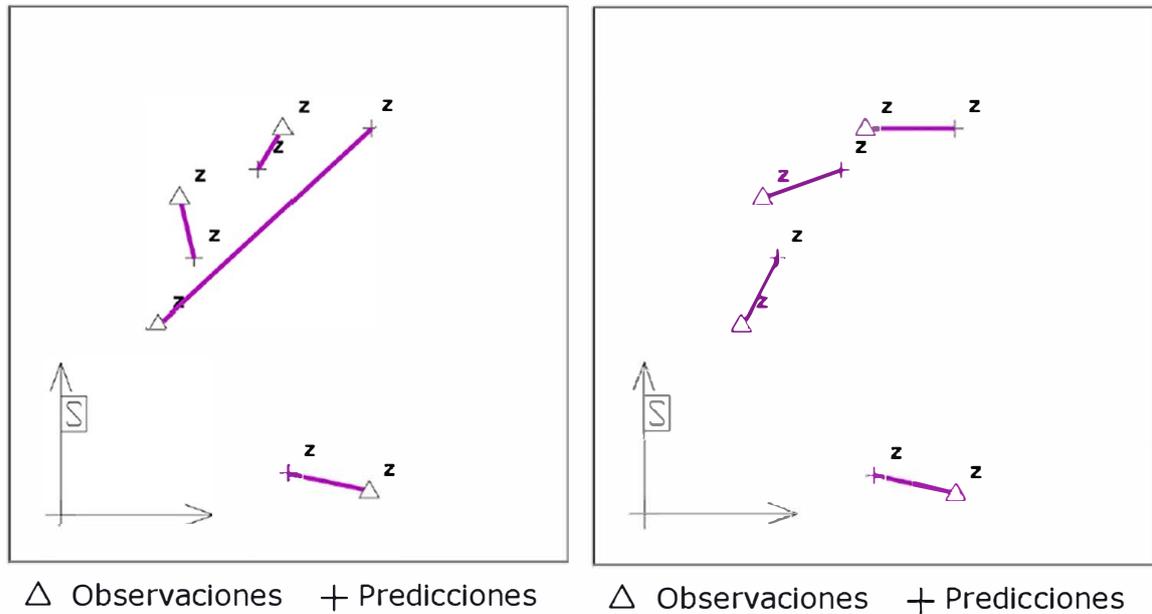


FIGURA 2.11: Asignación de observaciones a pistas usando NNSF (Izquierda) y usando GNN (Derecha).

La principal ventaja de este filtro es su baja carga computacional, sin embargo, puede llevar al caso en que una detección sea asociada a más de una pista, o que el resultado de las asociaciones dependa del orden en el que se evalúan los datos. Para evitar estos casos se utiliza el filtro de vecino más próximo global (*Global Nearest Neighbour-GNN*).

2.4.3. Global Nearest Neighbour

El algoritmo de *Global Nearest Neighbour* evita situaciones como la de la Figura 2.11, en la que, según el orden en el que se asocian las detecciones a las pistas existentes, puede darse una situación como la de la izquierda. El algoritmo GNN reduce el costo global de las asignaciones mediante el algoritmo Húngaro (también llamado Kuhn–Munkres) (Kuhn, 1955), (Munkres, 1957).

Algoritmo Húngaro

A continuación se ilustra brevemente el problema resuelto por Munkres.

Sea C una matriz de dimensión $n \times n$ representando el costo de n trabajadores $\{a, b, c\}$ para realizar n trabajos $\{p, q, r\}$. El problema es asignar trabajos a trabajadores para minimizar el costo total. Como cada trabajador puede realizar un solo trabajo, y cada trabajo puede ser asignado a un solo trabajador, las asignaciones constituyen un set independiente de la matriz C . El costo computacional de resolver el problema por fuerza bruta es $O(n!)$. El costo del algoritmo Húngaro es $O(n^3)$. A continuación se lista el algoritmo.

- **Paso 0:** Crear una matriz de costos, en la que cada elemento sea el costo de asignar el trabajo j al trabajador i .

$$C_{i,j} = \begin{matrix} & p & q & r \\ a & \left[\begin{array}{ccc} 15 & 40 & 45 \\ 20 & 60 & 35 \\ 20 & 40 & 25 \end{array} \right] \end{matrix} \quad (2.15)$$

- **Paso 1:** Se encuentra el menor elemento de cada fila, y se lo resta a todos los elementos de esa fila.

$$\begin{bmatrix} 0 & 25 & 30 \\ 0 & 40 & 15 \\ 0 & 20 & 5 \end{bmatrix}$$

- **Paso 2:** Se repite el procedimiento del Paso 1 con las columnas (el mínimo elemento de cada columna se resta a todos los elementos de la misma).

$$\begin{bmatrix} 0 & 5 & 25 \\ 0 & 20 & 10 \\ 0 & 0 & 0 \end{bmatrix}$$

- **Paso 3:** Tapar los ceros de la matriz usando la menor cantidad de líneas horizontales y verticales.

$$\begin{bmatrix} \left| \begin{array}{ccc} 0 & 5 & 25 \\ 0 & 20 & 10 \\ \hline 0 & 0 & 0 \end{array} \right. \end{bmatrix}$$

- **Paso 4:** Si la mínima cantidad de líneas es igual a n , los ceros son las asignaciones y el problema está resuelto. En caso contrario, pasar al Paso 5.
- **Paso 5:** Encontrar el menor elemento sin tapar. Restar ese valor a todas las filas no tapadas.

$$\begin{bmatrix} -5 & 0 & 20 \\ -5 & 15 & 5 \\ 0 & 0 & 0 \end{bmatrix}$$

Después se suma ese valor a todas las columnas tapadas.

$$\begin{bmatrix} 0 & 0 & 20 \\ 0 & 15 & 5 \\ 5 & 0 & 0 \end{bmatrix}$$

Volver al Paso 3 y continuar el procedimiento. Se tapan nuevamente los ceros.

$$\begin{bmatrix} \left| \begin{array}{ccc} \hline 0 & 0 & 20 \\ 0 & 15 & 5 \\ \hline 5 & 0 & 0 \end{array} \right. \end{bmatrix}$$

Como se necesitan 3 líneas, se ha encontrado la solución.

$$C_{i,j} = \begin{array}{c} a \\ b \\ c \end{array} \begin{array}{c} p \\ q \\ r \end{array} \begin{bmatrix} 15 & 40 & 45 \\ 20 & 60 & 35 \\ 20 & 40 & 25 \end{bmatrix}$$

Capítulo 3

Desarrollo

En este capítulo se describe el procedimiento llevado a cabo para implementar el sistema de *Track While Scan*–TWS. Como se explicó en el Capítulo 1, este trabajo es continuación de (Montamat, 2015), por lo que sus datos de entrada serán el producto del Software Combinador. Se comenzará haciendo una breve descripción de ese proceso.

3.1. Extractor Digital de Datos de Radar–EDDR

El EDDR es un equipo desarrollado por Fuerza Aérea Argentina para digitalizar la salida de radares con procesamiento analógico de datos. En las Figuras 3.1 y 3.2, se puede observar el hardware correspondiente al sistema EDDR.

3.1.1. Combinador

En la Figura 3.3 se puede ver un diagrama de bloques que ubica al EDDR en el sistema radar. El radar Bendix FPS-113 genera los datos de video primarios y secundarios en forma analógica. Estos datos ingresan al EDDR a través de sus procesadores primarios y secundarios respectivamente. Ahí son digitalizados y transmitidos al combinador. El combinador, mediante una ventana de asociación o combinación, combinará los datos primarios y secundarios que considere que corresponden a una misma aeronave. La salida del combinador es una estructura de datos denominada *plot*. Existirán de esta manera,

- *Plots* primarios, que no tienen una detección secundaria asociada.
- *Plots* secundarios, que no tienen una detección primaria asociada.
- *Plots* combinados, que combinan una detección primaria y una secundaria correspondientes al mismo objetivo.

El sistema divide a la zona de análisis en sectores, como se observa en la Figura 3.4. Los datos se combinan por sector, transmitiéndose mensajes de cambio de sector y de TOP Norte, cuando la antena recorre los 0° de azimut.

La información asociada a las tres estructuras de datos de salida del combinador, compatible con formato Asterix Eurocontrol, se puede ver detalladamente en (Safety of Air Navigation, 2012). La información más relevante para este trabajo es,

- SAC: *System Area Code*, se indica el código de país, 0xE2 para Argentina.
- SIC: *System Identification Code*, se indica el código de radar. 134 (0x86) corresponde a Posadas, 135 (0x87) Resistencia.
- Hora, proporcionada por GPS.



FIGURA 3.1: El subrack del sistema EDDR contiene los procesadores primario, secundario, simulador y combinador-trackeador.

- Coordenadas polares, ρ (rango), θ (azimut) de la detección.
- En caso de *plots* secundarios o combinados, código de Modo 3/A. Indica el código de identificación de la aeronave.

La salida de datos en formato Asterix Eurocontrol Cat. 48 es compatible con las consolas desarrolladas por INVAP para representación multipropósito de datos de radar.

3.2. Trackeador y Generador de Pistas

El ciclo de *tracking* a desarrollar es el ciclo clásico que se muestra en la Figura 3.5, (Richards, 2014). El seguimiento se realizará en dos dimensiones, con rango y azimut como datos de entrada del radar, y seguimiento en coordenadas cartesianas.

3.2.1. Asociación de Detecciones a Pistas

Al ser un proceso iterativo, la Asociación de Detecciones a Pistas utiliza como punto de comparación la predicción de la posición de cada pista a partir del filtrado en el tiempo $k - 1$. El proceso es el siguiente,

1. Predicción de la nueva posición de las pistas (Eq. 2.5).
2. Cálculo de distancia de Mahalanobis entre cada posición predicha y la nueva detección (Eq. 2.14).



DintelGID.com.ar

Esteban G. Brea

FIGURA 3.2: El sistema EDDR montado en su rack correspondiente.

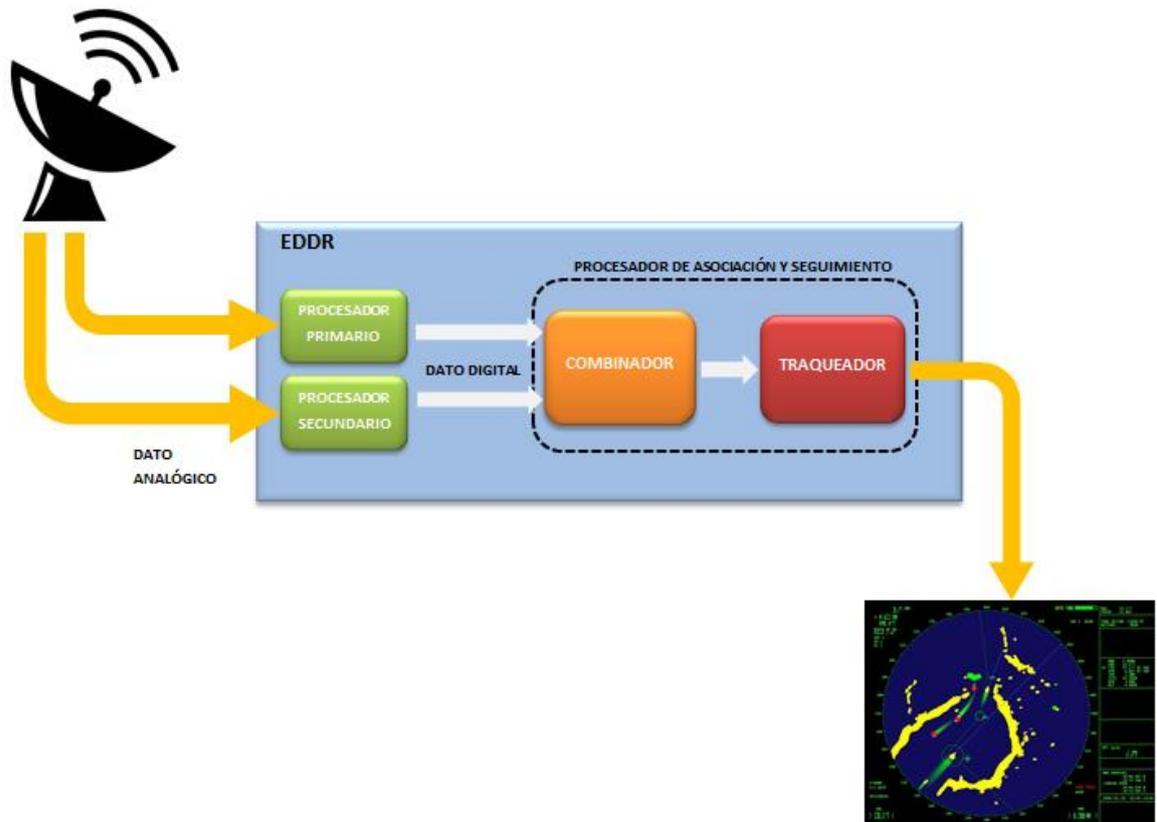


FIGURA 3.3: El sistema EDDR recibe datos analógicos del radar y produce salida digital para ser transmitida a los centros de control.

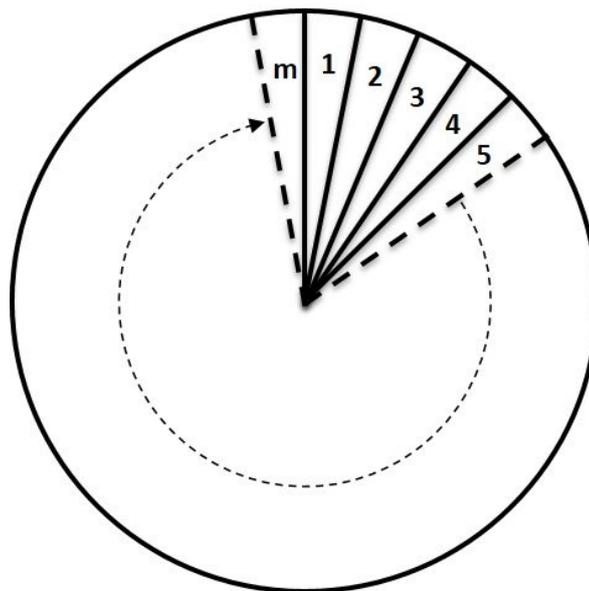


FIGURA 3.4: El sistema EDDR divide en sectores al área de búsqueda del radar.

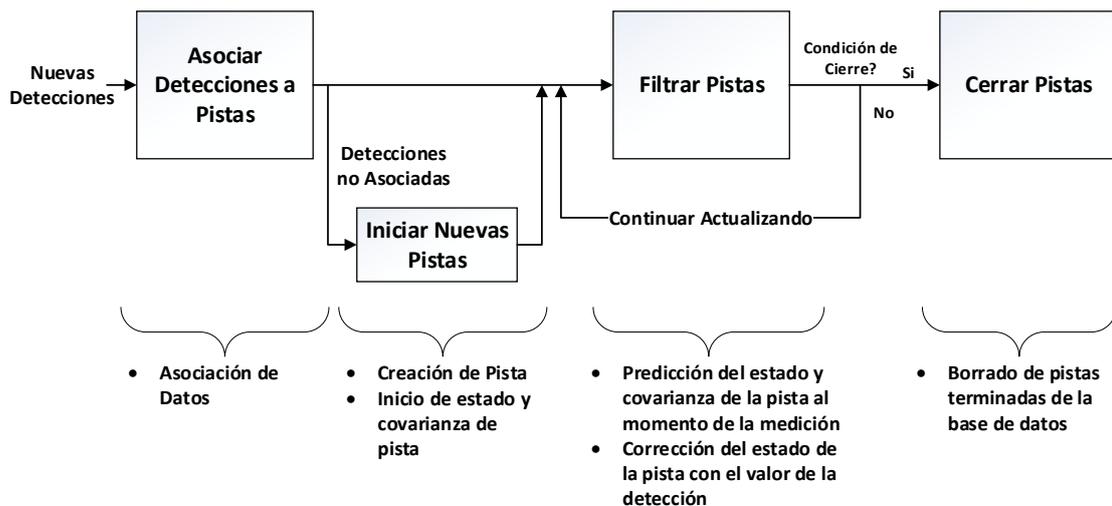


FIGURA 3.5: El ciclo de *tracking* clásico.

3. Generación de matriz de costos entre pistas y detecciones (2.15).
4. Ejecución del algoritmo de Munkres para la asociación (Sección 2.4.3).

Se establece en el archivo de configuración un parámetro para especificar el Costo de No Asociación (*costOfNonAssignment*), por encima del cual la detección no se asociará a ninguna pista, dando lugar a una nueva pista tentativa. Este parámetro definirá el tamaño de la ventana de detección definida en la Sección 2.4.1.

3.2.2. Iniciación de Pistas

El proceso de iniciación de pista se ejecuta cada vez que se decide que una detección no será asociada a ninguna pista existente. En cada ciclo de *tracking* (Figura 3.5), se toma la lista de detecciones sin asociar y se las asigna como pistas tentativas. Las pistas tentativas serán tratadas como pistas comunes, salvo que no serán enviadas por el programa, hasta que alcancen la categoría de pista definida. La pista será definida cuando alcance una vida de 3 detecciones consecutivas, valore seleccionado en forma experimental. Este parámetro (*ageToRealTrack*) es configurable mediante el archivo de configuración del programa.

En este proceso se inicializan las matrices de Transición de Estado (F), Medición (H), Covarianza de Ruido de Proceso (Q), y Covarianza de Ruido de Medición (R). También se establecen las condiciones iniciales para el Valor Inicial Estimado (posición igual a la detección, velocidad y aceleración iguales a cero) y Error de Covarianza Inicial Estimada (*initialStateCovariance*).

No se tiene medición directa de la velocidad ni aceleración del objetivo, sino que éstos parámetros se calculan a partir de las detecciones. Es por esto, que una única detección no tiene información de posición o aceleración, y se la inicializa en cero, que será el valor medio de sus posibles valores. Posteriormente, se establecerán valores elevados para sus varianzas iniciales.

3.2.3. Filtrado de Pistas

Se puede descomponer el movimiento del objetivo a seguir en dos movimientos independientes en coordenadas cartesianas, esto lleva a un conjunto de ecuaciones de la forma,

$$\begin{cases} x_{k+1} = x_k + \dot{x}_k T + \frac{1}{2} \ddot{x}_k T^2 \\ \dot{x}_{k+1} = \dot{x}_k + \ddot{x}_k T \\ \ddot{x}_{k+1} = \ddot{x}_k + n_{ax} \\ y_{k+1} = y_k + \dot{y}_k T + \frac{1}{2} \ddot{y}_k T^2 \\ \dot{y}_{k+1} = \dot{y}_k + \ddot{y}_k T \\ \ddot{y}_{k+1} = \ddot{y}_k + n_{ay} \end{cases} \quad (3.1)$$

La Eq. 3.1 supone un modelo cartesiano de aceleración perturbada según los parámetros n_{ax} y n_{ay} . Dicho modelo se puede expresar en forma matricial de la forma,

$$\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \\ y \\ \dot{y} \\ \ddot{y} \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & T & \frac{T^2}{2} & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & \frac{T^2}{2} \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \\ y \\ \dot{y} \\ \ddot{y} \end{bmatrix}_k + \begin{bmatrix} 0 \\ 0 \\ n_{ax} \\ 0 \\ 0 \\ n_{ay} \end{bmatrix}_k \quad (3.2)$$

Las Eqs. 3.1 y 3.2 describen el modelo del sistema o ecuación de sistema, definido en la Eq. 2.3. La matriz de avance del sistema, F se la define como invariante en el tiempo,

$$F = \begin{bmatrix} 1 & T & \frac{T^2}{2} & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & \frac{T^2}{2} \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

La matriz de perturbación del sistema, v_k , será un parámetro que afectará únicamente a la aceleración. La perturbación v_k , se asume gaussiana con media cero y varianza σ^2 . La matriz de covarianzas Q_k es considerada invariante en el tiempo, por lo que se la denomina matriz Q . Esta matriz es definida como,

$$\begin{aligned} Q &= E \left\{ v_k, v_k^T \right\} = E \left\{ \begin{bmatrix} 0 \\ 0 \\ n_{ax} \\ 0 \\ 0 \\ n_{ay} \end{bmatrix} \begin{bmatrix} 0 & 0 & n_{ax} & 0 & 0 & n_{ay} \end{bmatrix} \right\} \\ &= E \left\{ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & n_{ax}^2 & 0 & 0 & n_{ax}n_{ay} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & n_{ay}n_{ax} & 0 & 0 & n_{ay}^2 \end{bmatrix} \right\} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{ax}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{ay}^2 \end{bmatrix} \quad (3.4) \end{aligned}$$

Se asume por simplicidad, que las perturbaciones en X y en Y son independientes.

La posición del objetivo es observada por el radar, ubicado en el origen del sistema de

coordenadas. Posteriormente se pueden hacer las transformaciones de coordenadas necesarias para ubicar la trayectoria en ejes tierra o la referencia que sea necesaria.

El radar envía al *tracker* la distancia (ρ) a la que se ubica el objetivo, y el ángulo de la antena (θ) en ese instante.

$$\begin{cases} \hat{\rho}_k = \rho_k + n_\rho \\ \hat{\theta}_k = \theta_k + n_\theta \end{cases} \quad (3.5)$$

Donde n_ρ y n_θ son los errores en la medición de rango y de ángulo. El error en la medición de rango se debe al ancho de pulso del transmisor. El blanco se puede ubicar en cualquier lugar dentro del ancho de éste. El error en la medición angular se debe al ancho del haz del lóbulo de la antena. El radar no puede identificar en qué posición dentro del haz se encuentra el eco. Solamente puede leer el valor del *encoder* o sincro del pedestal al momento de recibir el eco. La Eq. 3.5 expresada en forma matricial es,

$$\begin{bmatrix} \hat{\rho}_k \\ \hat{\theta}_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \rho_k \\ \dot{\rho}_k \\ \ddot{\rho}_k \\ \theta_k \\ \dot{\theta}_k \\ \ddot{\theta}_k \end{bmatrix} + \begin{bmatrix} n_\rho \\ n_\theta \end{bmatrix} \quad (3.6)$$

La Eq. 3.6 es la ecuación de medición definida en la Eq. 2.4. Definimos entonces la matriz de medición H , también invariante en el tiempo como,

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (3.7)$$

La matriz de covarianzas de la perturbación en la medición w_k utilizada en la Eq. 2.4, se define entonces como,

$$R = \begin{bmatrix} \sigma_\rho^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix} \quad (3.8)$$

Suponiendo nuevamente independencia entre las perturbaciones de rango y ángulo.

Se han definido hasta aquí, las ecuaciones de Modelo de Sistema (Eq. 3.2 \leftrightarrow Eq. 2.3) y de Medición (Eq. 3.6 \leftrightarrow Eq. 2.4). El Modelo de Sistema está representado en coordenadas cartesianas, mientras que el Modelo de Medición en coordenadas polares. Las mediciones z_k pueden ser convertidas de polares a cartesianas sin necesidad de modificar la matriz H . Sin embargo, la conversión no será directa para w_k . Se expresa entonces la ecuación de medición en coordenadas cartesianas,

$$\begin{bmatrix} x \\ y \end{bmatrix}_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \\ y \\ \dot{y} \\ \ddot{y} \end{bmatrix}_k + [w_k^{polar} \rightarrow w_k^{cart}] \quad (3.9)$$

De la Figura 3.6 se pueden derivar las siguientes ecuaciones de transformación,

$$\begin{cases} \Delta x = \Delta \rho \cos \theta - \rho \Delta \theta \sin \theta \\ \Delta y = \Delta \rho \sin \theta + \rho \Delta \theta \cos \theta \end{cases} \quad (3.10)$$

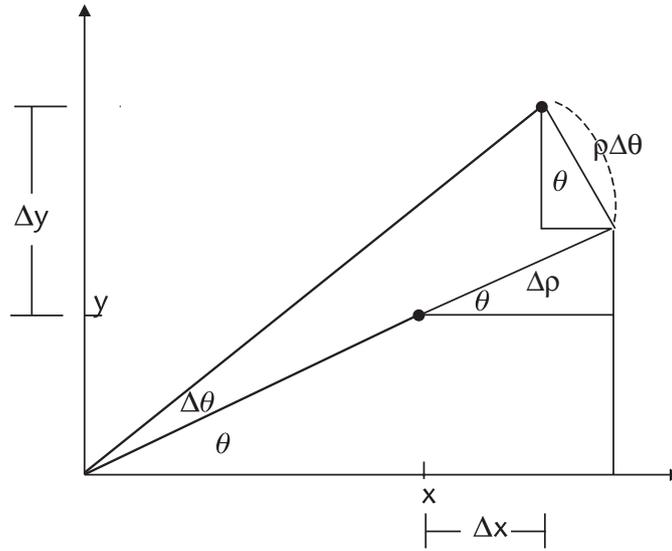


FIGURA 3.6: Transformación de coordenadas polares a cartesianas para las detecciones de blancos de radar.

La Eq. 3.10 expresada en forma matricial,

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \cos \theta & -\rho \sin \theta \\ \sin \theta & \rho \cos \theta \end{bmatrix} \begin{bmatrix} \Delta \rho \\ \Delta \theta \end{bmatrix} \quad (3.11)$$

El Jacobiano utilizado para la transformación queda definido entonces como,

$$J_k(\rho, \theta) = \begin{bmatrix} \cos \theta_k & -\rho_k \sin \theta_k \\ \sin \theta_k & \rho_k \cos \theta_k \end{bmatrix} \quad (3.12)$$

Y la transformación $[w_k^{polar} \rightarrow w_k^{cart}]$, o $R(\rho, \theta) \rightarrow R(x, y)$ se obtiene por la multiplicación de las matrices,

$$R_k(x, y) = J_k(\rho, \theta) R(\rho, \theta) J_k^T(\rho, \theta) \quad (3.13)$$

Como se puede observar, la matriz de covarianzas de la incertidumbre en la medición (R) es invariante en el tiempo en coordenadas polares, pero dependiente del tiempo en coordenadas cartesianas, ya que la conversión es dependiente del ángulo y rango del objetivo.

La expresión para la conversión queda,

$$\begin{aligned} R_k(x, y) &= \begin{bmatrix} \cos \theta_k & -\rho_k \sin \theta_k \\ \sin \theta_k & \rho_k \cos \theta_k \end{bmatrix} \begin{bmatrix} \sigma_\rho^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix} \begin{bmatrix} \cos \theta_k & \sin \theta_k \\ -\rho_k \sin \theta_k & \rho_k \cos \theta_k \end{bmatrix} \\ R_k(x, y) &= \begin{bmatrix} \sigma_\rho^2 \cos^2 \theta_k + \sigma_\theta^2 \rho_k^2 \sin^2 \theta_k & \frac{1}{2} \sin(2\theta_k)(\sigma_\rho^2 - \sigma_\theta^2 \rho_k^2) \\ \frac{1}{2} \sin(2\theta_k)(\sigma_\rho^2 - \sigma_\theta^2 \rho_k^2) & \sigma_\rho^2 \sin^2 \theta_k + \sigma_\theta^2 \rho_k^2 \cos^2 \theta_k \end{bmatrix} \end{aligned} \quad (3.14)$$

La expresión 3.14 depende únicamente del valor de la medición de rango y ángulo, y de la matriz R^{polar} (Eq. 3.8).

Para calcular los parámetros σ_ρ^2 y σ_θ^2 , se usan las especificaciones del radar referidas al ancho de pulso y ancho del haz de la antena. Como se indicó en el Capítulo 1, el ancho de pulso que usa el transmisor del radar Bendix es de 4 μ s. En (Richards, 2010) se usa como aproximado para la varianza en rango,

$$\sigma_p^2 \geq \left(\frac{c\tau}{2SNR} \right)^2 \quad (3.15)$$

Donde c es la velocidad de la luz, τ es el ancho del pulso y SNR es la relación señal ruido del sistema. Se utilizará una $\sigma_p = 200$ mts.

Para el cálculo de σ_θ^2 , también se referencia (Richards, 2010),

$$\sigma_\theta^2 = \frac{\theta_3^2}{N \cdot SNR \cdot \alpha^2} \quad (3.16)$$

N es la cantidad de pulsos integrados en el intervalo de lectura, SNR es la relación señal/ruido, θ_3 es el ancho del haz de la antena. α es una constante que depende del patrón de radiación de la antena y la velocidad de giro. Para radares de búsqueda, $\alpha \approx 0,34$. Como se indicó en el Capítulo 1, el ancho del haz de la antena es de 1° . Se tomará como valor de $\sigma_\theta = 0,6^\circ$.

Con esto se completa el proceso de filtrado, en el que el algoritmo ejecuta un ciclo de la Figura 2.5 cada vez que recibe un nuevo dato.

3.2.4. Cierre de Pistas

Las pistas se cierran cuando el objetivo que se está siguiendo no se encuentra dentro del rango de detección del radar, o la pista no cumple con las condiciones para ser considerada válida. Las condiciones que generan un cierre de pista son,

- La pista ha dejado de recibir asociaciones por más de 3 vueltas consecutivas del radar (`invisibleForTooLong`).
- La pista es "joven", es decir, tiempo de vida menor a 8 vueltas (`ageThreshold`) y ha tenido una visibilidad menor al 60% (`visibility`).

Todos estos parámetros indicados entre paréntesis, son configurables a través del archivo de configuración del programa.

3.2.5. Implementación del Algoritmo

El programa se implementa como un algoritmo iterativo respondiendo al diagrama de alto nivel de la Figura 3.5.

Caso Tracking vuelta a vuelta

Se describe a continuación el proceso detallado en la Figura 3.7. Este proceso comienza cada vez que se recibe un pulso de *TOP NORTE* de la antena, es decir, cada vez que la antena pasa por el norte.

1. Cada vez que comienza el ciclo, se predice la nueva posición de todas las pistas, a través de sus filtros de kalman. Se actualiza la predicción del estado y de la covarianza de cada pista.
2. A continuación, se calcula la distancia entre cada una de las detecciones realizadas en la última vuelta con todas las posiciones predichas en el paso anterior. Con esta distancia se asignan detecciones a pistas, para finalizar la etapa de asociación.

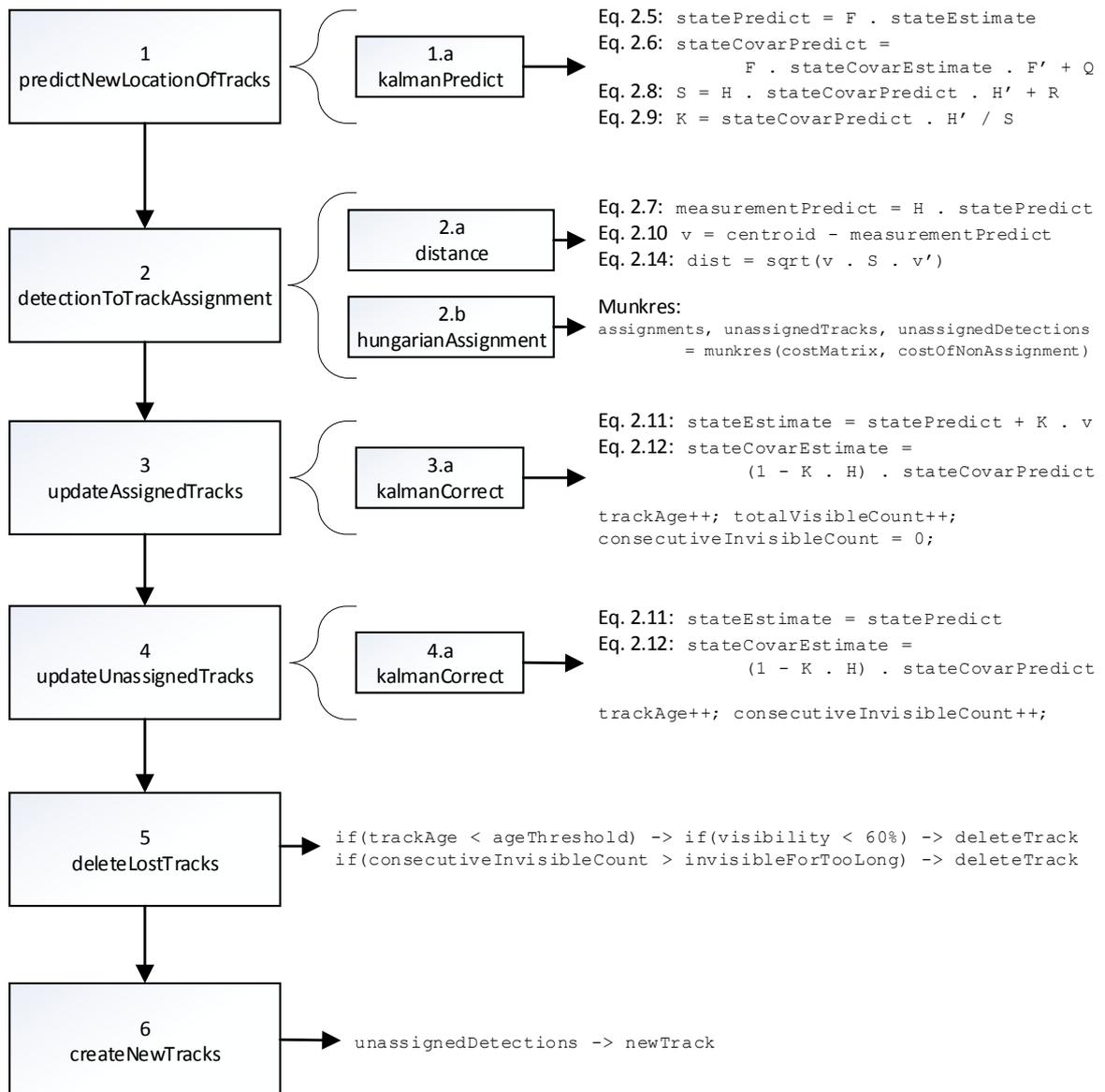


FIGURA 3.7: El algoritmo de *tracking* desglosado en sus subprocesos.

3. Posteriormente se actualizan aquellas pistas que han sido asignadas, mediante la ecuación de actualización de estado. También se actualiza la puntuación de la pista, incrementando su tiempo de vida y su tiempo visible. En esta etapa se decide si la pista pasará de ser tentativa a pista real, en caso de superar cierta cantidad de detecciones consecutivas. Si este es el caso, estará disponible para ser enviada por el generador de tramas Asterix.
4. Se actualizan aquellas pistas que no han sido asignadas con ninguna detección, considerando que la detección ha sido igual a la predicción. Se inicia el contador de pista invisible.
5. Se eliminan aquellas pistas que cumplen con la condición de pista no válida.
6. Se utilizan todas las detecciones no asignadas en el punto 2 para crear nuevas pistas tentativas.

Caso Track While Scan

Para el caso de TWS, se debe proceder a realizar el cálculo tan pronto como una detección llega al radar. De esta manera, cada vez que se recibe una detección,

1. Se calcula la distancia entre la nueva detección y todas las posiciones predichas (`detectionToTrackAssignment`).
2. Se actualiza la pista asignada mediante la ecuación de actualización de estado (`updateAssignedTracks`).

Cada vez que se recibe un *TOP NORTE*,

1. Se predice la nueva posición de todas las pistas (`predictNewLocationsOfTracks`).
2. Se actualizan todas aquellas pistas que no han sido asignadas en la última iteración (`updateUnassignedTracks`).
3. Se eliminan las pistas que cumplen la condición de fin de vida (`deleteLostTracks`).
4. Se crean nuevas pistas tentativas con las detecciones no asignadas (`createNewTracks`).

3.3. Simulador

Se han desarrollado dos simuladores para probar el sistema. El primero genera trayectorias simples lineales, es utilizado para probar el sistema de hardware completo, incluyendo procesador primario, procesador secundario, combinador y trackeador.

El segundo genera trayectorias de forma aleatoria, y únicamente alimenta al *tracker*, es utilizado para verificar la algorítmica de seguimiento y parámetros de los filtros.

3.3.1. Simulador por Hardware

En la Figura 3.8, se observa la comunicación del simulador por hardware con el EDDR. El objetivo principal de este módulo es probar el conexionado entre todos los módulos del EDDR y que el flujo de información sea correcto.

El simulador (Generador de *Plots*) genera puntos en 8 de los 16 sectores de forma alterna, aumentando el rango en cada vuelta. De esta forma, se generan 8 trayectorias radiales salientes, como se observa en la Figura 3.9. Mediante el panel frontal del EDDR se selecciona si los datos de entrada a los procesadores primario y secundario serán del simulador o del radar. El simulador alimenta a los procesadores por interfaz serial.

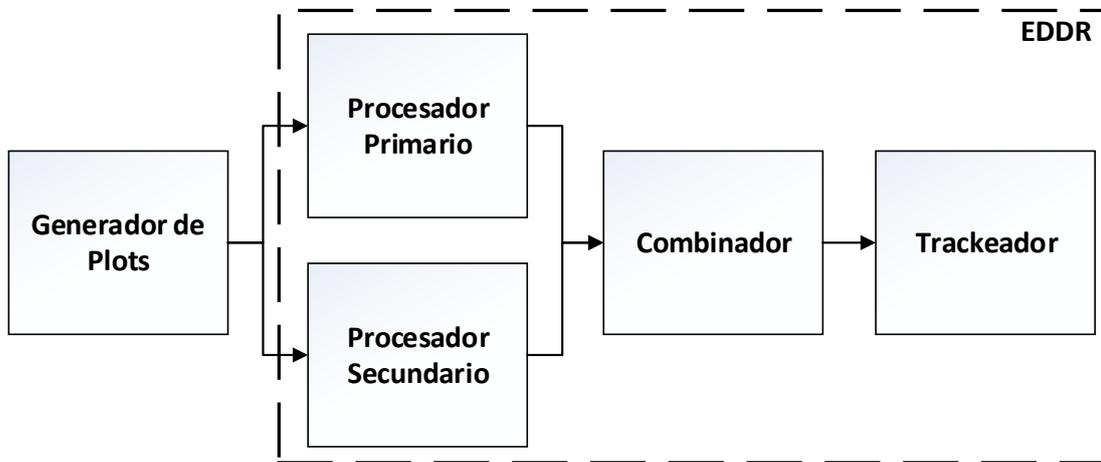


FIGURA 3.8: Diagrama de bloques del Simulador por Hardware (Generador de *Plots*) conectado al EDDR.

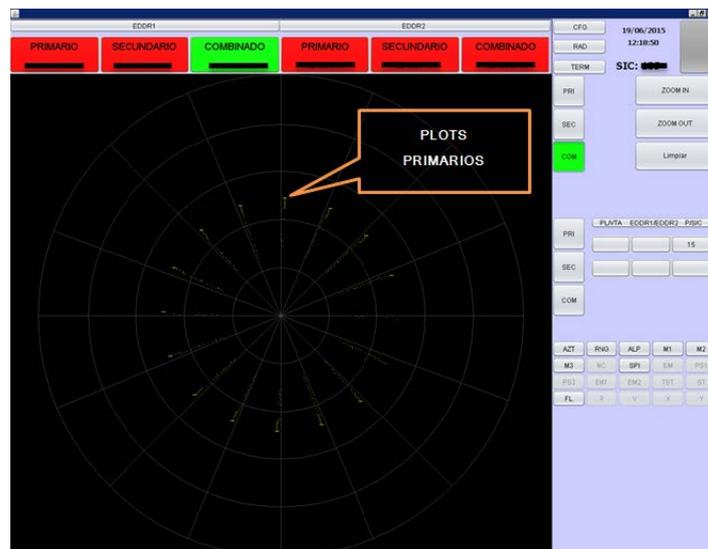


FIGURA 3.9: Trayectorias generadas por el simulador de hardware integrado al EDDR.

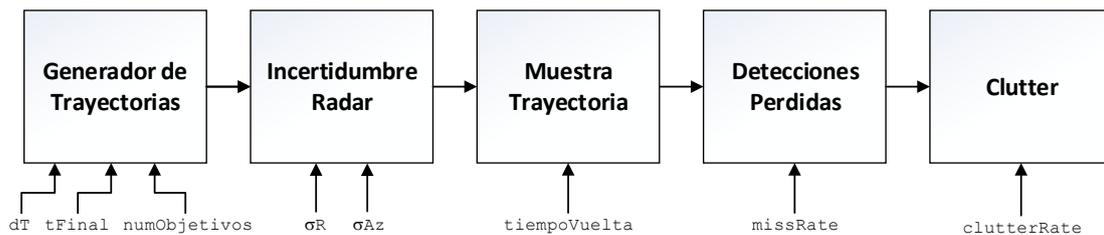


FIGURA 3.10: Diagrama de bloques del Simulador por Software (Generador de Trayectorias).

3.3.2. Simulador por Software

El simulador por software ha sido desarrollado en Matlab. Consta de,

- Un generador de trayectorias en dos dimensiones cartesianas (X, Y).
- Un módulo que agrega ruido en coordenadas polares (azimut y rango), emulando de esta forma las indeterminaciones en la medición de ángulo y rango del radar.
- Un módulo que muestrea la trayectoria una vez por vuelta del radar.
- Un proceso que elimina aleatoriamente muestras, para emular detecciones perdidas.
- Un proceso que agrega detecciones aleatorias, para emular *clutter*.

En la Figura 3.10 se muestra un diagrama de bloques del simulador con todos sus parámetros configurables. Se configura,

- Resolución de tiempo para la generación de la trayectoria.
- Tiempo total de simulación.
- Cantidad de objetivos a simular.
- Varianza de ruido en rango y azimut.
- Tiempo de vuelta del radar.
- Tasa de pérdida de datos.
- Tasa de *clutter*.

En las Figuras 3.11, 3.12 y 3.13 se muestran las salidas de corridas aleatorias del simulador.

Se han generado aleatoriamente 5 trayectorias, posteriormente se las muestra con ruido de medición. En el tercer gráfico (detecciones) se muestra el resultado de las detecciones para la siguiente configuración:

- Resolución temporal de las trayectorias: 0.1 s.
- Tiempo de simulación: 1000 s.
- Cantidad de objetivos: 5.

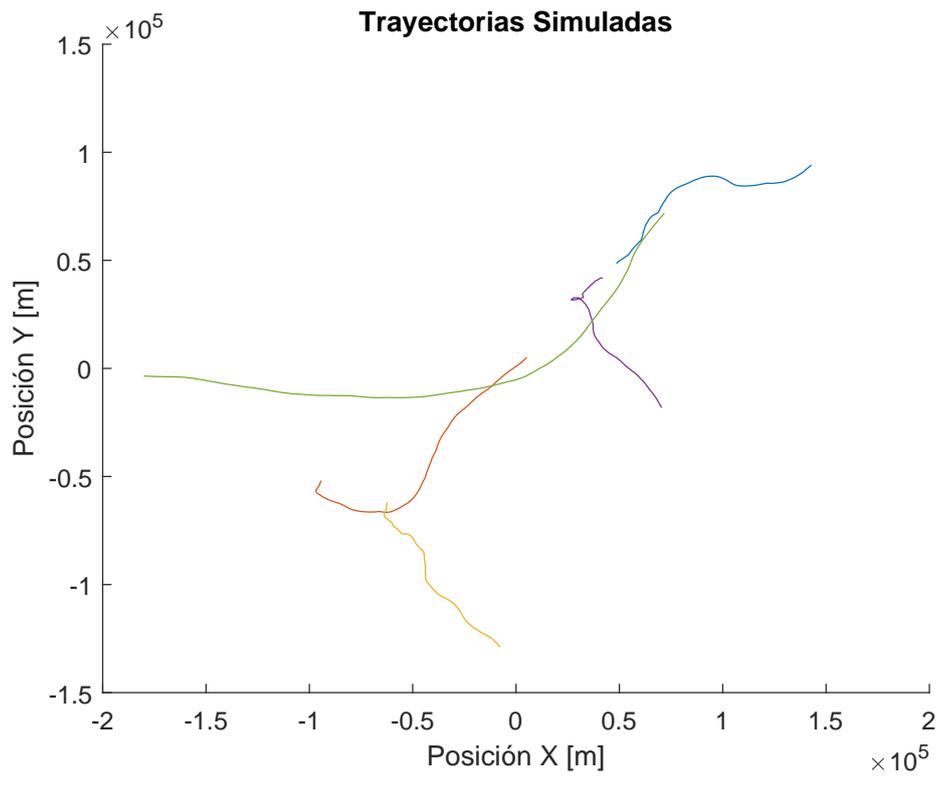


FIGURA 3.11: Trayectorias generadas con el simulador.

- Tiempo de vuelta: 12 s.
- *Miss Rate*: 30 % (se elimina el 30 % de las detecciones correspondientes a las trayectorias).
- *Clutter Rate*: 8 % (por vuelta de radar se agrega un 8 % de detecciones en posiciones aleatorias).
- Como se desarrolló en las Eqs. 3.15 y 3.16, $\sigma_\rho = 200$ mts y $\sigma_\theta = 0,6^\circ$.

En la Figura 3.14 se muestran las detecciones realizadas por el radar simulado. En este punto ya se ha perdido la información de identidad de cada detección. Será responsabilidad del algoritmo de *tracking* y asociación la de identificar qué detecciones corresponden a cuál trayectoria.

Los módulos de generación de trayectorias y de incertidumbre de radar han sido desarrollados en Simulink. En la Figura 3.15 se muestra la implementación del generador de trayectorias. Se genera en cada iteración un valor aleatorio de aceleración en X y en Y. Esta aceleración se integra para generar la velocidad y se integra nuevamente para obtener la posición en X e Y del objetivo virtual. Se parte de una velocidad y posición inicial, también aleatorias.

En la Figura 3.16 se muestra la implementación del módulo que agrega error de medición. Se convierte la posición simulada de coordenadas cartesianas a polares, luego se suma el ruido aleatorio a partir de los parámetros de configuración (σ_ρ y σ_θ – en radianes) y se convierte la posición obtenida nuevamente a coordenadas cartesianas.

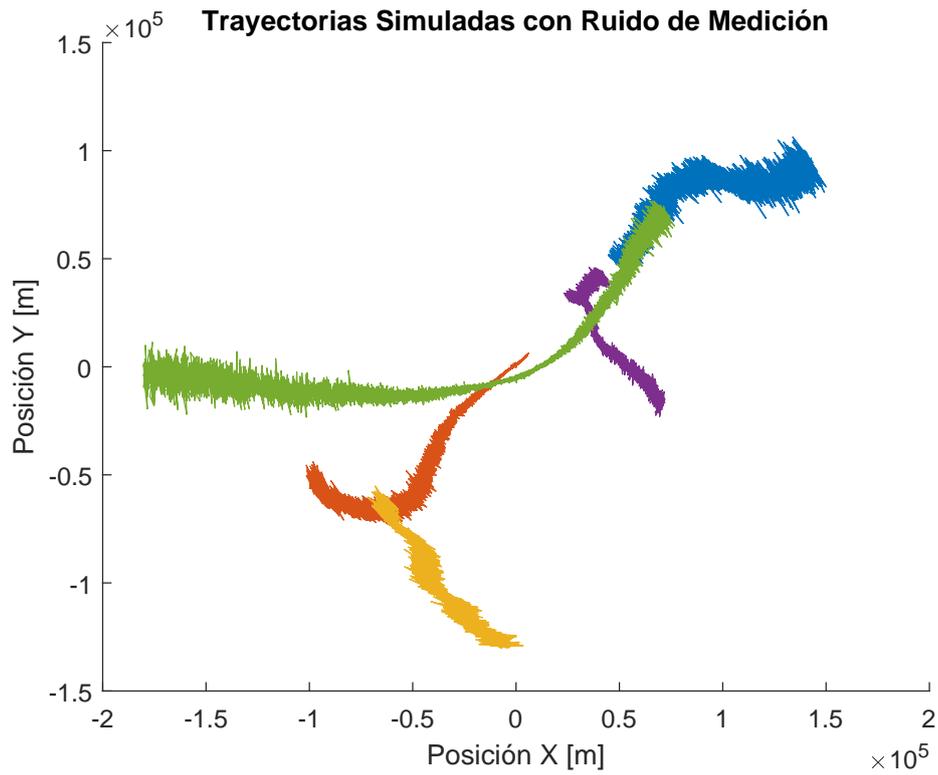


FIGURA 3.12: Trayectorias generadas con el simulador con el error de medición del radar ($\sigma_\rho = 200$ mts y $\sigma_\theta = 0,6^\circ$).

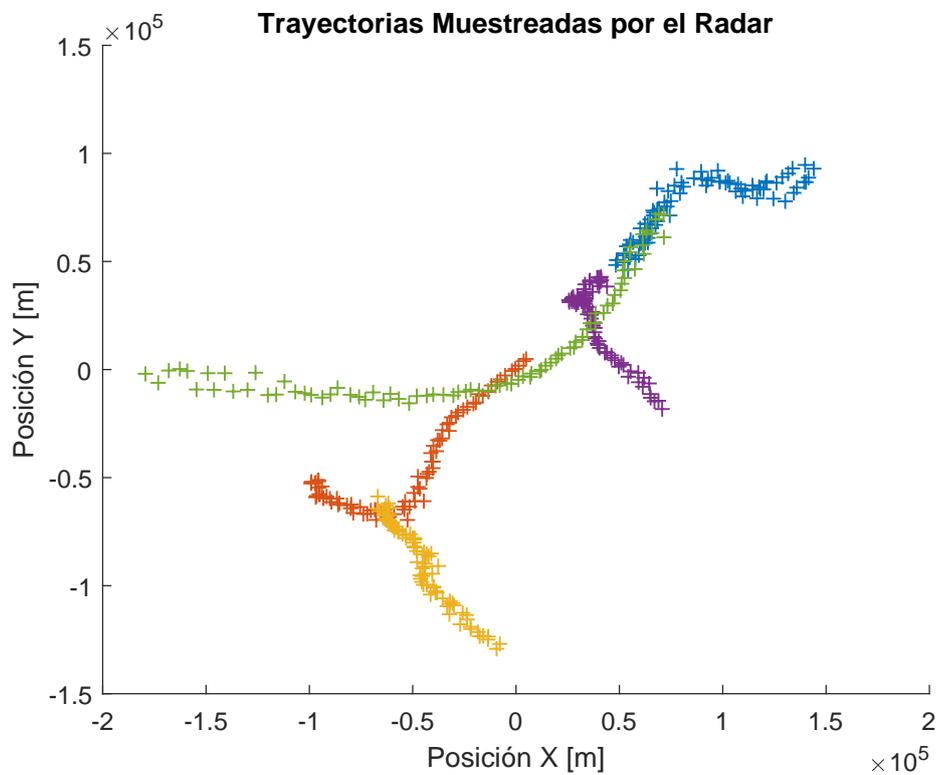


FIGURA 3.13: Trayectorias muestreadas una vez por vuelta de radar (12 segundos).

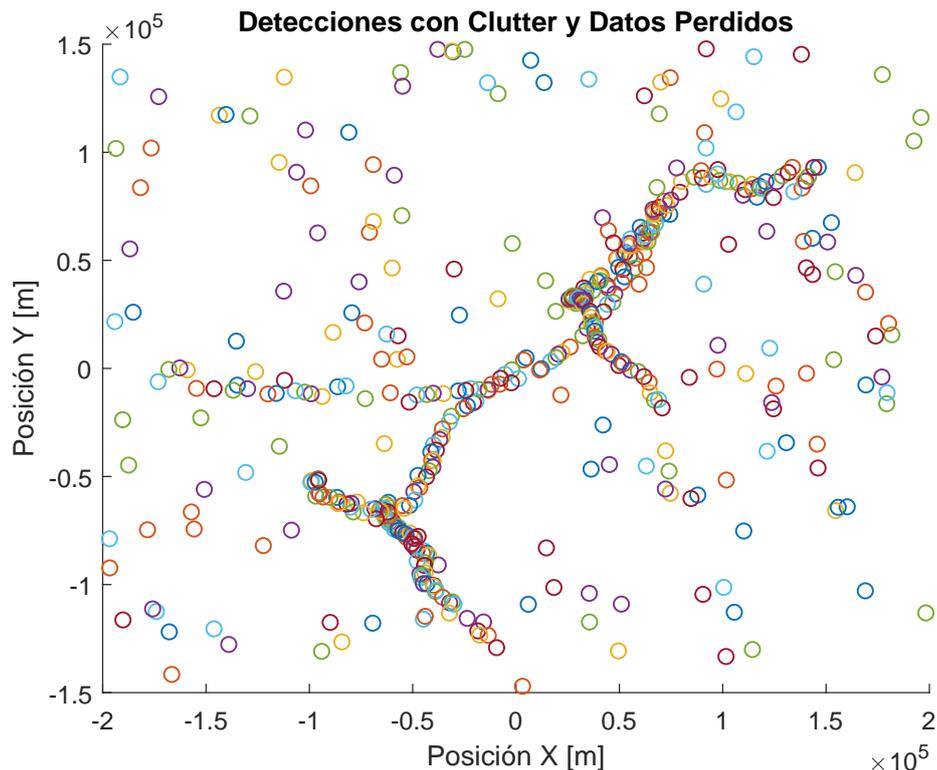


FIGURA 3.14: Detecciones realizadas por el radar simulado, con el 30% de las detecciones perdidas y 8% de clutter.

3.4. Implementación en Hardware

Una vez verificado el funcionamiento del *tracker* con el simulador por software, se pasa a la implementación en hardware, para que pueda ser integrado al EDDR. El *Tracker*, desarrollado inicialmente en Matlab, se migra a lenguaje C++ para que pueda ser ejecutado por la SBC (*Single Board Computer*) correspondiente al módulo Combinador-Trackeador del EDDR. Esta SBC es de la firma WinSystems, modelo LBC-GX500. En la Figura 3.17 se muestra una imagen de la placa utilizada como SBC. Sus especificaciones principales son las siguientes,

- Procesador AMD Geode GX500@1W (367 MHz).
- 256 MB de SDRAM, SODIMM PC2700 DDR.
- Compatible con memorias CompactFlash tipo I y II.
- Controlador Ethernet 10/100 Mbps Intel 82551ER PCI.
- Formato PC/104 con conectores de expansión PC/104-Plus.
- Compatible con Sistema Operativo GNU/Linux.
- Puertos seriales RS-232 tipo 16550A

La SBC corre un Sistema Operativo Debian Lenny (Kernel 2.6.26).

Estas características de Hardware y Software tienen que ser tenidas en cuenta para poder hacer una correcta implementación del sistema en el EDDR.

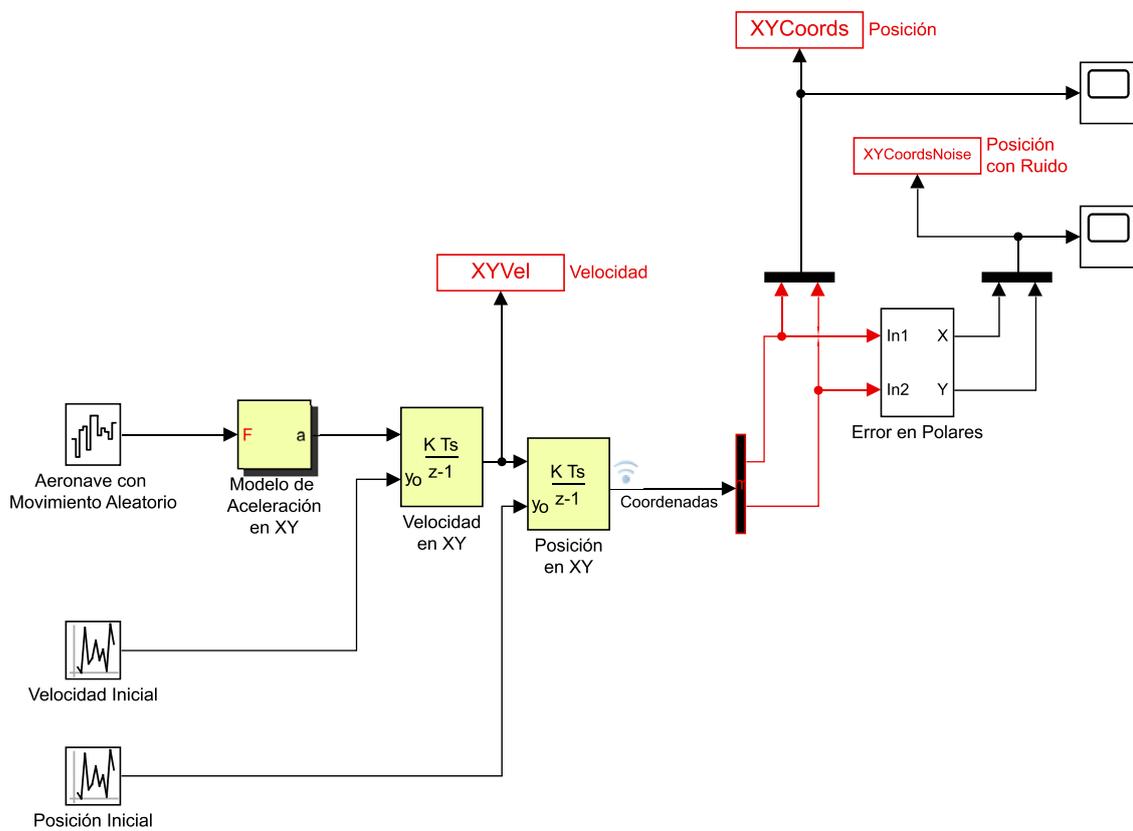


FIGURA 3.15: Diagrama de bloques de la implementación del simulador de trayectorias realizado en Simulink ®.

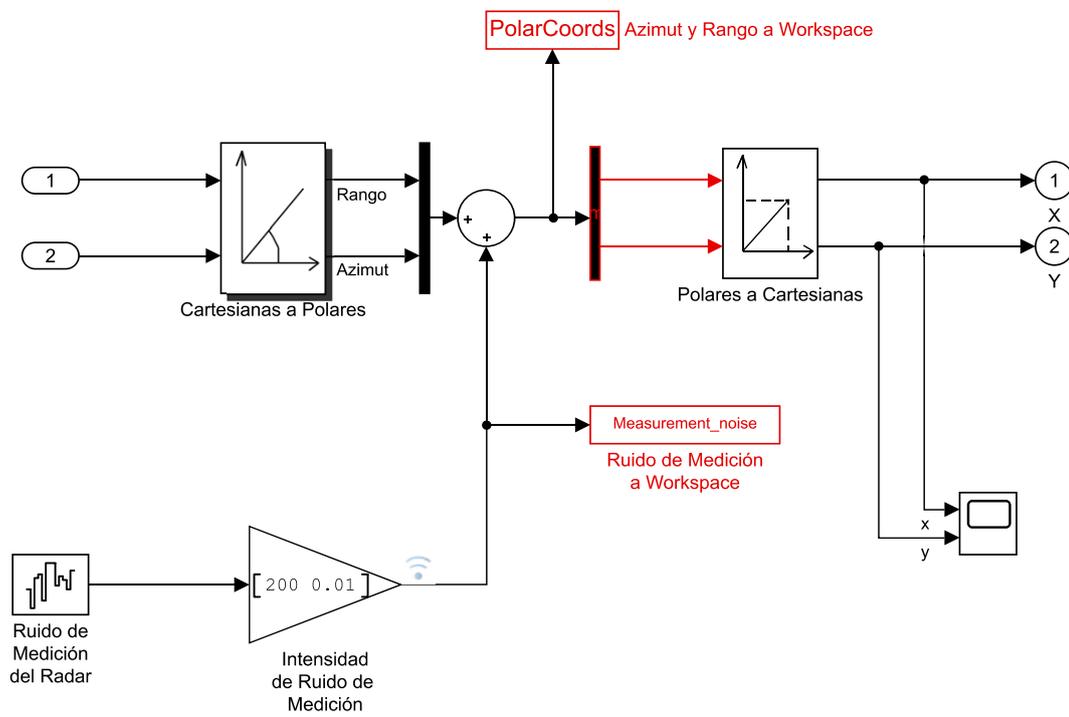


FIGURA 3.16: Adición de ruido de medición en coordenadas polares.

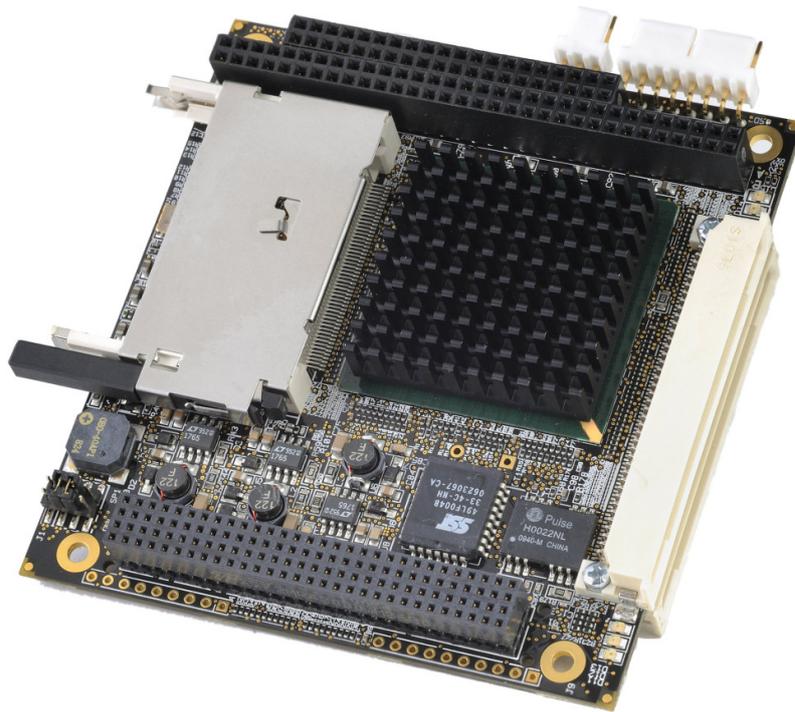


FIGURA 3.17: SBC LBC-GX500 sobre la que se ejecuta el Trackeador.

Por las características de Hardware del sistema, Sistema Operativo sobre el cual va a correr y sus requerimientos de tiempo real, se considera que C++ es el lenguaje mas apropiado para la implementación.

En la Figura 3.18 se muestra el diagrama de clases simplificado de la implementación en C++ del programa en la SBC. El programa consta de los siguientes módulos:

- CombTrack: Clase Principal.
- Configurador: Encargado de recibir del usuario todos los parámetros configurables.
- Receptor: Consta de dos hilos, cada uno escuchando un puerto serie correspondientes al procesador primario y secundario. Recibe de éstos los plots y los comunica al Combinador.
- Combinador: Contiene los *buffers* de Blancos Primarios y Secundarios y realiza el proceso de combinación. Posteriormente mediante el método *empaquetar* genera la trama Asterix correspondiente a los datos combinados. El Combinador le comunica al Tracker los *plots* procesados.
- Tracker: Contiene todos los métodos y estructuras para realizar el proceso de asociación y filtrado. Contiene la lista de pistas vigentes (`tracks:list<track_struct>`), la lista de asignaciones (`assignments:list<match>`) y la lista de detecciones sin asignar (`unassignedCentroids:list<centroid>`).
- Transmisor: Envía por Puerto Serie y por Ethernet las tramas generadas.

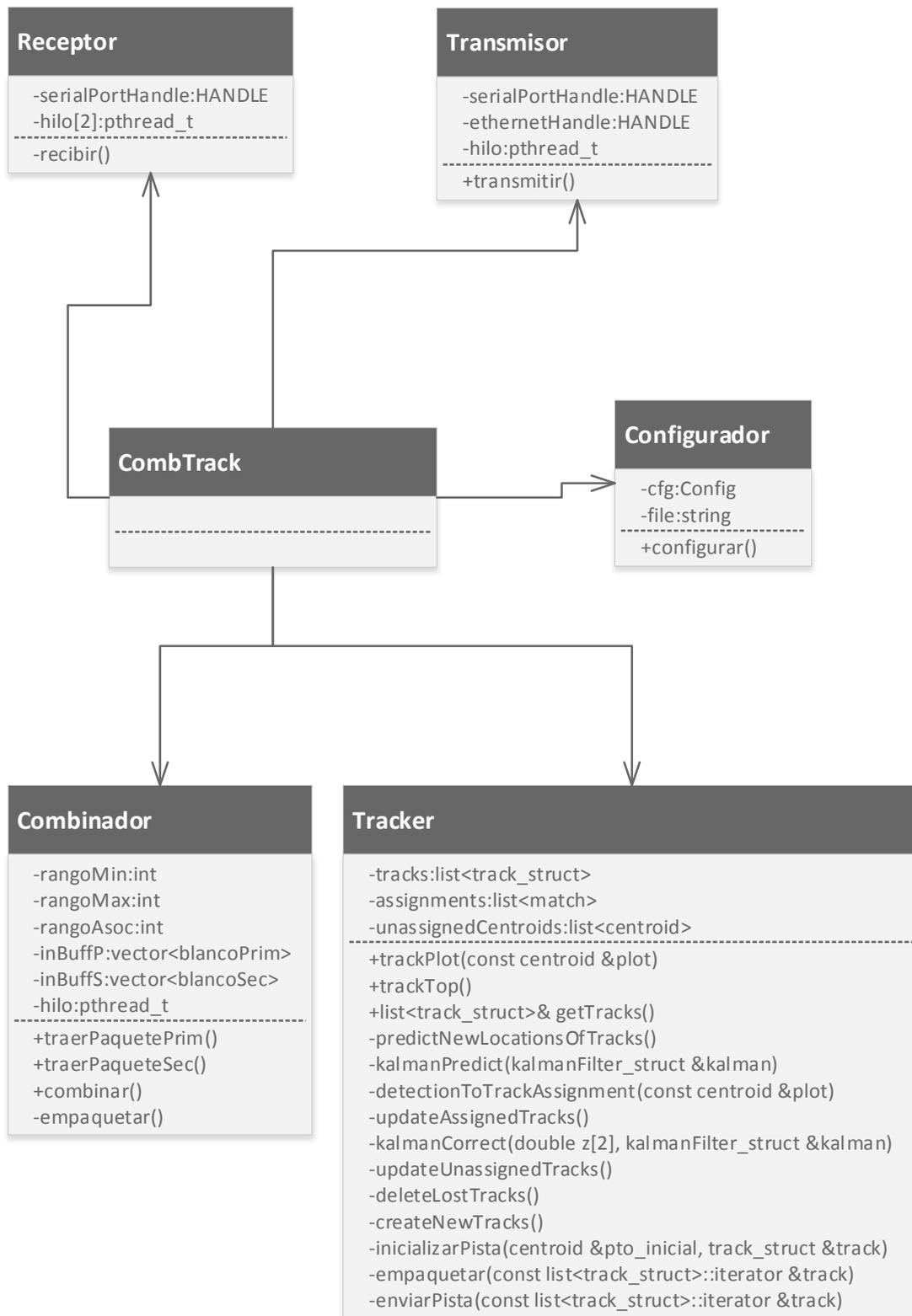


FIGURA 3.18: Diagrama de Clases simplificado del software en C++ CombTrack.

El programa al ejecutarse, inicia las conexiones con los puertos de lectura de procesadores primarios y secundarios, y queda a la espera de recepción de *plots* o eventos de TOP NORTE. En la Figura 3.19 se ilustra la secuencia de acciones que desencadena el evento de recepción de plot primario por parte del procesador primario. El hilo receptor se despierta por la llegada de un evento desde el procesador primario (`recibir()`). Este dispara el inicio de acciones por parte del combinador (`traerPaquetePrimario()`). Para mas detalles del proceso realizado por el programa combinador, referirse a (Montamat, 2015). El combinador, después de distinguir la recepción como un plot válido, llama al *tracker* a través de la secuencia de acciones `trackPlot()`. En este caso, se ejecutarán los procesos explicados en los puntos 2 y 3 de la Figura 3.7. Finalmente, se envían las tramas por red en formato Asterix.

Muy similar es el proceso iniciado por el evento de recepción de plot secundario. Este se ilustra en la Figura 3.20.

Cuando se recibe el Top Norte, se desencadena la secuencia ilustrada en la Figura 3.21. El combinador lee la hora del GPS para utilizar como marca de tiempo y como sincronización del sistema. Posteriormente llama al *tracker* a través de `trackTop()`. En este punto se ejecutan los procesos explicados en los puntos 1, 4, 5 y 6 de la Figura 3.7.

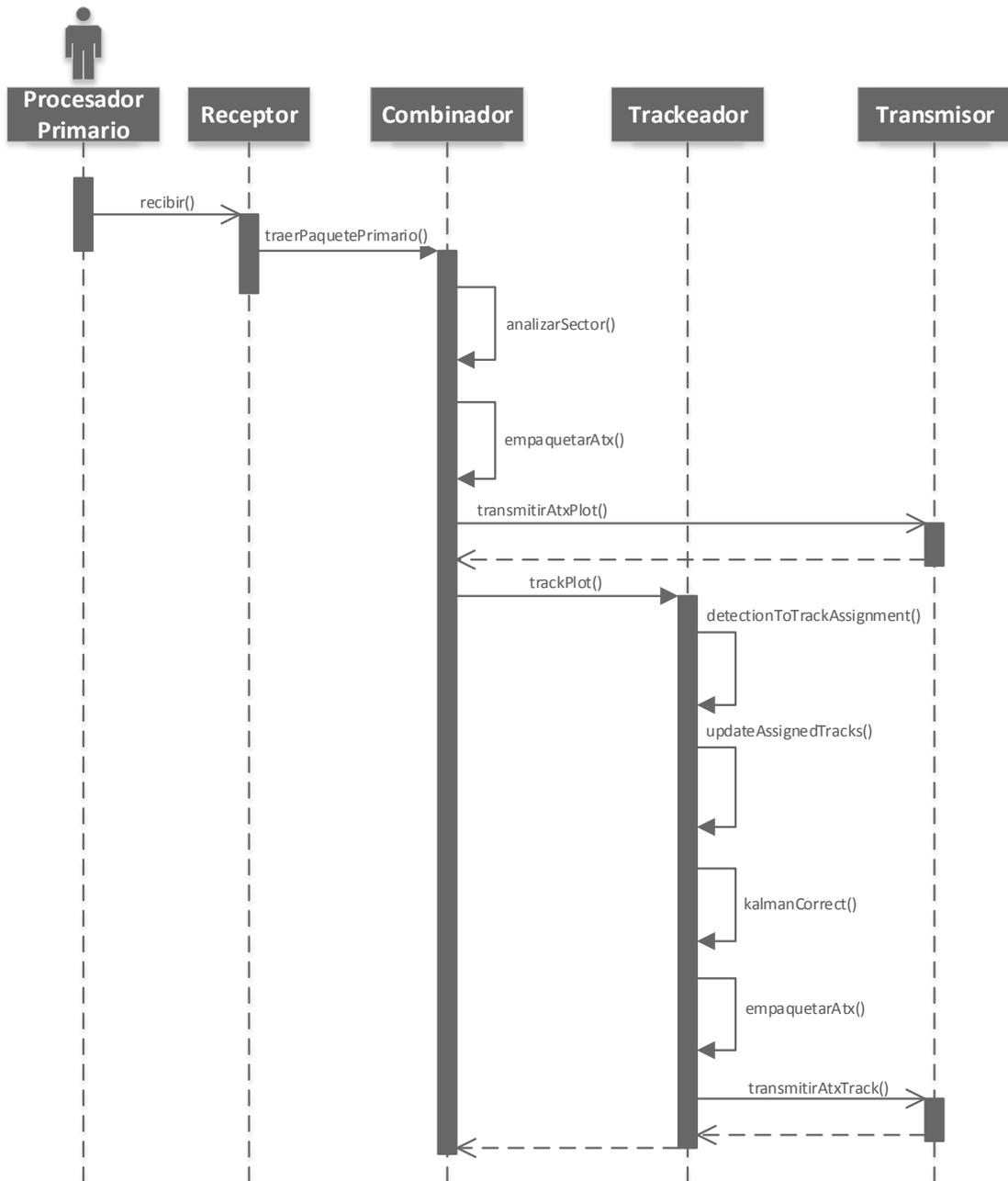


FIGURA 3.19: Diagrama de secuencia para el evento de recepción de *Plot* Primario.

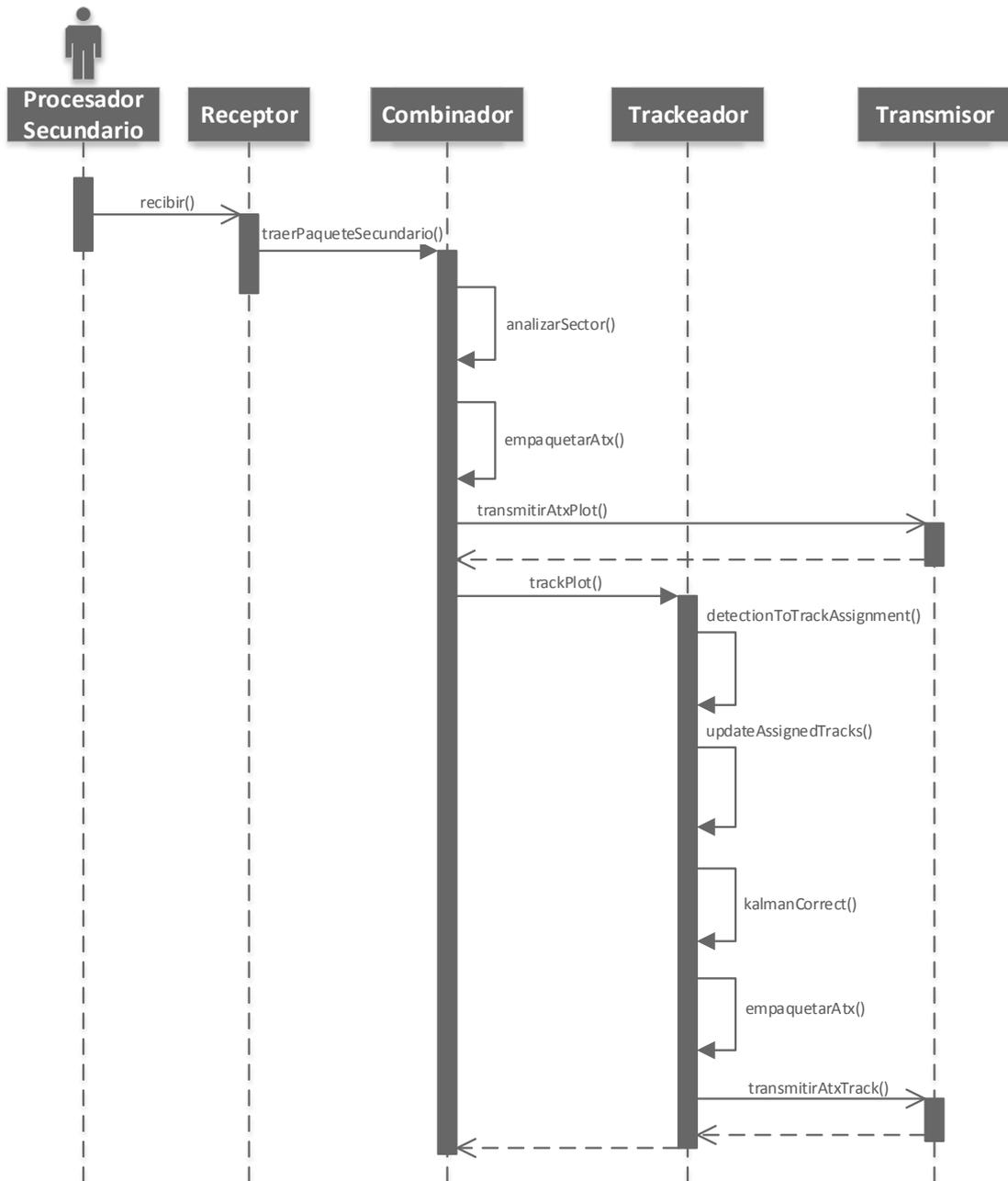


FIGURA 3.20: Diagrama de secuencia para el evento de recepción de *Plot* Secundario.

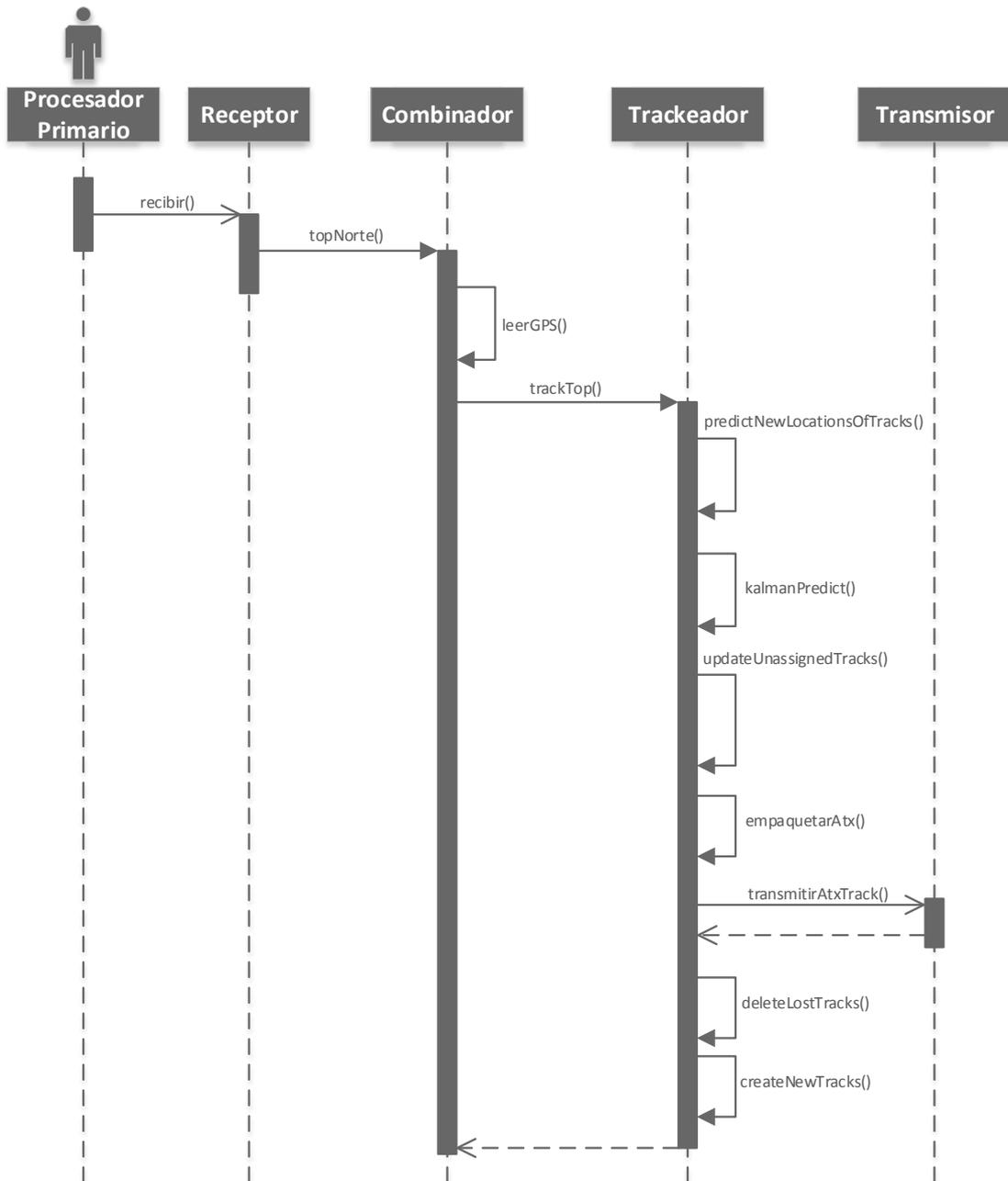


FIGURA 3.21: Diagrama de secuencia para el evento de recepción de Top Norte.

Capítulo 4

Resultados

Se hará un recorrido por las diferentes partes del sistema, mostrando resultados a partir de datos simulados, o reales cuando sea posible.

4.1. Filtro de Kalman

Se comenzará por la parte central del desarrollo, que es el filtrado de los datos de una trayectoria.

4.1.1. Trayectoria simulada sin maniobra en 2D

Se utiliza el Simulador descrito en la sección 3.3.2. Se genera en primer lugar una única trayectoria, con velocidad constante, a la que se le agrega ruido de medición del radar según los parámetros especificados anteriormente, $\sigma_\rho = 200$ mts y $\sigma_\theta = 0,6^\circ$. Estos datos de entrada al filtro se muestran en la Figura 4.1. La línea gruesa central corresponde a la trayectoria proporcionada por el simulador, dibujada sobre los datos contaminados por ruido de medición. Se observa que la dispersión aumenta a medida que la trayectoria se aleja del radar (ubicado en las coordenadas $[0, 0]$), debido a la componente de azimut de la incertidumbre del radar.

El filtro se configura con los siguientes parámetros iniciales,

$$R = \begin{bmatrix} 200^2 & 0 \\ 0 & 0,01^2 \end{bmatrix} \quad (4.1)$$

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & (1e^{-3})^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & (1e^{-3})^2 \end{bmatrix} \quad (4.2)$$

Los únicos 2 elementos no nulos de la matriz Q en la Eq. 4.2, indican la incertidumbre que acepta el sistema para la aceleración en X y en Y entre iteración e iteración. Las iteraciones tienen un Δt de 12 segundos, correspondientes a una detección por cada vuelta del radar virtual.

Cuando se inicia la pista, hay que dar valores al Estado Estimado en tiempo 0 ($\hat{x}_{0|0}$) y a la Covarianza de Estado en tiempo 0 ($P_{0|0}$).

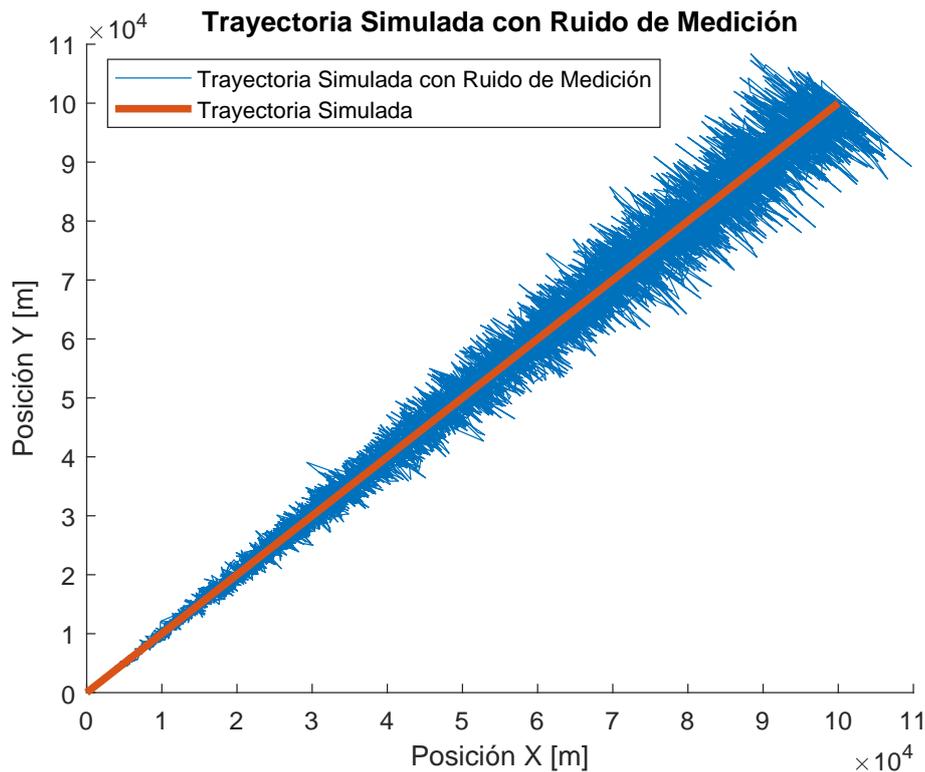


FIGURA 4.1: Una única trayectoria de velocidad constante generada por el simulador.

$$\hat{x}_{0|0} = \begin{bmatrix} z_0(x) \\ 0 \\ 0 \\ z_0(y) \\ 0 \\ 0 \end{bmatrix} \quad (4.3)$$

La matriz de estado se inicializa con la posición en X e Y correspondiente a la medición. Las componentes correspondientes a velocidad y aceleración se inicializan en 0 ya que a partir de una única detección no es posible conocer el sentido o signo de estas magnitudes, dando valor a la media cero y varianza alta.

$$P_{0|0} = \begin{bmatrix} 0,1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0,1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 \end{bmatrix} \quad (4.4)$$

La Eq. 4.4 indica que al inicio del filtro, la incertidumbre en la posición será baja (ya que parte de la detección realizada), pero que la incertidumbre en las componentes de velocidad y aceleración será alta, ya que se le ha dado valor 0.

En la Figura 4.2 se observan las detecciones de entrada al filtro (un dato cada 12 segundos) como puntos discretos. La pista filtrada por el algoritmo de kalman con los parámetros ya explicados se muestra en una traza continua.

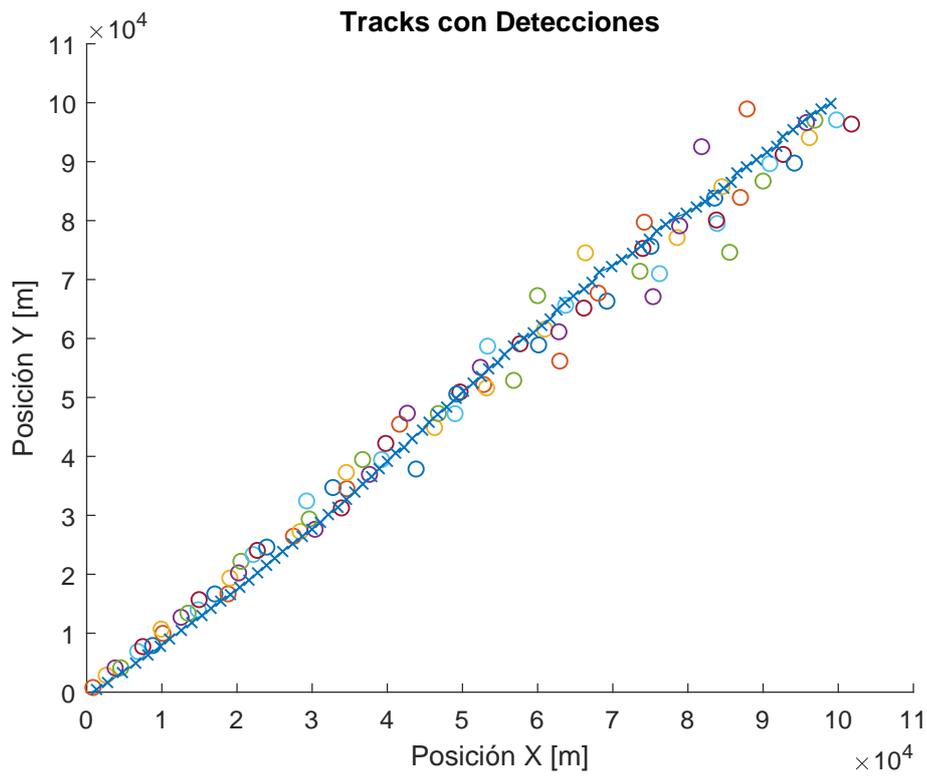


FIGURA 4.2: Detecciones correspondientes a una única trayectoria de velocidad constante junto con la pista filtrada.

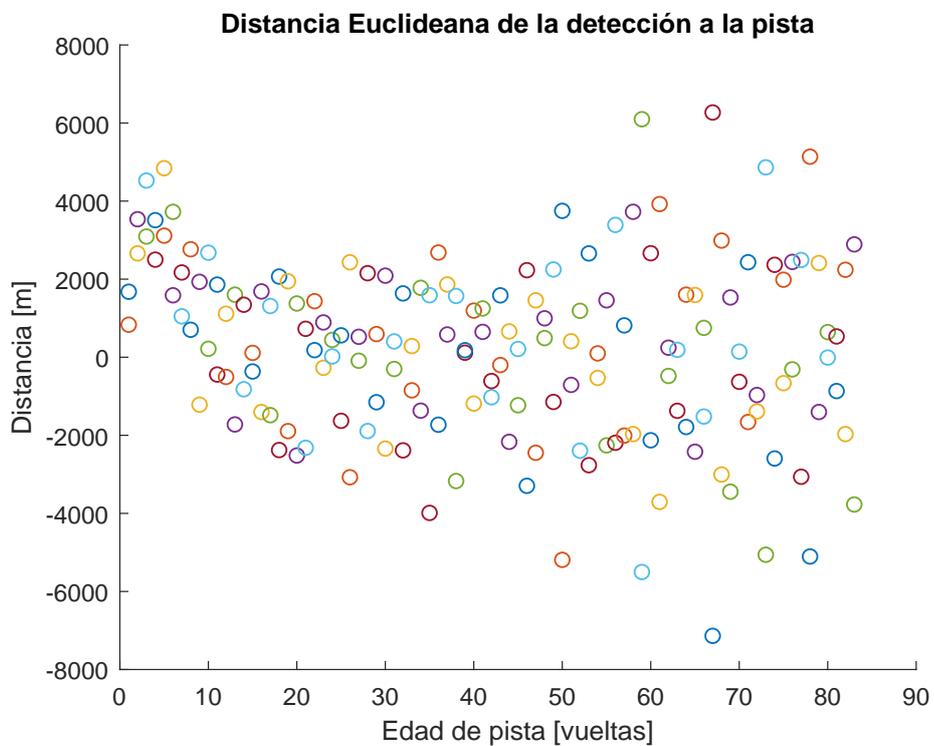


FIGURA 4.3: Distancia Euclidiana medida entre la detección y la posición predicha de la trayectoria según el filtro, a medida que avanza la pista.

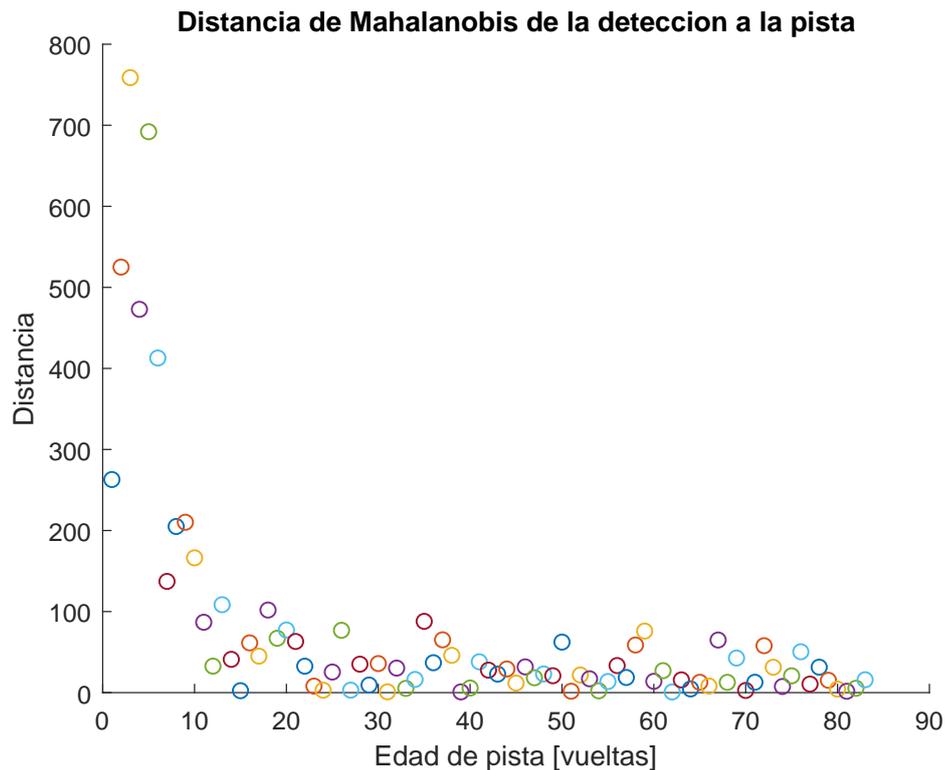


FIGURA 4.4: Distancia Estadística entre la detección y la posición predicha de la trayectoria según el filtro.

Entrando en más detalle en el funcionamiento interno del sistema, se puede observar en la Figura 4.3 un gráfico que muestra la distancia entre cada detección efectuada y la predicción de la posición de la pista en ese instante. En abscisas se representa el número de detección, o edad de la pista, que para este caso es la misma magnitud. Se observa que a medida que avanza el tiempo, debido a que el objetivo simulado se está alejando, la dispersión de la distancia aumenta. Esto se debe a la componente de error angular de la medición del radar.

Sin embargo, el comportamiento es diferente si consideramos la distancia estadística o distancia de Mahalanobis. En la Figura 4.4 se representa esta magnitud. Se observa una rápida tendencia decreciente de la distancia a medida que avanza el filtro, debido al ajuste de sus matrices de covarianzas. Esto indica que a pesar de que la dispersión de los datos de entrada aumenta, el filtro está bien ajustado para la condición simulada.

En la Figura 4.5 se puede observar el desempeño del algoritmo. Se muestra el error en las coordenadas X e Y de la trayectoria entre la estimación producida por el filtro de kalman y la trayectoria generada por el simulador. Se observa un error grande inicial que disminuye a medida que avanzan las iteraciones. Al igual que el caso de la distancia de las detecciones, el comportamiento del filtro mejora a medida que tiene mas datos para actualizar sus matrices de estado y covarianzas.

En la Figura 4.6 se muestra la velocidad estimada por el filtro en las coordenadas X e Y. Se observa que a medida que avanza la simulación la velocidad tiende a 100 m/s en ambas coordenadas.

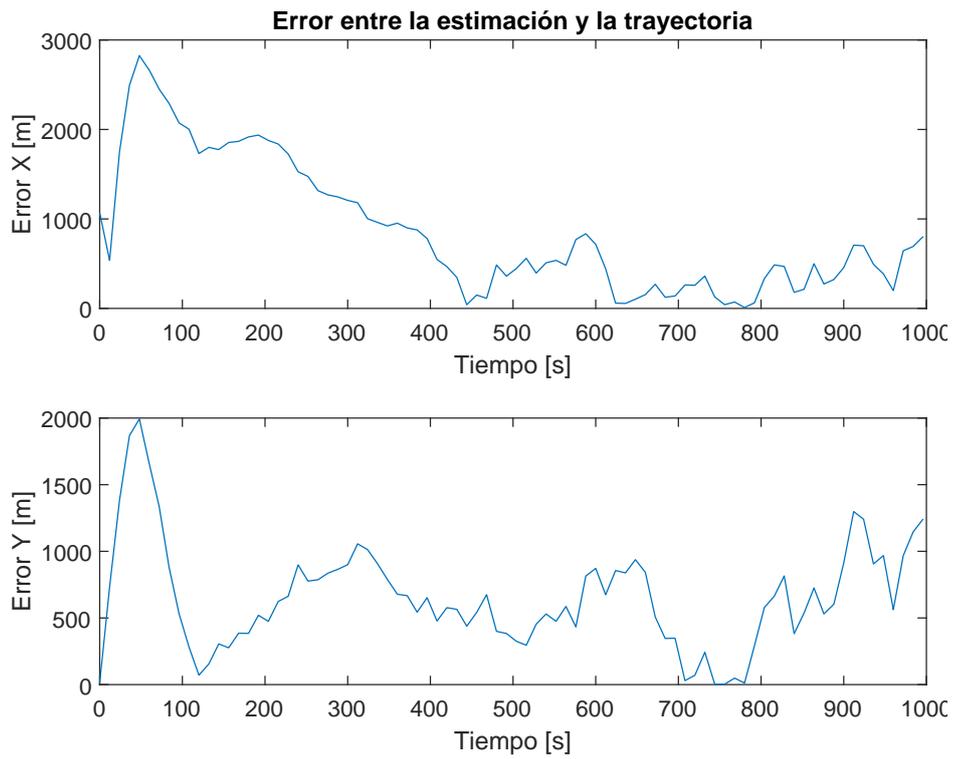


FIGURA 4.5: Error de estimación con respecto a la trayectoria simulada.

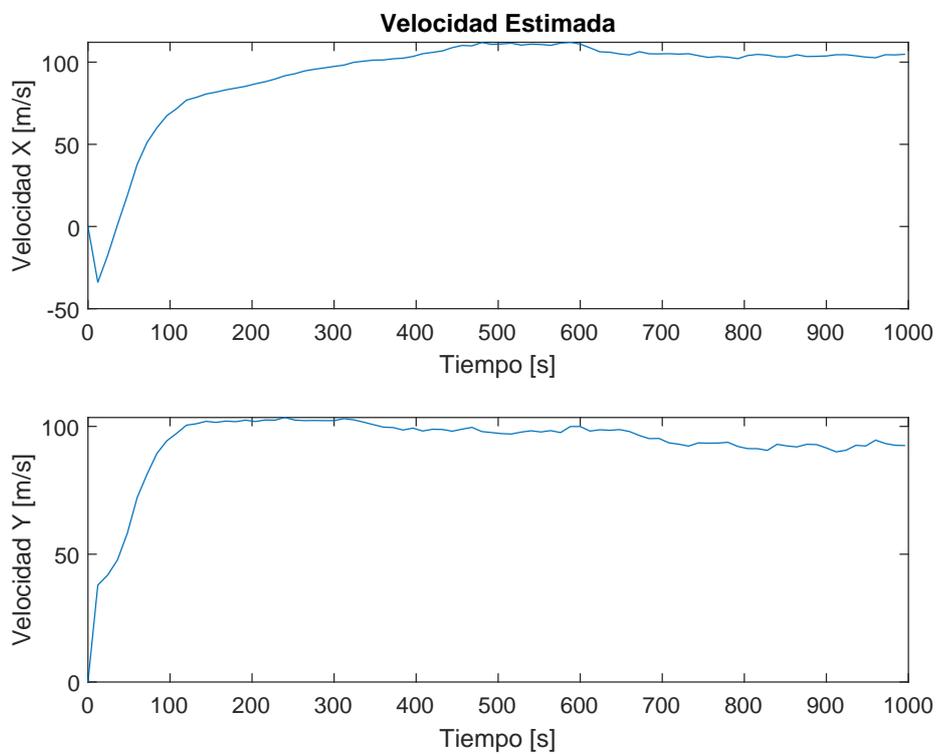


FIGURA 4.6: Velocidad estimada por el filtro.

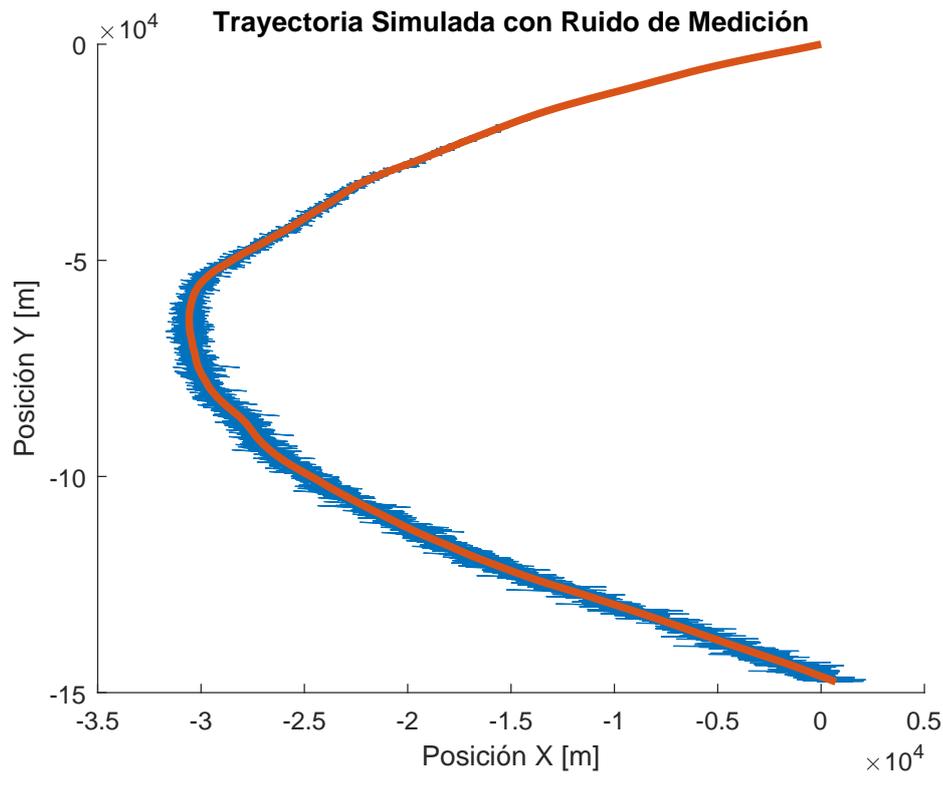


FIGURA 4.7: Trayectoria simulada con capacidad de maniobra.

4.1.2. Trayectoria simulada con maniobra en 2D

Se utiliza a continuación el simulador para generar la trayectoria de un objetivo con capacidad de maniobra aleatoria. En la Figura 4.7 se muestra esta trayectoria. El movimiento se puede interpretar como una aeronave comercial haciendo un cambio de rumbo a baja velocidad.

En la Figura 4.8 se muestra la salida del filtro para esta simulación. En la Figura 4.9 se grafican superpuestos los errores de estimación y de detección. Se observa que los mayores errores de estimación se corresponden a grandes errores de detección. A medida que aumenta el tiempo, debido a que el objetivo se aleja en rango, los errores de detección aumentan considerablemente, mientras que los errores de estimación se mantienen dentro del promedio.

4.1.3. Trayectoria real sin maniobra en 2D

Para este caso de prueba se utilizarán datos proporcionados por el radar de trayectografía Vitro RIR 778C, nombrado en el segundo capítulo. Este radar ha sido recuperado en el Centro de Investigaciones Aplicadas de la Fuerza Aérea Argentina (S. A. Rodríguez y Anún, 2011). Este radar fue utilizado a fin de la década de 1980 en los programas de coherencia de la FAA. Quedó en desuso desde 1990 y con su computadora central fuera de servicio (S. A. Rodríguez, Anún y Menara, 2013). A partir de programas de financiación del ministerio de defensa, entre el año 2009 y 2018 se logró recuperar su estado de operación. Se construyó una nueva computadora, sus interfaces con todos los subsistemas, se desarrolló el software de aplicación y de seguimiento. Los filtros desarrollados en este trabajo son parte de los algoritmos que ejecuta este radar.

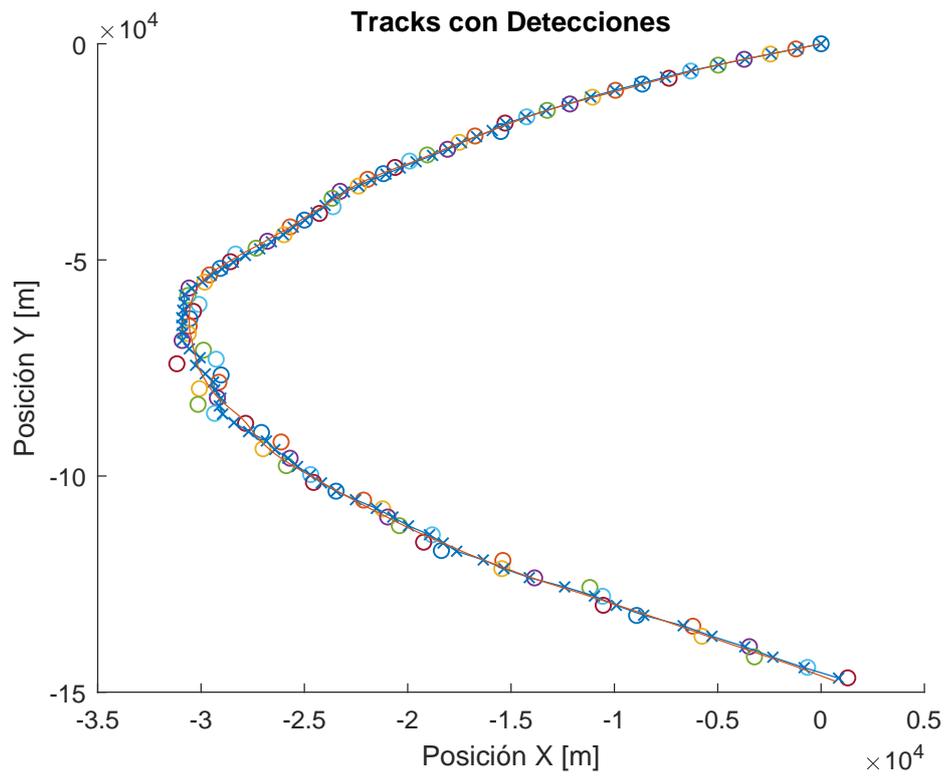


FIGURA 4.8: Trayectoria simulada con capacidad de maniobra, filtrada, con sus detecciones correspondientes.

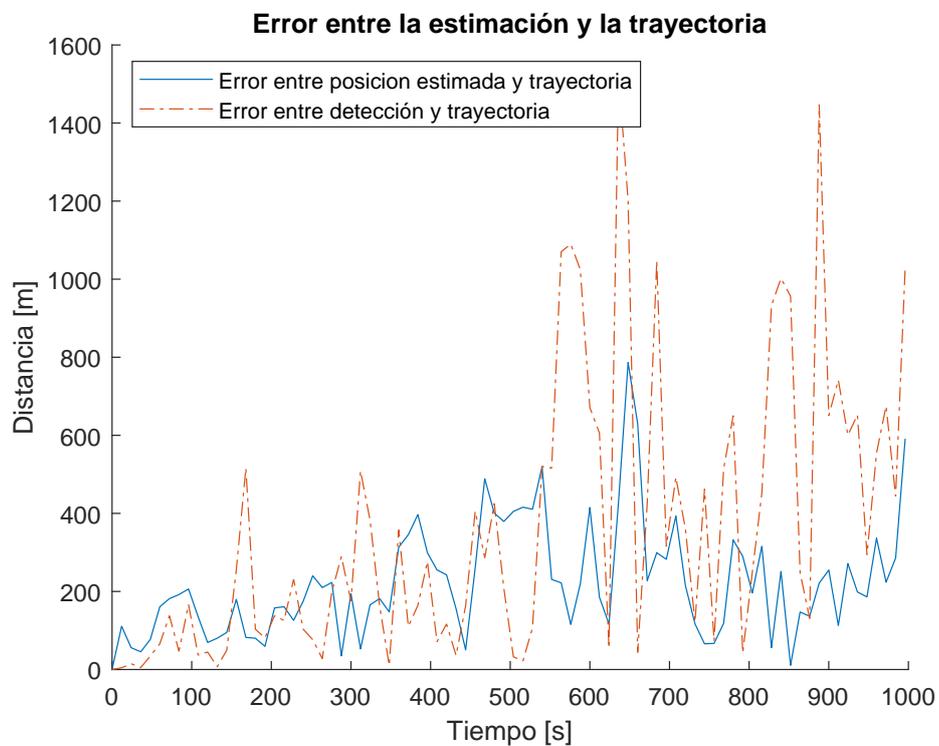


FIGURA 4.9: En línea continua se muestra la distancia entre cada estimación y el objetivo simulado. En línea punteada, distancia entre detecciones y objetivo.



FIGURA 4.10: Ruta seguida por los vuelos entre Sao Paulo (Brasil) y Santiago de Chile. Información proporcionada por la plataforma FlightRadar24.

La prueba a la que se hará referencia en esta sección, corresponde al seguimiento del vuelo de una aeronave comercial realizado el 15/11/2018 desde el Centro de Investigaciones Aplicadas de la Fuerza Aérea Argentina, en Córdoba, lugar donde está emplazado este radar. La aeronave fue un Boeing 737 haciendo un vuelo desde Sao Paulo, Brasil a Santiago de Chile. Esta ruta pasa por Córdoba, por lo que es un buen escenario de pruebas para este radar. Además, en el tramo en el que la ruta puede ser visible por el radar, se puede considerar que la trayectoria no tiene maniobra, por lo que es una buena prueba inicial.

En la Figura 4.10 se muestra la trayectoria que sigue generalmente esta ruta según el software FlightRadar24. Se observa el tipo de avión, hora de pasada y longitud de la ruta. Se ve también que la ruta pasa exactamente sobre la ciudad de Córdoba. En la Figura 4.11 se observan las detecciones del radar junto con la trayectoria filtrada formando una pista. La tasa de datos de este radar es de 50 Hz, o 20 ms. Esto permite una resolución de seguimiento mucho mayor a la lograda con el radar Bendix, que tiene un tiempo de vuelta de 12 segundos. El radar Vitro se ubica en el origen de las mediciones, indicado con el rombo, en las coordenadas [0, 0].

Se puede ver en la Figura 4.12 el tramo inicial de la trayectoria ampliada. Se observa el tiempo que lleva al filtro establecerse hasta que puede predecir correctamente el comportamiento del objetivo. En la Figura 4.13 se amplía la parte final de la trayectoria donde se ve que el ruido de detección es mayor. La pista sigue su trayectoria suavizada.

En la Figura 4.14 se muestra la velocidad estimada por el filtro para este caso. Se observa una velocidad de 800 Km/h aproximadamente, que coincide con la velocidad reportada por la aeronave en la plataforma FlightRadar24. Se puede ver que en aproximadamente 10 segundos el filtro converge a partir de su velocidad inicial establecida de 0 m/s.

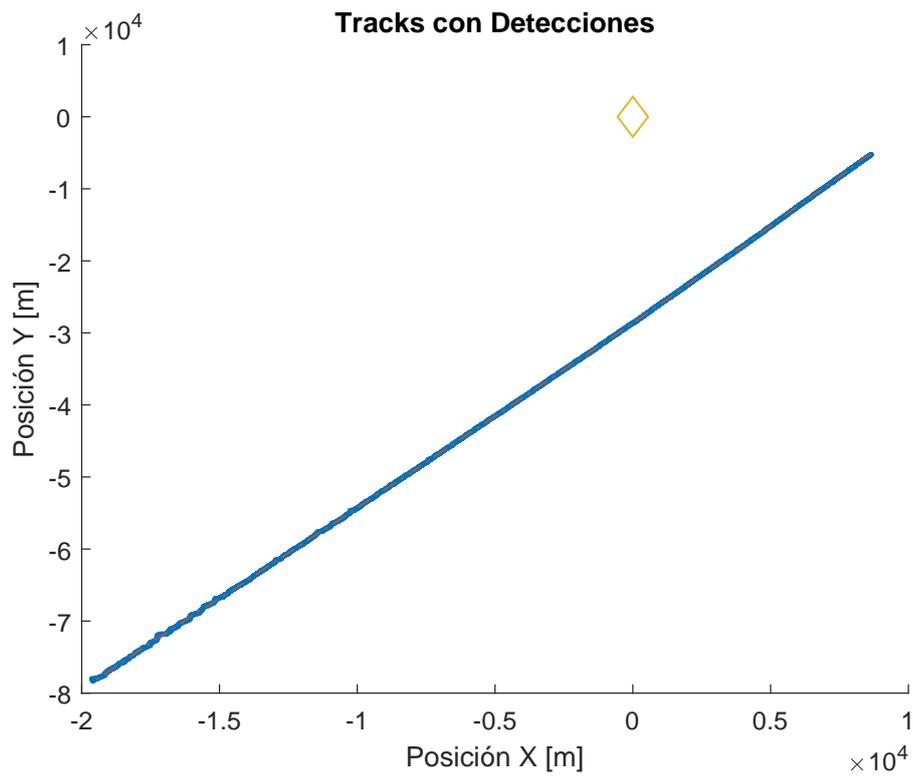


FIGURA 4.11: Trayectoria filtrada del Boeing 737 junto con sus detecciones. El rombo indica la posición del radar.

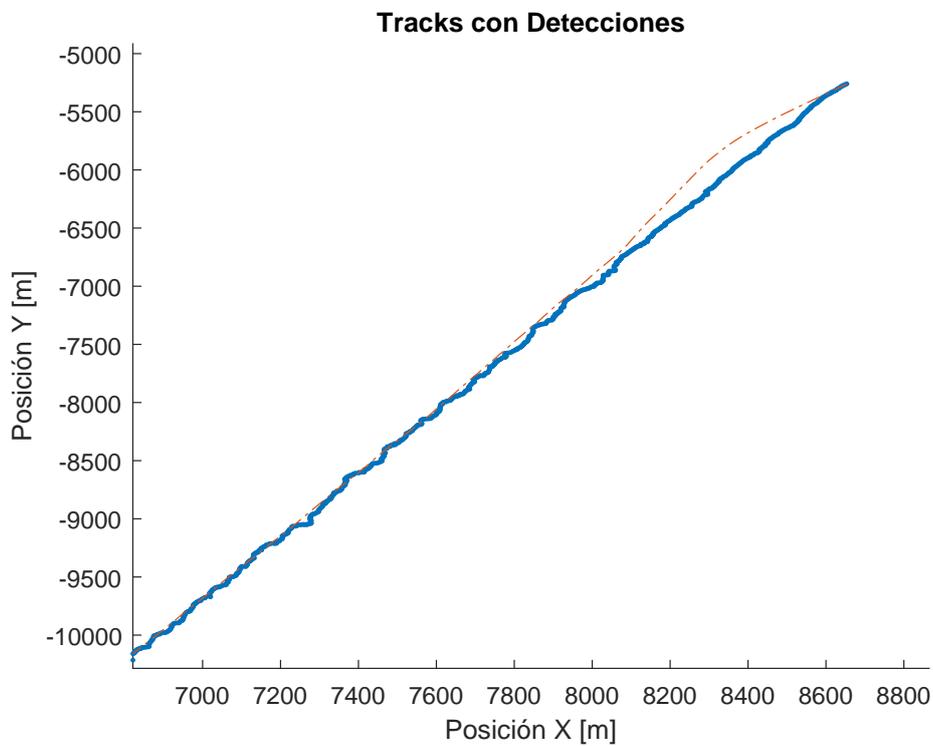


FIGURA 4.12: Trayectoria del Boeing 737 en el inicio de la pista.

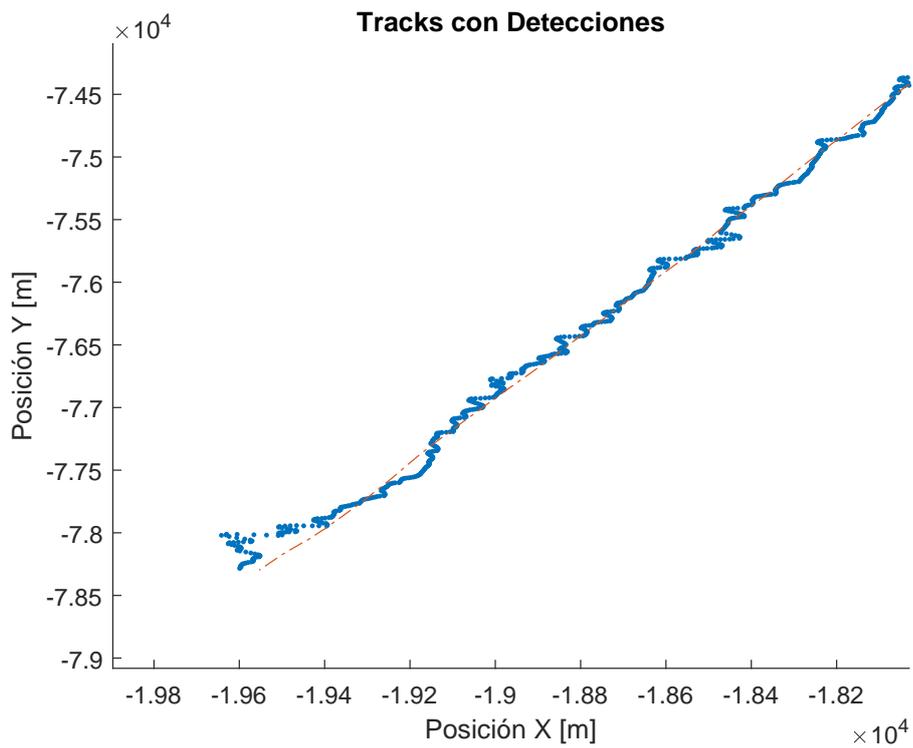


FIGURA 4.13: Trayectoria filtrada del Boeing 737 en el final de la pista.

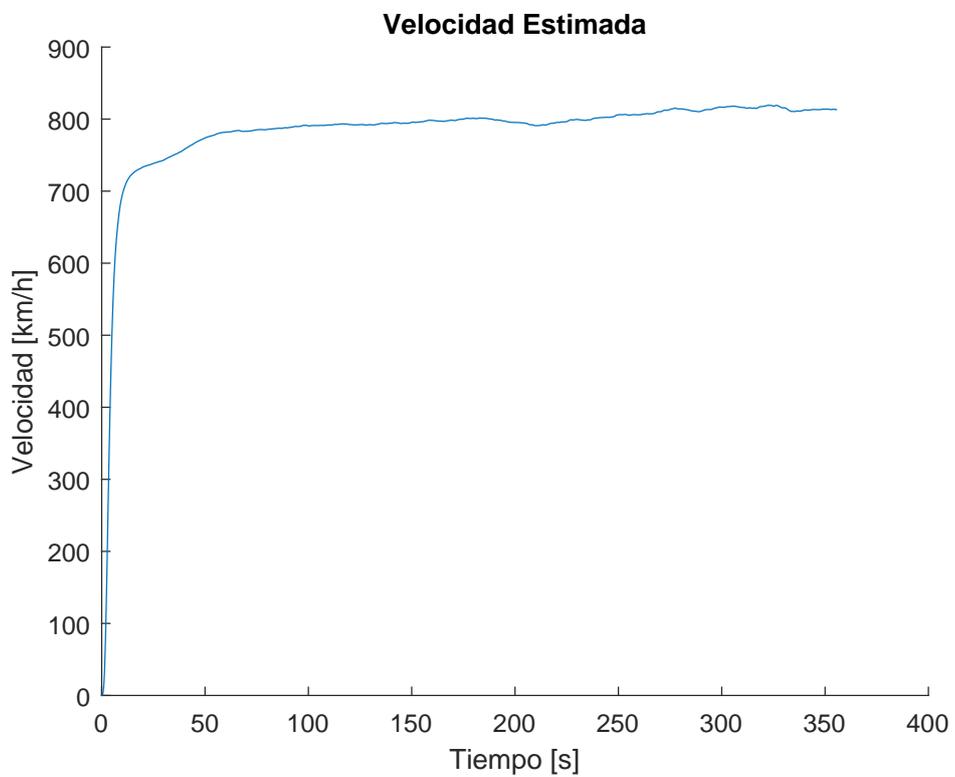


FIGURA 4.14: Trayectoria filtrada del Boeing 737 en el final de la pista.

4.1.4. Trayectoria real sin maniobra en 3D

Para continuar el análisis de los datos del caso anterior, la aeronave con destino a Santiago de Chile, se hace una extensión al filtro de Kalman desarrollado para hacer el seguimiento en 3 dimensiones. El radar Vitro RIR 778C, es un radar 3D, a diferencia del radar Bendix. Se puede aprovechar esta capacidad para hacer un mejor seguimiento.

Se muestran a continuación las matrices utilizadas para extender el filtro para un seguimiento en 3D.

La Matriz de Transición de Estado se extiende a,

$$F = \begin{bmatrix} 1 & T & \frac{T^2}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & \frac{T^2}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & T & \frac{T^2}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.5)$$

En donde $T = 0,02$ s. Esta es la principal característica que hace a un radar de *Tracking*, la alta tasa de datos.

La Matriz de Medición,

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (4.6)$$

Indicando que ahora se tomarán mediciones de la posición en las coordenadas X, Y y Z.

La Matriz de Covarianza del ruido de proceso,

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{ax}^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{ay}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{az}^2 \end{bmatrix} \quad (4.7)$$

Permite ahora aceleración en la componente Z. La Matriz de Incertidumbre de la medición,

$$R = \begin{bmatrix} \sigma_\rho^2 & 0 & 0 \\ 0 & \sigma_\theta^2 & 0 \\ 0 & 0 & \sigma_\varphi^2 \end{bmatrix} \quad (4.8)$$

Agrega el parámetro de varianza de la incertidumbre de la medición del ángulo de elevación. El ángulo de elevación y azimut de la antena del Radar Vitro es de 1° al igual que el ángulo de azimut del Bendix, por lo que σ_θ^2 y σ_φ^2 se mantienen con el mismo valor utilizado. El Radar Vitro utiliza un ancho de pulso de $0,5 \mu s$; menor al del Bendix ($4 \mu s$). De esta manera, disminuye la incertidumbre en la medición del rango ($\sigma_\rho = 50$).

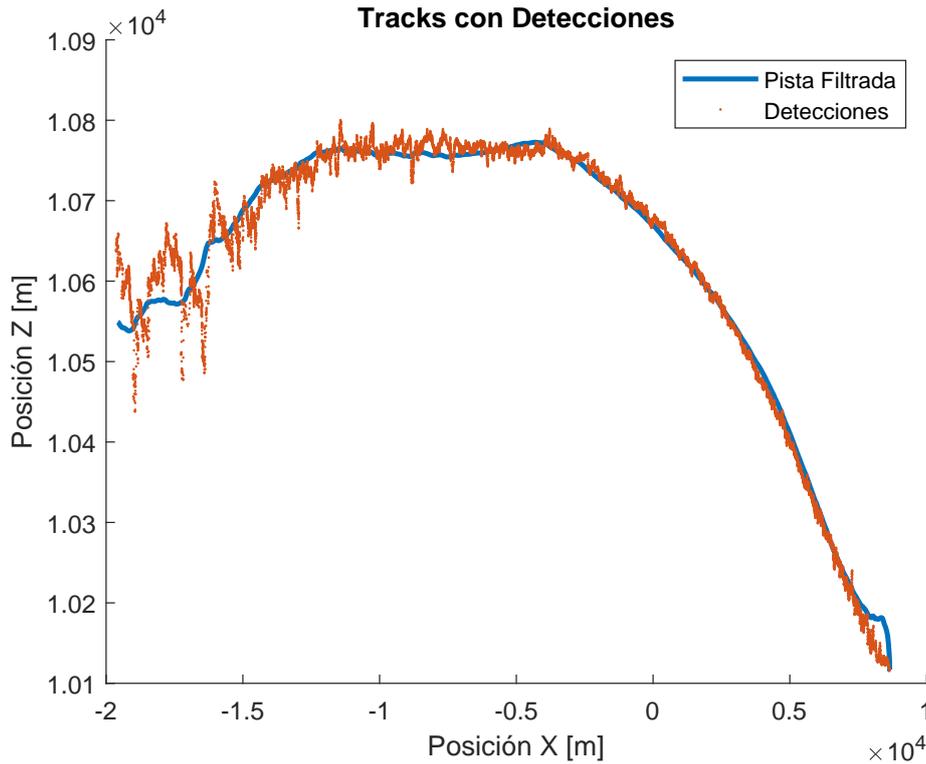


FIGURA 4.15: Trayectoria del Boeing 737 en los planos X-Z. El sentido de la trayectoria es de derecha a izquierda.

La única matriz que resta extender para el seguimiento en 3D es el Jacobiano utilizado para la conversión de coordenadas esféricas a cartesianas del ruido de medición.

$$J(\rho, \theta, \varphi) = \begin{bmatrix} \cos \varphi \cos \theta & -\rho \cos \varphi \sin \theta & -\rho \sin \varphi \cos \theta \\ \cos \varphi \sin \theta & \rho \cos \varphi \cos \theta & -\rho \sin \varphi \sin \theta \\ \sin \varphi & 0 & \rho \cos \varphi \end{bmatrix} \quad (4.9)$$

Con esta matriz se calcula el error de medición en coordenadas cartesianas, dependiendo la posición de la detección. Este error se usa para corregir la estimación producida por el filtro de kalman.

Se puede graficar ahora la posición del objetivo en los planos X-Z, como se observa en la Figura 4.15. El objetivo se mueve de derecha a izquierda. Notar que el eje de la posición en Z se encuentra ampliado para apreciar la diferencia entre las detecciones y el filtrado.

En la Figura 4.16 se muestra la trayectoria reconstruida en 3 dimensiones. Se incluye en la coordenada [0, 0, 0] la posición del radar.

Con los datos en 3 dimensiones de la trayectoria filtrados, se puede representar el vuelo capturado como se observa en la Figura 4.17. Se toma la trayectoria filtrada y se la proyecta sobre una imagen con el terreno extraída de Google Earth. Se muestra en un marcador la posición del Radar Vitro. El vuelo va en dirección Oeste, hacia Santiago de Chile. Para este caso particular, se tuvo señal radar hasta los 85 Km aproximadamente, pasando las Altas Cumbres.

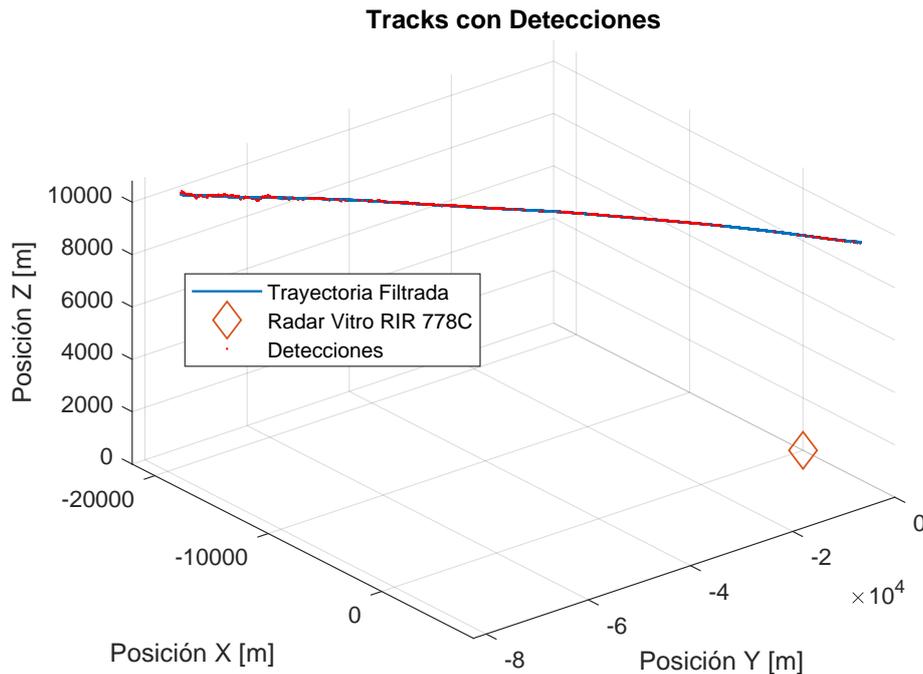


FIGURA 4.16: Trayectoria del Boeing 737 reconstruida en 3 dimensiones.

4.1.5. Trayectoria real con maniobra en 3D

Se mostrará ahora el caso para un objetivo real con maniobra. El seguimiento fue realizado con el Radar Vitro el 14/11/2018. Corresponde a un Grob G-120 TP de la Escuela de Aviación Militar, en Córdoba. En la Figura 4.18 se observan las detecciones producidas por el radar en el plano X-Y.

Para objetivos con alta capacidad de maniobra como el de este caso, el seguimiento en coordenadas cartesianas genera mucho error, lo que hace muy difícil ajustar el filtro. En las Figuras 4.19 y 4.20 se muestran zonas de la trayectoria estimada con gran diferencia entre las detecciones y las estimaciones.

También influye en el origen de este error, que esta trayectoria ha sido tomada por el radar con una tasa de datos de 100 ms, a diferencia de los 20 ms del caso anterior. Para un objetivo con alta capacidad de maniobra es crucial la alta tasa de datos, para que se pueda linealizar el movimiento entre punto y punto.

Adaptando el algoritmo de filtrado para hacer el seguimiento en coordenadas esféricas se logra un mejor comportamiento. Se muestra a continuación la modificación en las matrices para el seguimiento en coordenadas esféricas.

El vector de estado se define en la Eq. 4.10,

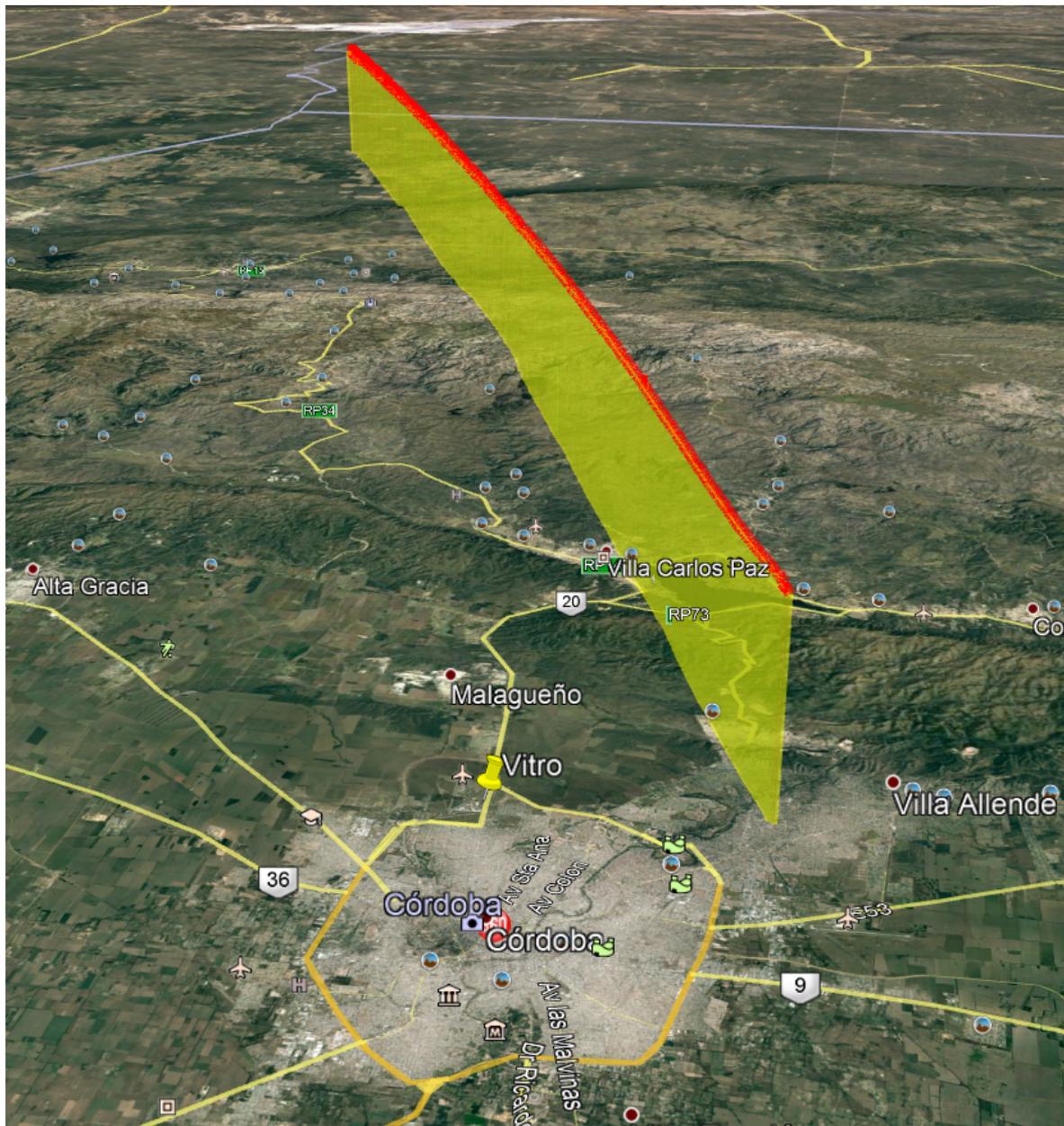


FIGURA 4.17: Trayectoria del Boeing 737 reconstruida en 3 dimensiones sobre el terreno.

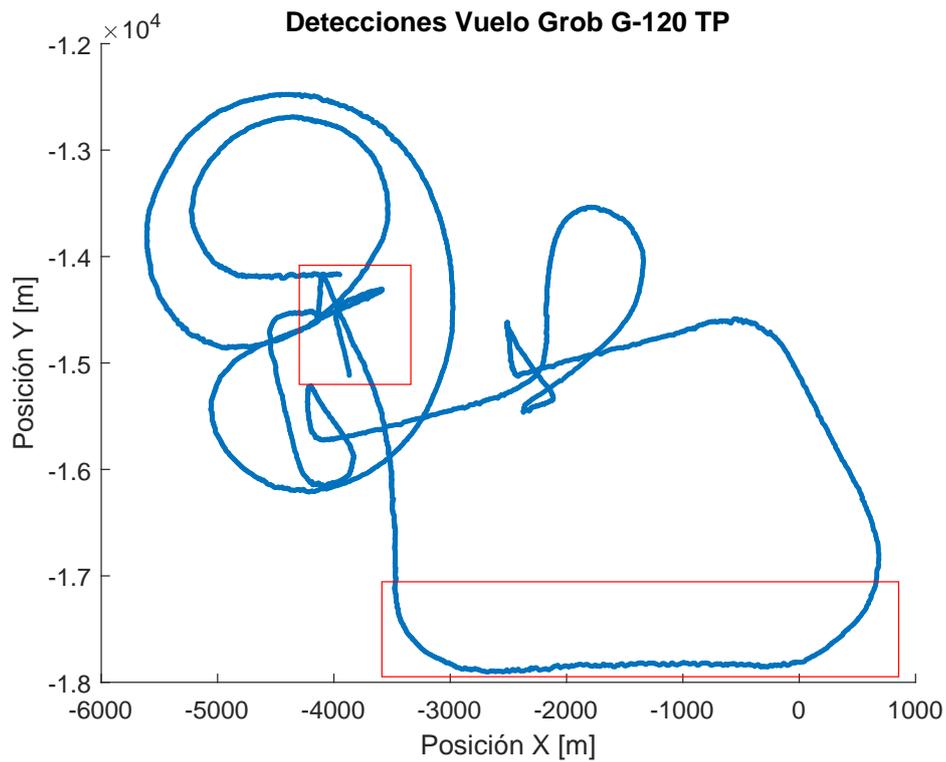


FIGURA 4.18: Detecciones del vuelo del Grob proyectadas sobre el plano X-Y. Se marca con recuadros las zonas ampliadas en la Figura 4.19 y 4.20.

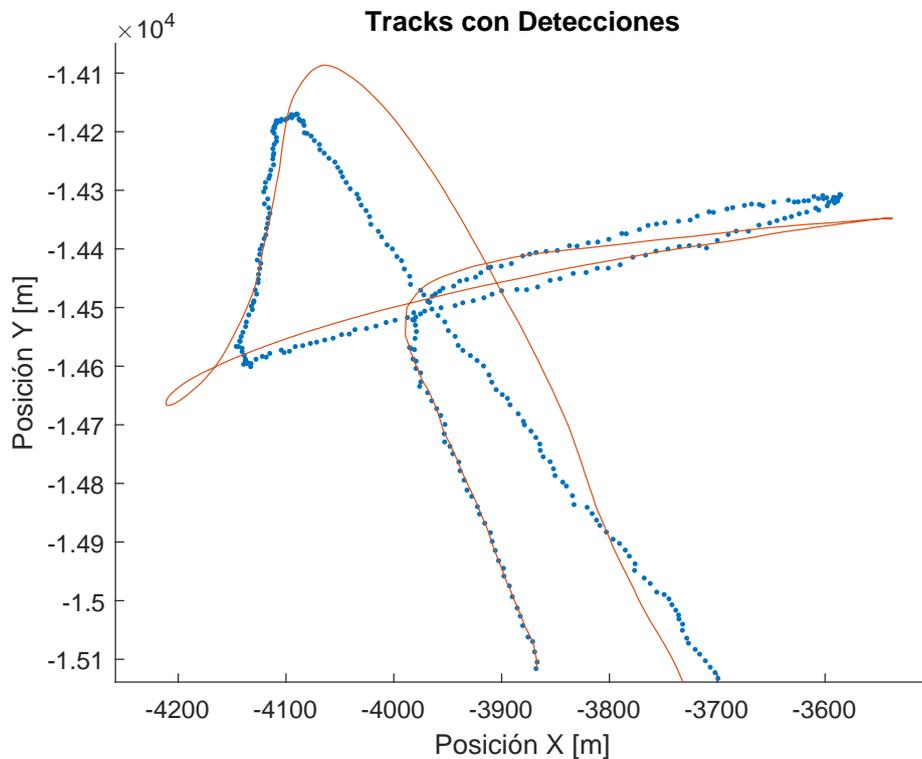


FIGURA 4.19: Diferencia de estimación en coordenadas cartesianas y detecciones del vuelo del Grob G-120 TP en su tramo inicial.

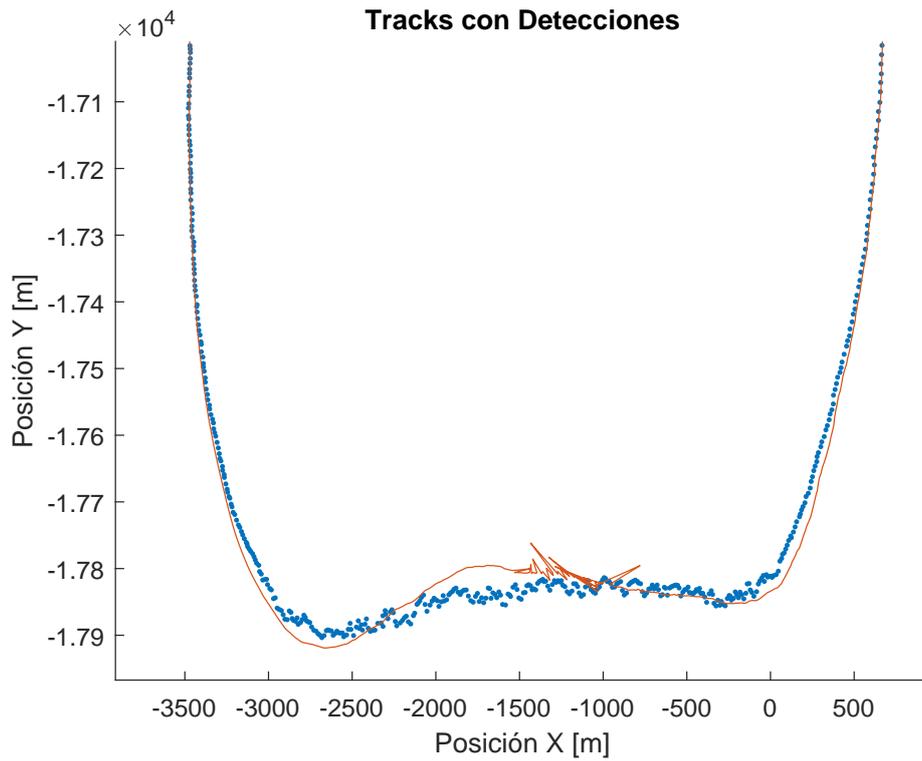


FIGURA 4.20: Diferencia de estimación en coordenadas cartesianas y detecciones del vuelo del Grob G-120 TP en tramo intermedio.

$$x_{esferica} = \begin{bmatrix} \rho \\ \dot{\rho} \\ \ddot{\rho} \\ \theta \\ \dot{\theta} \\ \ddot{\theta} \\ \varphi \\ \dot{\varphi} \\ \ddot{\varphi} \end{bmatrix}_k \quad (4.10)$$

Donde θ es el ángulo de azimut y φ es el ángulo de elevación.

El seguimiento se hará en los 3 ejes de forma independiente, siguiendo las ecuaciones listadas en 4.11,

$$\left\{ \begin{array}{l} \rho_{k+1} = \rho_k + \dot{\rho}_k T + \frac{1}{2} \ddot{\rho}_k T^2 \\ \dot{\rho}_{k+1} = \dot{\rho}_k + \ddot{\rho}_k T \\ \ddot{\rho}_{k+1} = \ddot{\rho}_k + n_{a\rho} \\ \theta_{k+1} = \theta_k + \dot{\theta}_k T + \frac{1}{2} \ddot{\theta}_k T^2 \\ \dot{\theta}_{k+1} = \dot{\theta}_k + \ddot{\theta}_k T \\ \ddot{\theta}_{k+1} = \ddot{\theta}_k + n_{a\theta} \\ \varphi_{k+1} = \varphi_k + \dot{\varphi}_k T + \frac{1}{2} \ddot{\varphi}_k T^2 \\ \dot{\varphi}_{k+1} = \dot{\varphi}_k + \ddot{\varphi}_k T \\ \ddot{\varphi}_{k+1} = \ddot{\varphi}_k + n_{a\varphi} \end{array} \right. \quad (4.11)$$

Debido a esto, la matriz de transición de estado (F), no sufrirá modificaciones.

La matriz de covarianzas del ruido de medición (R), seguirá teniendo la misma forma, pero ya no hará falta el Jacobiano para hacer la conversión a coordenadas cartesianas de ese error.

Por ultimo, la matriz de covarianzas del ruido de proceso (Q) en sus únicos elementos no nulos ahora indicará varianza en la incertidumbre de la aceleración en rango (ρ), azimut (θ) y elevación (φ).

$$Q_{esferica} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{a\rho}^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{a\theta}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{a\varphi}^2 \end{bmatrix} \quad (4.12)$$

Con estas modificaciones se realiza el seguimiento y se muestra el resultado en la Figura 4.21. Se observa un mejor comportamiento de las estimaciones sin perturbaciones como las observadas en la Figura 4.20, y una rápida convergencia a pesar de las abruptas maniobras efectuadas sobre el inicio de la trayectoria.

En la Figura 4.22 se muestran las componentes independientes en coordenadas esféricas de la trayectoria.

Para este caso, la facilidad del filtrado en coordenadas esféricas, radica en que es más fácil describir la trayectoria como 3 componentes independientes, sin que el ruido de medición se combine en forma de covarianzas en la matriz $R_{cartesiana}$. Sin embargo, la perturbación al sistema ingresada a través de la matriz Q, pasa a ser dependiente de la posición del objetivo con respecto al radar, lo que puede no ser conveniente en algunos casos.

Al igual que en el caso anterior, se muestra la trayectoria reconstruida en 3 dimensiones en la Figura 4.23 y sobre el terreno en la Figura 4.24.

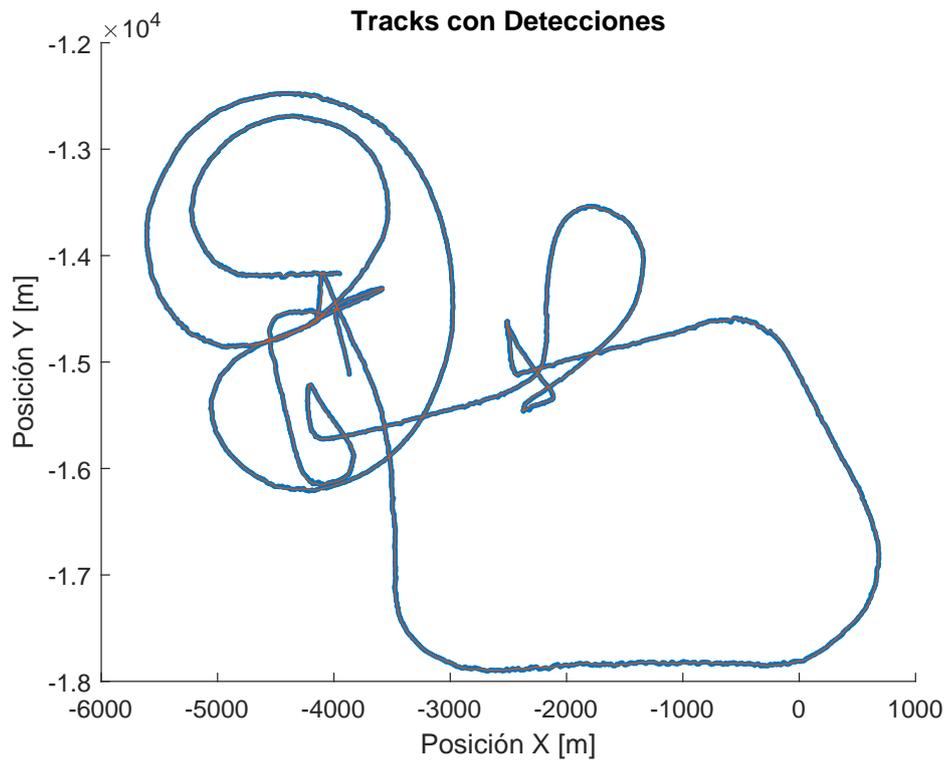


FIGURA 4.21: Pista filtrada en Coordenadas Esféricas del vuelo del del Grob G-120 TP.

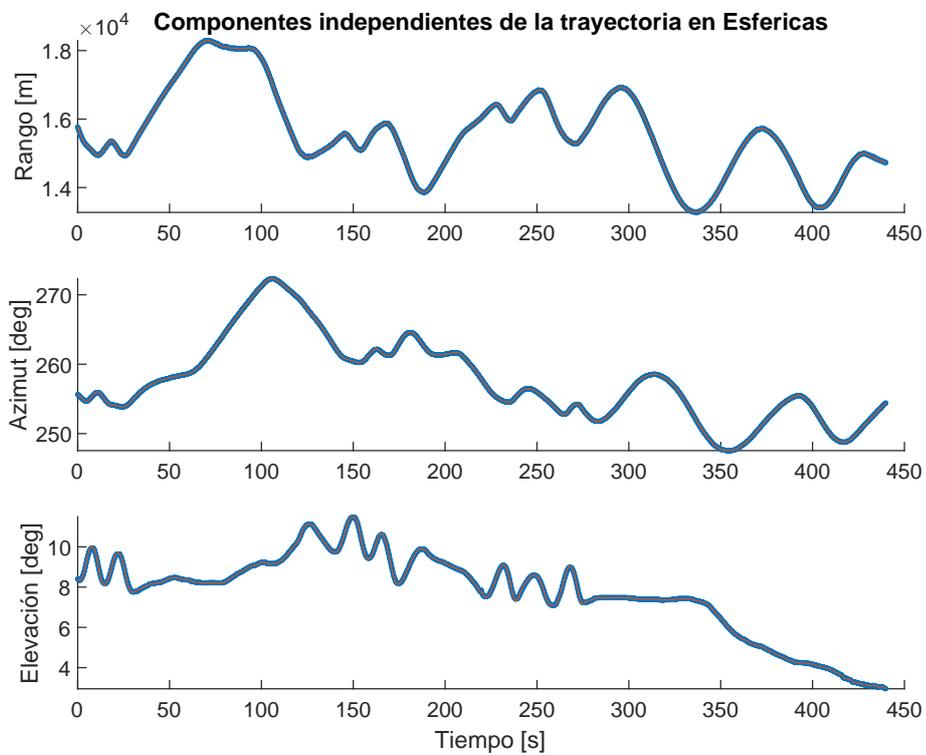


FIGURA 4.22: Componentes independientes de la trayectoria del Grob en coordenadas Esféricas.

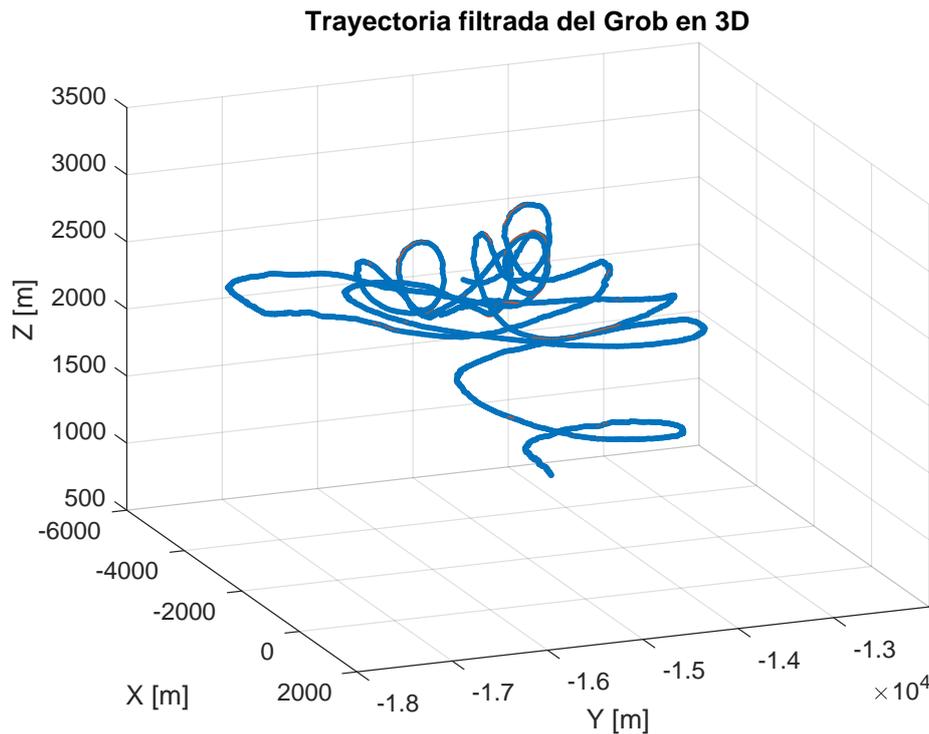


FIGURA 4.23: Reconstrucción del vuelo del Grob en 3 dimensiones.

4.2. Asociación de Datos

Se mostrarán a continuación resultados del seguimiento de múltiples objetivos, en los que está involucrado el filtrado de las trayectorias y la asociación de datos para la determinación de correspondencia entre detecciones y pistas. También se incluyen en esta sección resultados con *clutter* y pérdida de detecciones.

4.2.1. Múltiples trayectorias simuladas en 2D

Como primer caso de asociación, se utiliza el simulador para generar 5 trayectorias aleatorias. Estas trayectorias tienen asociado ruido de detección según los parámetros del radar Bendix. Las trayectorias tienen puntos en los que se cruzan unas con otras, para demostrar el funcionamiento del algoritmo de asociación. Se muestran en la Figura 4.25 las trayectorias generadas junto con las detecciones y las pistas estimadas.

Los pasos que se siguen son los siguientes,

1. Se genera un número n de trayectorias en 2D con la resolución de tiempo (dT), tiempo final (t_{Final}), posición inicial ($pos_{Inicial}$), velocidad inicial ($v_{Inicial}$) y aceleración ($varA$) configurable.
2. Se le agrega ruido en coordenadas polares ($varNoise$) para emular incertidumbre de medición.
3. Se hace un submuestreo de las trayectorias, emulando una detección de radar por vuelta ($tiempoVuelta$; 12 segundos para el radar Bendix).
4. Se eliminan detecciones de forma aleatoria con un parámetro configurable ($missRate$; no agregado para este ejemplo).

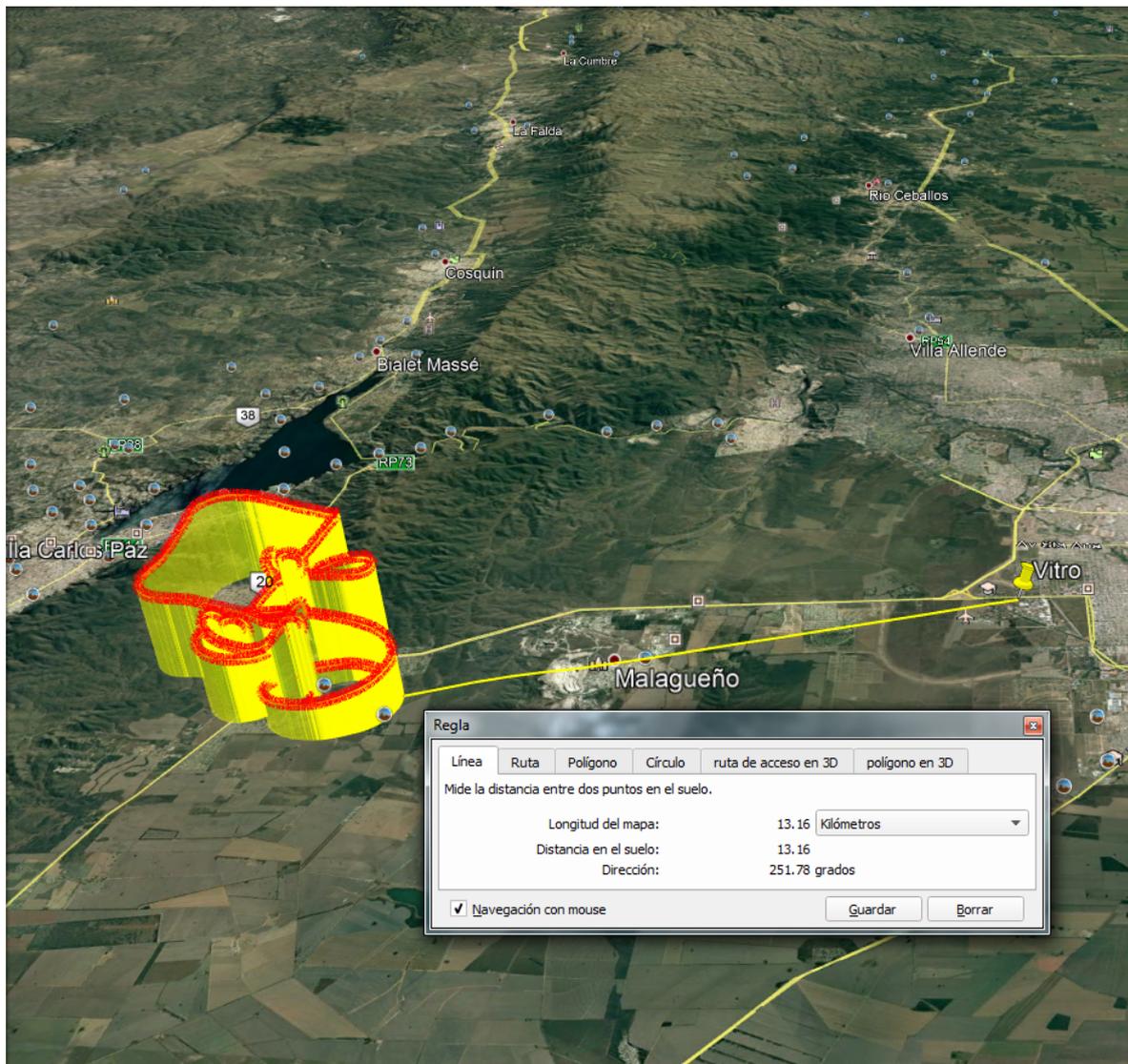


FIGURA 4.24: Reconstrucción del vuelo del Grob en 3 dimensiones sobre el terreno.

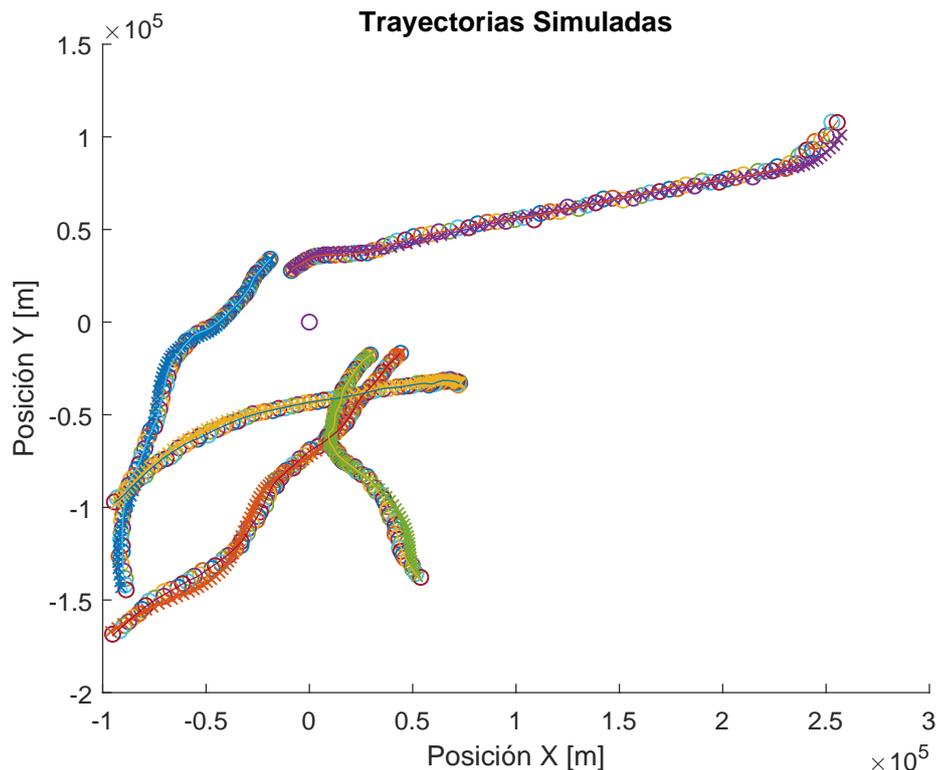


FIGURA 4.25: Trayectorias Simuladas junto con las pistas estimadas incluyendo el proceso de asociación de datos.

5. Se agregan detecciones de forma aleatoria para emular *clutter*, mediante el parámetro (`clutterRate`).
6. Luego comienza el ciclo de asociación y filtrado, en el que se hacen tantas repeticiones como vueltas de radar implique el tiempo de simulación. Por cada vuelta de radar, se toman los datos de detecciones correspondientes a esa vuelta y se ingresan al ciclo descrito en la Figura 3.7.

4.2.2. Trayectoria simulada en 2D con pérdida de datos

Se muestra en la Figura 4.26 una trayectoria simulada en la que se ha establecido el parámetro `missRate = 0.2`. Esto implica un 20% de detecciones aleatorias perdidas. Se observa cómo el algoritmo de kalman (a través de la función `kalmanPredict` llena los huecos generados por las detecciones faltantes y completa la trayectoria.

4.2.3. Trayectoria simulada en 2D con clutter

En este caso, se simula una trayectoria y se agrega un 8% de detecciones extra en posiciones aleatorias, mediante el parámetro `clutterRate = 0.08`. En la Figura 4.27 se muestra la trayectoria simulada con su pista asociada y las detecciones que han sido descartadas sin generar pistas.

En situaciones como esta cobra relevancia la función `detectionToTrackAssignment` que asigna a la pista la detección correcta a partir de la mínima distancia a la posición predicha.

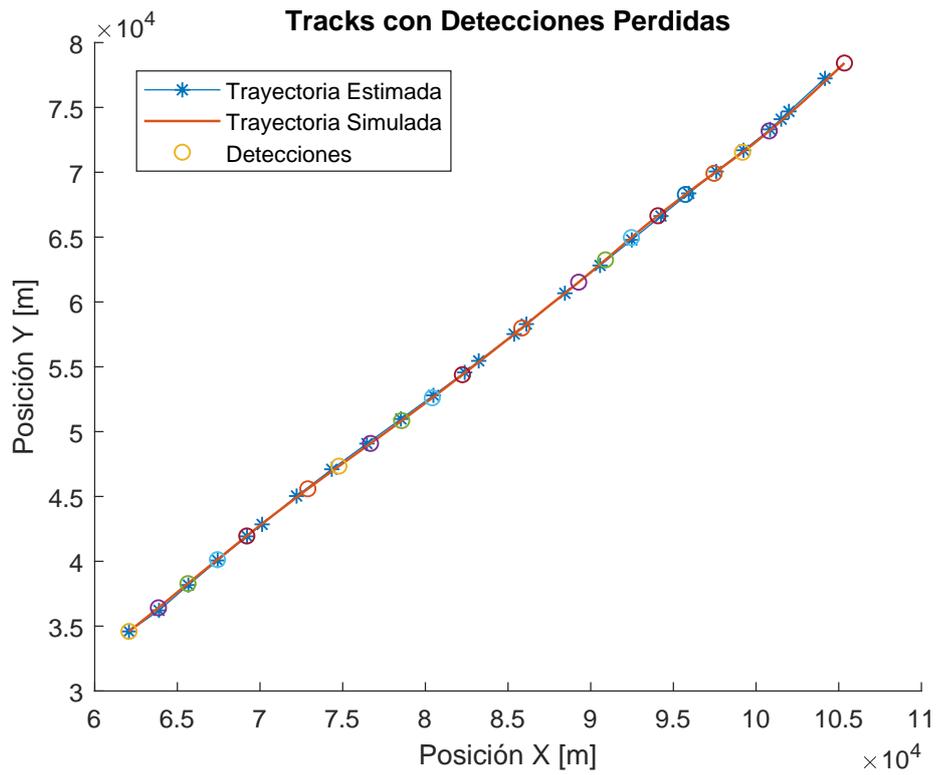


FIGURA 4.26: Trayectoria Simulada con 20% de detecciones aleatorias perdidas ($missRate = 0.2$), junto con las predicciones de las posiciones.

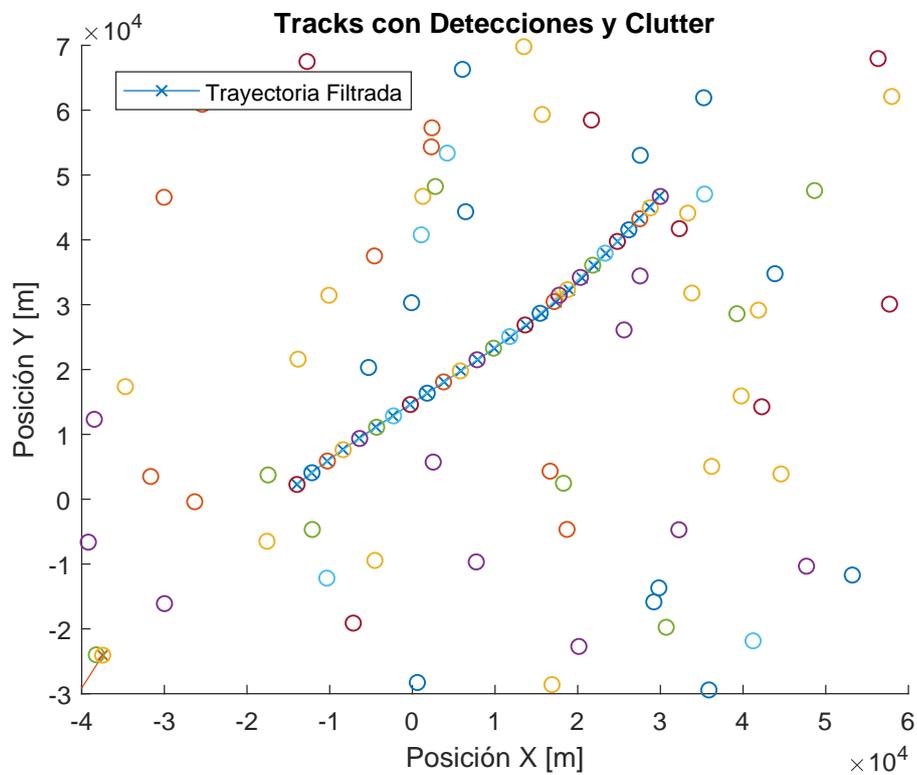


FIGURA 4.27: Trayectoria Simulada con 8% de *clutter* agregado en cada vuelta ($clutterRate = 0.08$), junto con la pista estimada.

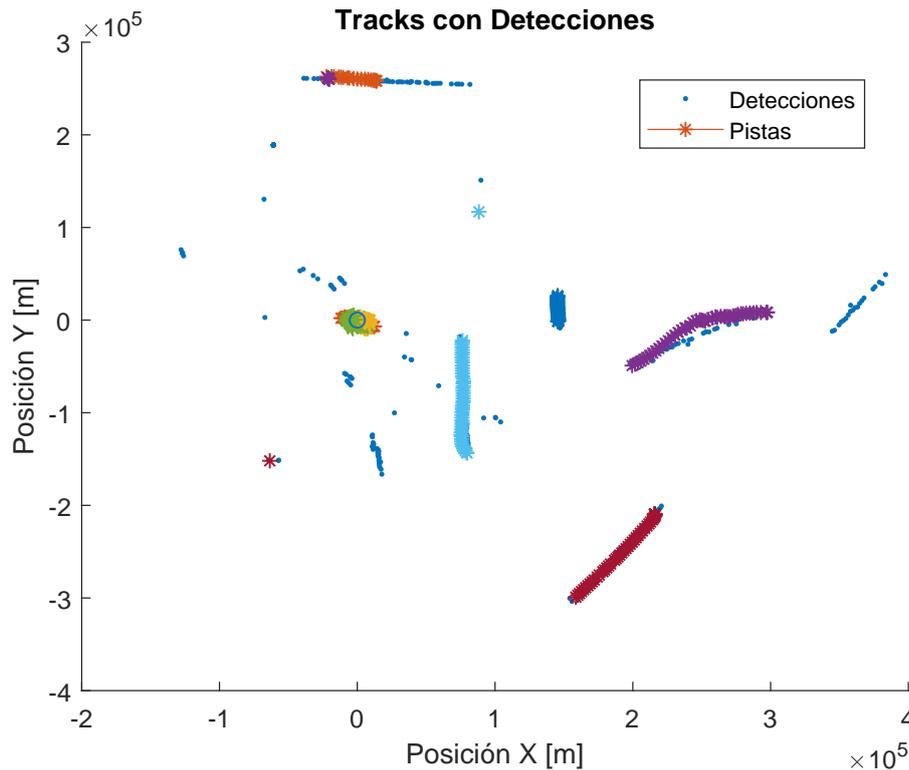


FIGURA 4.28: Trayectorias Reales capturadas por radar junto con las pistas generadas.

Mediante el parámetro `costOfNonAssignment` se descarta generar pistas a partir de *clutter*. Para este caso, esta distancia de no asignación se ha establecido en 1000 metros.

Todas las detecciones correspondientes a *clutter* generarán pistas tentativas, a las cuales se les hará predicción de posición, cálculo de distancia a otras detecciones y asociación a detecciones cercanas. Es importante en este caso la función `deleteLostTracks` para eliminar aquellas pistas que se interpreta que corresponden a *clutter*. En caso de no eliminarlas, el cómputo necesario crece de forma exponencial a la cantidad de detecciones. Para este ejemplo, los parámetros `invisibleForTooLong` es igual a 3. Esto implica que si la pista no recibe detecciones por más de 3 vueltas es eliminada. El parámetro `ageThreshold` se ha establecido en 8. Esto implica que si la pista tiene un tiempo de vida menor a 8, necesita tener un porcentaje de detecciones mayor al establecido en `visibility`, 60% para este ejemplo.

Por último, la pista se hace visible al usuario cuando su tiempo de vida es mayor al especificado en el parámetro `ageToRealTrack`. En este caso se ha establecido en 3 vueltas.

4.2.4. Trayectorias reales en 2D

A continuación se muestran algunas capturas de la salida del algoritmo funcionando en operación, en tiempo real, en modo TWS.

En la Figura 4.28 se muestra la situación de un radar con relativamente poco *clutter* (se observa una mayor concentración de detecciones y pistas generadas en torno al punto $[0, 0]$). Las detecciones que se observan sin pistas asociadas se deben a que son detecciones de vueltas pasadas que han quedado persistentes en pantalla, pero el algoritmo ha eliminado por no seguir siendo pistas válidas.

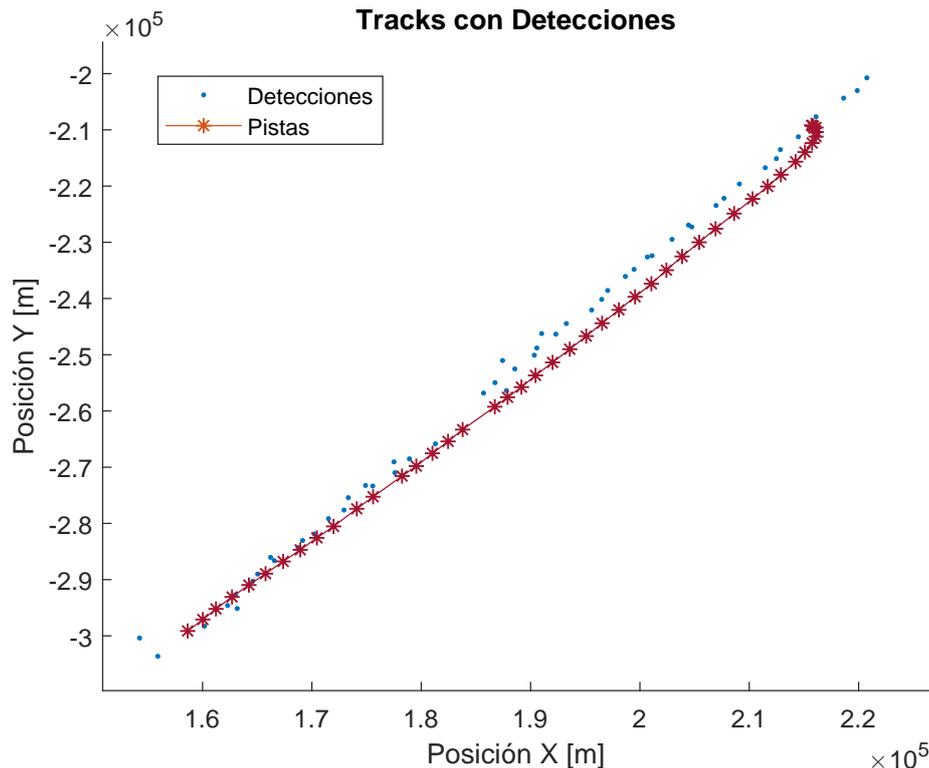


FIGURA 4.29: Trayectoria Real seguida con Radar TWS.

En la Figura 4.29 se muestra una de las trayectorias de la Figura 4.28 en la que se observa la predicción de la pista en aquellos casos en que no hay detección, y el suavizado de la misma pese a lo ruidoso de las detecciones.

En la Figura 4.30 se observa otra situación real, en un escenario de 8 minutos de datos, con gran cantidad de *clutter*. La región central del mapa (cercano al radar) ha de ser descartada, y su tratamiento en detalle escapa al alcance de este trabajo. Por fuera de esta zona se observa un buen comportamiento de las predicciones.

4.3. Performance Computacional

Se hará a continuación una breve reseña sobre el desempeño del programa en el hardware del radar. Se ha realizado un *profiling* sobre la SBC que ejecuta el software *tracker* del radar ingresando datos correspondientes a 1 hora de funcionamiento del radar. El escenario sobre el que se han capturado los datos es el de alto *clutter* de la Figura 4.30.

4.3.1. Uso de Memoria

Mediante la herramienta *Massif* del kit de *Valgrind*, se hace un análisis del uso de memoria durante este tiempo.

En la Figura 4.31 se observa en la curva superior el uso total de memoria. El uso de memoria ronda los 300 KB de RAM y es estable a lo largo del tiempo de ejecución.

Las dos estructuras de datos de relevancia en el programa son la lista que contiene las estructuras de las pistas, con las matrices del kalman y el estado de la pista; y la lista que contiene las detecciones a medida que van ingresando al *tracker*. En las Figuras 4.31, 4.32, 4.33 y 4.34 se muestra la participación de estos datos dentro de los datos totales consumidos.

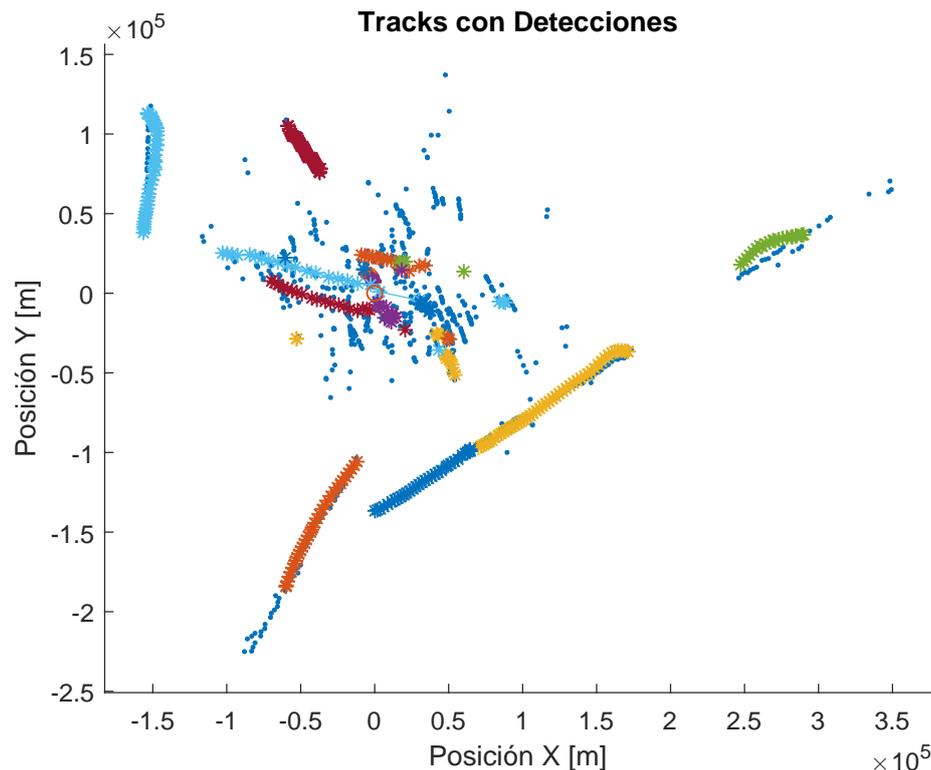


FIGURA 4.30: Trayectorias Reales seguidas con Radar TWS en situación de alto clutter.

A partir de estos gráficos se determina que hay un buen uso de la memoria, principalmente debido a la política de borrado de pistas una vez que no son válidas.

4.3.2. Uso de CPU

Haciendo funcionar el programa en tiempo real, mediante la herramienta *Callgrind* de *Valgrind*, se obtiene el siguiente porcentaje de tiempo de ejecución de las siguientes funciones, ya desarrolladas en el trabajo,

- trackTop: 25.2 %
 - updateUnassignedTracks: 44.01 % (11.09 % sobre el total)
 - kalmanCorrect: 84.58 % (9.38 % sobre el total)
 - predictNewLocationOfTracks: 32.42 % (8.17 % sobre el total)
 - kalmanPredict: 99.14 % (8.1 % sobre el total)
 - createNewTracks: 17.74 % (4.47 % sobre el total)
 - deleteLostTracks: 5.83 % (1.47 % sobre el total)
- trackPlot: 13.72 %
 - detectionToTrackAssignment: 99.05 % (13.59 % sobre el total)
 - updateAssignedTracks: 0.87 % (0.12 % sobre el total)

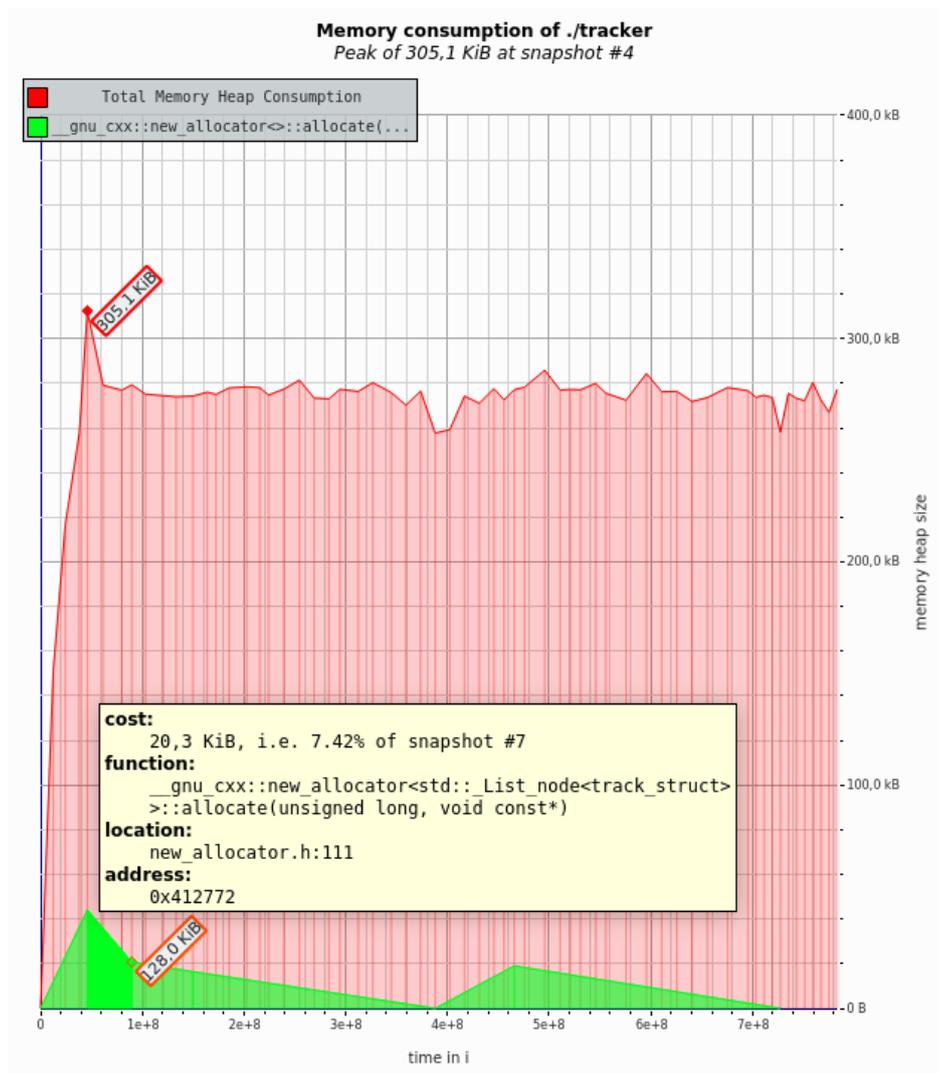


FIGURA 4.31: Consumo de Memoria de la lista que contiene las pistas.

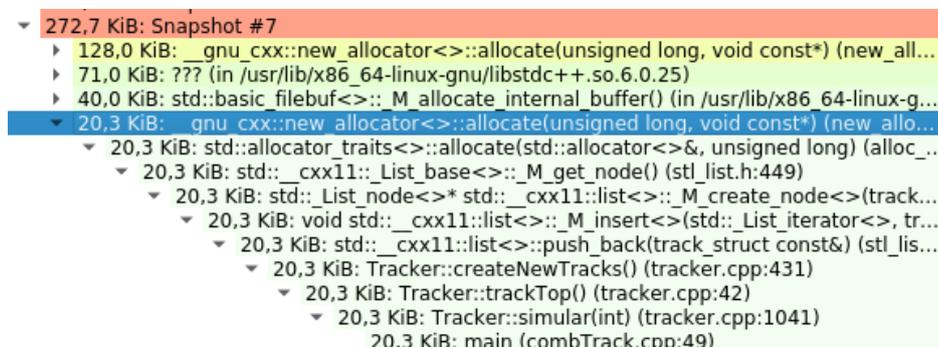


FIGURA 4.32: Llamada de funciones que incrementan la estructura de datos de las pistas.

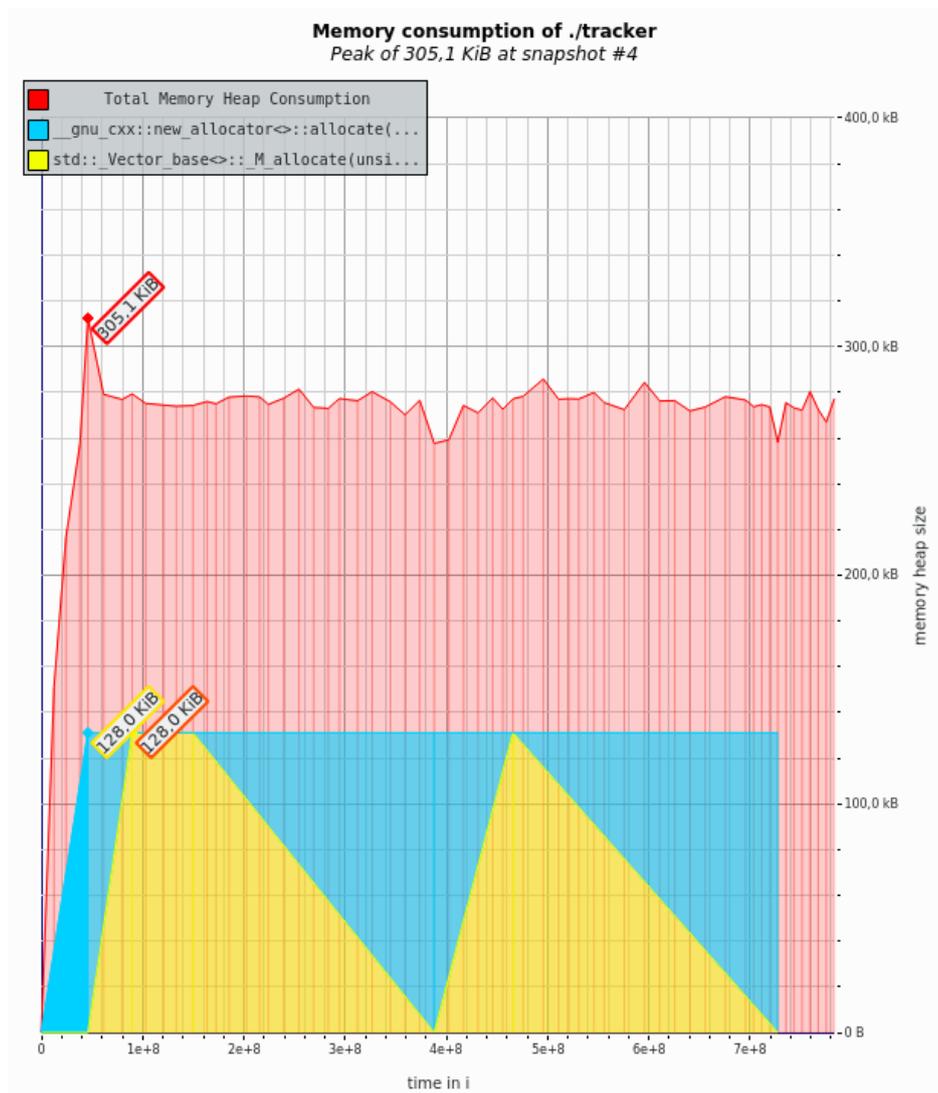


FIGURA 4.33: Consumo de Memoria de las listas que contienen las detecciones.

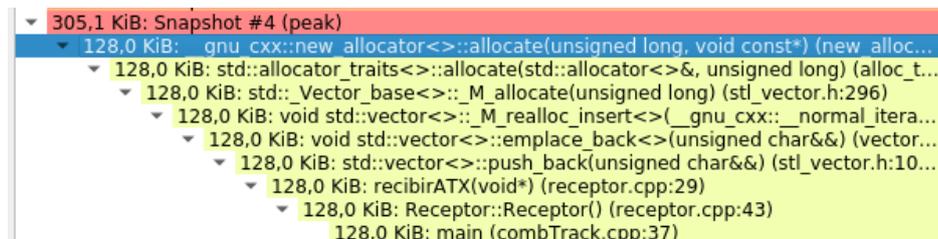


FIGURA 4.34: Llamada de funciones que incrementan las estructuras de datos de detecciones.

Entre las dos funciones de entrada al *tracker*, *trackTop* y *trackPlot*, se consume un 38.92% del tiempo de CPU. El resto del tiempo la computadora está en estado *idle*, a la espera de detecciones.

Los otros porcentajes mostrados, indican el porcentaje de tiempo utilizado de cada función en relación a la función que los llama, y entre paréntesis en relación al tiempo total. Se observa a partir de esto, que las funciones que más peso tienen sobre el sistema son *detectionToTrackAssignment*, que implica el cálculo de distancias y el algoritmo de Munkres, y las funciones de predicción y corrección de kalman, que implican las multiplicaciones de matrices.

En términos relativos, el tiempo de uso de CPU es relativamente bajo, lo que posibilita un buen comportamiento en tiempo real, incluso en ambientes de gran cantidad de *clutter*.

Capítulo 5

Conclusiones

En este trabajo se ha desarrollado un Sistema de Seguimiento y Asociación de Datos para radares de la Fuerza Aérea Argentina, que no contaban con esta capacidad.

Se comenzó en el Capítulo 1 por una reseña histórica sobre los radares involucrados en el trabajo. Posteriormente se abordó el problema a resolver, con los desafíos que presentan los Sistemas de Seguimiento a partir de datos no certeros. Se habló sobre Sistemas de Vigilancia, en qué consiste el *tracking* basado en Filtros de Kalman. Se presentó el problema de la asociación de datos, mostrando los algoritmos necesarios para abordar cada uno de estos pasos. En el Capítulo 3, se presentó el sistema sobre el cual este trabajo iba a tener que funcionar, el EDDR. Se profundizó posteriormente en cómo se implementan los algoritmos explicados en el Marco Teórico y como se realiza su implementación final en el hardware del radar. Se mostraron las herramientas (simuladores) utilizadas para validar el sistema.

En el Capítulo 4, Resultados, se mostraron casos reales y simulados del comportamiento del sistema. Se simulan trayectorias simples en 2D para validar las ecuaciones de los filtros. Posteriormente se muestra el comportamiento del sistema con datos del Radar Vitro RIR 778C, un radar de seguimiento 3D. Para esto, se extendió el filtro y las ecuaciones de seguimiento a 3 dimensiones. Se mostró la conveniencia de un seguimiento en coordenadas esféricas para ciertos casos, como el de una trayectoria compleja con gran capacidad de maniobra en 3 dimensiones. Se utilizó el simulador para generar varias trayectorias simultáneas, con cruces entre ellas, con pérdida de datos y *clutter* (intentando emular un escenario lo más real posible), con el cual validar los algoritmos de asociación.

Se mostraron finalmente capturas del funcionamiento del trabajo en radares de operación de la Fuerza Aérea Argentina.

Se hizo un estudio de *performance* del sistema mediante *profiling*, mostrando que, pese a la computadora de reducida capacidad sobre la cual el *tracker* funciona, el consumo de CPU y Memoria se mantiene dentro de parámetros aceptables.

Cabe destacar el gran esfuerzo que ha implicado acceder a datos reales de radar, con gran tiempo de pre-procesamiento, filtrado e interpretación de datos y señales hasta poderlas tener en formato de un vector de entrada al programa de seguimiento.

Resulta de gran interés que este trabajo sea continuado, existiendo grandes posibilidades de mejora y evolución, sobre todo teniendo acceso a datos reales. Entre las mejoras propuestas se incluye el desarrollo de un simulador por hardware que permita trayectorias más complejas, o que permita la generación de *clutter*. Entre los desarrollos a continuar sobre el sistema de seguimiento, se propone extender el Filtro de Kalman para lograr un mejor desempeño en coordenadas cartesianas para trayectorias complejas. Se propone también describir la trayectoria del móvil a seguir en coordenadas naturales, para poder discriminar un parámetro que indique varianza para la capacidad de maniobra permitida. También es posible extender el sistema para realizar una asociación más inteligente, mediante filtros del tipo JPDAF (*Joint Probabilistic Data Association Filter*).

Referencias

- Aeroplano (2011). «100 años de Aviación Militar Española». En: *Aeroplano* 29.
- Bar-Shalom, Yaakov y Thomas E. Fortmann (1988). «Tracking and Data Association». En: *Academic Press. Mathematics in Science and Engineering* 179.
- Bar-Shalom, Yaakov, Peter K. Willett y Xin Tian (2011). «Tracking and Data Fusion, A Handbook of Algorithms». En: *YBS Publishing*.
- Bouwman, R. (2009). «Fundamentals of Ground Radar For Air Traffic Control Engineers and Technicians». En: *Scitech Publishing, Inc.*
- Caranti, G. et al. (oct. de 2015). «GRC en la Radarización Meteorológica de Argentina». En: *XVI Reunión de Trabajo en Procesamiento de la Información y Control (RPIC-2015)*.
- Defense, United States Department of (2013). «MIL-STD-196F - Joint Electronics Type Designation System». En:
- González, Comodoro (R) VGM O. (oct. de 2014). «Vigilancia y Control Aéreo en Argentina - La Historia de la Especialidad VYCA en la Fuerza Aérea». En: *Ediciones Argentinidad*.
- Kalman, R. E. (1960). «A New Approach to Linear Filtering and Prediction Problems». En: *IEEE Transactions on Signal Processing*, págs. 35-45.
- Kuhn, Harold W. (1955). «The Hungarian Method for the Assignment Problem». En: *Naval Research Logistics Quarterly*, págs. 83-97.
- Montamat, I. (mayo de 2015). «Combinador de Información Primaria y Secundaria para Extractor Digital de Datos de Radar en Sistemas de Vigilancia». En: *Universidad Nacional de Córdoba*. URL: <http://www.famaf.unc.edu.ar/wp-content/uploads/2015/10/2-Montamat.pdf>.
- Munkres, James (1957). «Algorithms for Assignment and Transportation Problems». En: *Journal of the Society for Industrial and Applied Mathematics* 5.1, págs. 83-97.
- Paluszek, Michael y Stephanie Thomas (2017). «Matlab Machine Learning». En: *Academic Press*.
- Richards, Mark A. (2010). «Principles of Modern Radar». En: *SciTech Publishing*.
- (2014). «Fundamentals of Radar Signal Processing». En: *McGraw-Hill Professional Engineering*.
- Rodríguez, A. et al. (2017). «SiNaRaMe: El Primer Sistema Integrado de Radares Hidro-Meteorológicos de Latinoamérica». En: *Revista Facultad de Ciencias Exactas, Físicas Y Naturales* 4.1, págs. 41-48.
- Rodríguez, S. A. y M. Anún (abr. de 2011). «Radar VITRO RIR-778C». En: *Noticias de Nuestra Universidad* IX.I.
- Rodríguez, S. A., M. Anún y A. Menara (feb. de 2013). «Analysis, Simulation and Testing of RF Reception System of a RIR Type Radar». En: *IEEE Latin America Transactions* XI.1, págs. 65-70.
- Ross, Sheldon M. (2014). «Introduction to Probability Models». En: *Academic Press*, pág. 187.
- Safety of Air Navigation, European Organisation for the (2012). «Eurocontrol Standard Document for Surveillance Data Exchange, Part 4: Category 48. Transmission of Monoradar Target Reports. SUR.ET1.ST05.2000-STD-04-01». En: *European Air Traffic Management*.
- Stevens, M. (1988). «Secondary Surveillance Radar». En: *Artech House, Inc.*
- Winkler, D. (jul. de 1997). «Searching the Skies - The Legacy of the United States Cold War Defense Radar Program». En: *United States Air Force Air Combat Command*.