



Universidad Carlos III de Madrid  
Escuela Politécnica Superior

# Técnicas avanzadas de estimación y filtrado: análisis y evaluación en entornos marítimos

Trabajo de Fin de Máster  
Máster Universitario en Ciencia y Tecnología Informática  
2018 - 2019

Por  
**Daniel Amigo Herrero**

Supervisado por  
**Jesús García Herrero**

8 de julio de 2019

## Resumen

La presente tesis está enmarcada en el área de seguimiento y estimación de vehículos. Esta ha ido adquiriendo relevancia a lo largo de los años y ahora con los vehículos autónomos aún más.

Las técnicas de fusión de datos permiten realizar una combinación de información recogida por varias fuentes sobre un mismo objetivo, combinándola y obteniendo de esta manera información de mayor calidad. Estas técnicas aplicadas al área de seguimiento de vehículos tiene como principal objetivo la detección de un entorno cercano. Los sensores instalados en los vehículos tienen un ruido implícito que hace que su información no sea completamente precisa. Gracias a la fusión de datos, obtener una estimación más precisa del entorno hace que la toma de decisiones, ya sea de un vehículo autónomo o de un operario sea más informada y potencialmente más correcta.

Los sistemas de fusión de datos son estructuras muy complejas resultado de la intercomunicación de varios sub-sistemas. La estimación de posición es parte de uno de estos sub-sistemas.

En esta tesis, tras realizar una gran comprensión del estado del arte de esta rama, se diseña y desarrolla una herramienta que permite la comparación de varias técnicas de estimación de posición. Se implementa las cinco de las técnicas más famosas, así como cálculo de métricas para evaluar su rendimiento y gráficas que le otorguen la capacidad de análisis en un vistazo.

El sistema implementado habilita un análisis realizado con varios escenarios escogidos de entorno marítimo. Estos tienen unos resultados positivos, tanto de rendimiento como de consistencia de la estimación. La conclusión de dicho análisis indica que el filtro IMM otorga los mejores resultados.

## **Abstract**

This thesis is involved in the area of research of vehicle tracking and estimation of position. This area has become significant over the years, especially now with autonomous vehicles.

Data fusion techniques allow a combination of information collected by several sources on the same objective by merging it and obtaining higher quality information. These techniques applied to the vehicle tracking area have as their main objective the detection of nearby targets. The sensors used in the vehicles have an implicit white-noise that makes the generated information not completely accurate. Data fusion allows to obtain a more accurate estimate of the nearby target helping the decision-making, because any decision will be a more informed choice, whether is carried out by a machine or a human.

Data fusion systems are very complex structures resulting from the intercommunication of several sub-systems. Position estimation is part of one of these sub-systems.

In this thesis, after making a complete of the state of the art of this area of research, a framework is designed and developed to make comparisons among several techniques of position estimation. It implements five of the most famous estimation techniques. It also obtain metrics to evaluate their performance and graphics that give you the ability to analyze at a glance.

The implemented system allows an analysis with several selected scenarios of maritime environment. These have positive results, both in terms of performance and consistency of the estimate. The conclusion of this analysis indicates that the IMM filter gives the best results.

# Índice de contenidos

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	Motivación . . . . .	1
1.2	Objetivos . . . . .	3
1.3	Estructura del documento . . . . .	4
<b>2</b>	<b>Estado del arte</b>	<b>5</b>
2.1	Sistemas de fusión de datos . . . . .	5
2.1.1	Clasificación de fusión de datos . . . . .	6
2.1.2	Técnicas de fusión de datos . . . . .	11
2.2	Modelos de los objetivos . . . . .	18
2.3	Estimación del estado en sistemas dinámicos . . . . .	19
2.3.1	Introducción . . . . .	19
2.3.2	Clasificación . . . . .	19
2.3.3	Estimadores principales . . . . .	22
2.3.4	Métricas de funcionamiento . . . . .	40
2.4	Fuentes de información . . . . .	42
2.4.1	Visión por computador . . . . .	43
2.4.2	Radares marítimos . . . . .	45
2.4.3	Sistema de identificación automática (AIS) . . . . .	46
2.5	Entornos de estimación existentes . . . . .	46
2.5.1	MATLAB Toolbox . . . . .	47
2.5.2	Tracker Component Library . . . . .	47
2.5.3	Conclusión . . . . .	48
<b>3</b>	<b>Sistema a desarrollar. Diseño</b>	<b>49</b>
3.1	Sistema de seguimiento global . . . . .	49
3.1.1	Introducción . . . . .	49
3.1.2	Sistema de fusión básico . . . . .	50
3.2	Sistema a desarrollar . . . . .	54
3.2.1	Fuentes de datos. Sensores . . . . .	55
3.2.2	Base de datos de soporte . . . . .	57
3.2.3	Nivel 0 - Pre-procesado de las fuentes de datos . . . . .	57
3.2.4	Nivel 1 - Control de objetivos . . . . .	60
3.2.5	Interfaz humano-computador . . . . .	61

<b>4</b>	<b>Sistema a desarrollar. Implementación</b>	<b>62</b>
4.1	Entorno de desarrollo . . . . .	62
4.1.1	Lenguaje de programación . . . . .	62
4.1.2	Computador utilizado . . . . .	63
4.2	Estructura del proyecto . . . . .	63
4.2.1	Extracción de trazas . . . . .	65
4.2.2	Script tiempo real . . . . .	65
<b>5</b>	<b>Comparativa</b>	<b>70</b>
5.1	Tipos de escenarios . . . . .	70
5.1.1	Selección de escenario . . . . .	71
5.2	Configuración de cada filtro . . . . .	74
5.3	Análisis entre filtros . . . . .	75
5.3.1	Escenario 1 . . . . .	75
5.3.2	Escenario 2 . . . . .	76
5.3.3	Escenario 3 . . . . .	79
5.3.4	Escenario 4 . . . . .	81
5.3.5	Conclusión . . . . .	83
<b>6</b>	<b>Conclusiones y líneas futuras</b>	<b>85</b>
6.1	Conclusiones . . . . .	85
6.1.1	Cumplimiento de los objetivos . . . . .	85
6.1.2	Conclusiones personales . . . . .	87
6.2	Líneas futuras . . . . .	87
	<b>Bibliografía</b>	<b>89</b>

# Índice de algoritmos

Alg.4.1 Extracción de trazas . . . . .	65
Alg.4.2 Filtrado todas las trazas . . . . .	66
Alg.4.3 processing3RealTime . . . . .	67
Alg.4.4 functionFilter . . . . .	68
Alg.4.5 seeGraphic . . . . .	68

# Índice de ecuaciones

Ec.2.2.1	Modelo del proceso . . . . .	18
Ec.2.2.2	Modelo de observación . . . . .	18
Ec.2.3.1	Función de verosimilitud . . . . .	23
Ec.2.3.2	Estimación de máxima verosimilitud . . . . .	23
Ec.2.3.3	KF. Predicción . . . . .	24
Ec.2.3.4	KF. Covarianzas a priori . . . . .	25
Ec.2.3.5	KF. Ecuación modelo de observación . . . . .	25
Ec.2.3.7	KF. Ganancia de Kalman . . . . .	25
Ec.2.3.8	KF. Innovación del estado . . . . .	25
Ec.2.3.9	KF. Vector de estado a posteriori . . . . .	26
Ec.2.3.10	KF. Matriz de covarianzas a posteriori . . . . .	26
Ec.2.3.11	EKF. Jacobiano del modelo de transición de estados . . . . .	27
Ec.2.3.12	EKF. Jacobiano del ruido del modelo del proceso . . . . .	27
Ec.2.3.13	EKF. Jacobiano del modelo de espacios observación-estado . . . . .	27
Ec.2.3.14	EKF. Jacobiano del ruido del modelo de la observación . . . . .	27
Ec.2.3.15	EKF. Predicción . . . . .	27
Ec.2.3.16	EKF. Covarianzas a priori . . . . .	27
Ec.2.3.17	EKF. Ecuación modelo de observación . . . . .	27
Ec.2.3.19	EKF. Ganancia de Kalman . . . . .	27
Ec.2.3.20	EKF. Innovación del estado . . . . .	27
Ec.2.3.21	EKF. Vector de estado a posteriori . . . . .	28
Ec.2.3.22	EKF. Matriz de covarianzas a posteriori . . . . .	28
Ec.2.3.25	UT. Cálculo de los sigma points . . . . .	29
Ec.2.3.26	UT. Propagación sigma points . . . . .	29
Ec.2.3.28	UT. Aproximación gaussiana función no lineal . . . . .	29
Ec.2.3.31	UT. Pesos aproximación gaussiana . . . . .	30
Ec.2.3.33	UKF. Vector de estado . . . . .	30
Ec.2.3.35	UKF. Estadísticos iniciales . . . . .	30
Ec.2.3.36	UKF. Cálculo sigma points . . . . .	30
Ec.2.3.37	UKF. Propagación sigma points . . . . .	30
Ec.2.3.39	UKF. Predicción vector de estado y covarianzas . . . . .	31
Ec.2.3.40	UKF. Recálculo puntos sigma . . . . .	31
Ec.2.3.42	UKF. Predicción de la observación . . . . .	31

Ec.2.3.45	UKF. Ganancia de Kalman . . . . .	31
Ec.2.3.48	UKF. Corrección final . . . . .	32
Ec.2.3.49	EnKF. Predicción muestras . . . . .	32
Ec.2.3.50	EnKF. Predicción . . . . .	33
Ec.2.3.51	EnKF. Matriz de covarianzas "ensemble" . . . . .	33
Ec.2.3.52	EnKF. Ganancia de Kalman . . . . .	33
Ec.2.3.53	EnKF. Ecuación de corrección . . . . .	33
Ec.2.3.54	EnKF. Ecuación de corrección . . . . .	33
Ec.2.3.55	IMM. Matriz de transición . . . . .	35
Ec.2.3.56	IMM. Probabilidad predicha . . . . .	35
Ec.2.3.57	IMM. Probabilidad de mezclado . . . . .	36
Ec.2.3.58	IMM. Entrada vector de estado cada modo . . . . .	36
Ec.2.3.60	IMM. Entrada matriz de covarianzas cada modo . . . . .	36
Ec.2.3.61	IMM. Verosimilitud . . . . .	36
Ec.2.3.62	IMM. Actualizada probabilidad modo . . . . .	37
Ec.2.3.63	IMM. Vector de estado final . . . . .	37
Ec.2.3.65	IMM. Matriz de covarianzas final . . . . .	37
Ec.2.3.66	PF. Predicción cada partícula . . . . .	39
Ec.2.3.68	PF. Actualización . . . . .	39
Ec.2.3.69	PF. Actualización . . . . .	39
Ec.2.3.70	PF. Resampling SIR . . . . .	40
Ec.2.3.71	Métricas. RMSE con vector de estado . . . . .	41
Ec.2.3.72	Métricas. RMSE con innovación . . . . .	41
Ec.2.3.74	Métricas. Matriz de información de Fisher . . . . .	41
Ec.2.3.77	Métricas. Valores para calcular la FIM . . . . .	41
Ec.2.3.78	Métricas. NEES . . . . .	42
Ec.2.3.79	Métricas. NIS . . . . .	42
Ec.5.2.1	IMM. Matriz de transición utilizada . . . . .	74
Ec.5.3.1	IMM. Matriz de transición ruidosa . . . . .	81



# Índice de figuras

Fig.2.1	Modelo DFD [1] . . . . .	7
Fig.2.2	Modelo del proceso de fusión JDL [2] . . . . .	9
Fig.2.3	Modelo del proceso de fusión JDL-DFIG [1] . . . . .	10
Fig.2.4	Clasificación arquitectura sistemas fusión . . . . .	11
Fig.2.5	Ejemplo resultado algoritmo estimación (UKF) vs posición real vs mediciones [3] . . . . .	13
Fig.2.6	Ejemplo contactos reales sin identificador (b) filtrados y asociados en 8 trazas (a) [4] . . . . .	14
Fig.2.7	Dos blancos con medidas comunes [5] . . . . .	16
Fig.2.8	Clasificación filtros de estimación . . . . .	20
Fig.2.9	Distribución normal [6] . . . . .	21
Fig.2.10	Funcionamiento filtro de Kalman . . . . .	24
Fig.2.11	Comparativa EKF, UKF, EnKF/PF [3] . . . . .	34
Fig.2.12	Funcionamiento filtro de partículas [7] . . . . .	38
Fig.2.13	Detección de barcos mediante visión por computador [8] .	43
Fig.2.14	Sistema OSS de la marca Chess-Dynamics . . . . .	44
Fig.2.15	Detección de vehículos mediante SAR [8] . . . . .	45
Fig.2.16	Estimación Tracker Component Library [9] . . . . .	48
Fig.3.1	Flujo de información de un sistema de fusión [10] . . . . .	50
Fig.3.2	Componentes nodo de fusión [11] . . . . .	51
Fig.3.3	Componentes nodo de fusión. Nivel 0 . . . . .	51
Fig.3.4	Componentes nodo de fusión. Nivel 1 - Estimación . . . . .	52
Fig.3.5	Componentes nodo de fusión. Nivel 1 - Asociación . . . . .	53
Fig.3.6	Sistema a desarrollar. Nivel 0 - Pre-procesado . . . . .	58
Fig.3.7	Sistema a desarrollar. Nivel 0 - Extracción de trazas . . . . .	58
Fig.3.8	Sistema de coordenadas OSRP-NED . . . . .	60
Fig.3.9	Sistema a desarrollar. Nivel 0 - Extracción de trazas . . . . .	60
Fig.4.1	Estructura Matlab . . . . .	63
Fig.4.2	Extracción de trazas . . . . .	65
Fig.5.1	Escenario 1. Contactos . . . . .	71
Fig.5.2	Escenario 2. Contactos . . . . .	72
Fig.5.3	Escenario 3. Contactos . . . . .	72

Fig.5.4	Escenario 4. Contactos . . . . .	73
Fig.5.5	Escenario 1. Gráfico XY . . . . .	75
Fig.5.6	Escenario 1. Velocidad en cada eje . . . . .	76
Fig.5.7	Escenario 1. Tiempo de cómputo . . . . .	76
Fig.5.8	Escenario 2. Gráfico XY . . . . .	77
Fig.5.9	Escenario 2. Velocidad en cada eje . . . . .	77
Fig.5.10	Escenario 2. UKF . . . . .	78
Fig.5.11	Escenario 2. UKF modificado . . . . .	79
Fig.5.12	Escenario 2. UKF comparativa RMSE . . . . .	79
Fig.5.13	Escenario 3. Gráfico XY . . . . .	80
Fig.5.14	Escenario 3. Gráfico XY PF zoom . . . . .	80
Fig.5.15	Escenario 3. Probabilidades de modo IMM y NIS . . . . .	81
Fig.5.16	Escenario 4. Comparación IMM Gráfico XTime . . . . .	82
Fig.5.17	Escenario 4. Gráfico XY . . . . .	82
Fig.5.18	Escenario 4. Gráfico XY curva . . . . .	83
Fig.5.19	Escenario 4. Velocidad en cada eje ampliada . . . . .	83
Fig.5.20	Escenario 4 Probabilidades de modo IMM y NIS . . . . .	84

# Índice de tablas

Tab.2.1	Técnicas de fusión de datos . . . . .	12
Tab.3.2	Parámetros de AIS . . . . .	56
Tab.4.2	Componentes PC utilizado . . . . .	63
Tab.5.2	Características periodo escenarios . . . . .	73

# Glosarios

## Nomenclatura

This list of acronyms only specifies the most important ones

- $u_{t-1}$  Vector de entradas de la iteración anterior. Utilizado en: 24, 26, 27
- $P$  Matriz de covarianzas Utilizado en: 30, 31, 35–37, 67, 74
- $C$  Matriz de covarianzas del conjunto ensemble. Sustituye a la matriz  $Q$  Utilizado en: 33
- $P_t^-$  Matriz de covarianzas predicha a priori. Utilizado en: 25–28, 31
- $P_t$  Matriz de covarianzas en esta iteración. Utilizado en: 26, 28, 31, 35, 37, 42
- $P_{t-1}$  Matriz de covarianzas calculada en la iteración anterior. Utilizado en: 25, 27
- $P_x$  Matriz de covarianzas del vector  $x$  Utilizado en: 28, 29
- $P_z$  Matriz de covarianzas del vector  $z$  Utilizado en: 29
- $E$  Aplica la función Esperanza, explicada en el glosario Utilizado en: 30, 40
- $I$  Matriz identidad. Todos los valores son cero, excepto su diagonal que son unos. Utilizado en: 26, 28
- $\tilde{z}_t$  Innovación entre la medida y los valores de medida estimados a priori. Utilizado en: 25, 27, 28, 31, 37, 40–42
- $A$  Jacobiano del modelo de transición de estados. Utilizado en: 26, 27
- $H$  Jacobiano del modelo de espacios observación-estado. Utilizado en: 27
- $V$  Jacobiano del ruido del modelo de la observación. Utilizado en: 27
- $W$  Jacobiano del ruido del modelo del proceso. Utilizado en: 27
- $K_t$  Ganancia de Kalman. Utilizado en: 25–28, 31, 33
- $A$  Matriz de transición de estados. Relaciona el estado  $t-1$  con el  $t$ . Utilizado en: 24, 25, 41

$A^T$	Matriz A transpuesta. Utilizado en: 25
$B$	Matriz opcional que controla entradas del vector $u$ a introducir en el vector de estado $x$ . Utilizado en: 24
$H$	Matriz que relaciona el estado $x$ con la medida $z$ . Utilizado en: 25–28, 33, 40, 41, 67
$H^T$	Matriz H transpuesta. Utilizado en: 25, 33
$Q$	Matriz de covarianzas del ruido del estado. Utilizado en: iv, 25, 27, 28, 31, 33, 41, 74
$R$	Matriz de covarianzas del ruido de la medida. Utilizado en: 25, 27, 28, 31–33, 39, 41, 67, 74
$v$	Vector de ruido de la medida. Utilizado en: v, 27, 30, 31
$v_t$	Vector de ruido de la medida en esta iteración. Utilizado en: 18, 25, 33
$w$	Representa el ruido del modelo (o proceso). Utilizado en: v, 27, 30, 39
$w_t$	Representa el ruido del modelo (o proceso) en esta iteración. Utilizado en: 18, 24, 32
$w_{t-1}$	Representa el ruido del modelo (o proceso) en la iteración anterior. Utilizado en: 18
$\mu$	Probabilidad del filtro IMM de estar un blanco representado por un modelo $i$ Utilizado en: 35–37
$z_t^-$	Modelo de observación predicho a priori. Utilizado en: 25, 27, 31
$z_t$	Modelo de observación en esta iteración. Utilizado en: 18, 25, 27, 31, 33, 40, 42
$Z_t$	Modelo de observación en esta iteración, preparado para el conjunto ensemble Utilizado en: 33
$h$	Función (lineal o no lineal) que rige la relación entre el estado anterior y el actual. Utilizado en: 18, 24, 27, 31, 39
$z$	Modelo de observación. Utilizado en: 29, 39
$f$	Función (lineal o no lineal) que rige la relación entre el estado anterior y el actual. Utilizado en: 18, 24, 26, 27, 30, 32, 39
$\mathcal{X}$	Sigma points utilizados en la predicción del filtro UKF. Utilizado en: v, 29–31
$\mathcal{Z}$	Sigma points tras pasar por la función no lineal en la predicción del filtro UKF. Utilizado en: 29, 31
$\mathcal{X}^v$	Sigma points relativos al ruido de la medida Utilizado en: 30, 31
$\mathcal{X}^w$	Sigma points relativos al ruido del modelo Utilizado en: 30
$\mathcal{X}^x$	Sigma points relativos al estado Utilizado en: 30, 31
$x_t^-$	Vector de estado predicho a priori. Utilizado en: 24, 25, 27, 28, 31, 32, 40

$x_t$	Vector de estado en la iteración actual. Utilizado en: 18, 25, 28, 31–33, 35, 37
$x_{t-1}$	Vector de estado calculado en la iteración anterior. Utilizado en: 18, 24, 26, 27, 30
$x$	Vector de estado. Utilizado en: v, 26–28, 30, 31, 35–37, 39, 40
$X$	Conjunto de muestras del filtro EnKF Utilizado en: 32, 33
$\tilde{X}$	Diferencia entre la estimación y el vector de estado real Utilizado en: 40–42
$w$	Peso de cada partícula $i$ Utilizado en: 39
$\alpha$	Peso de cada partícula $i$ normalizado respecto al sumatorio Utilizado en: 39, 40
$W$	Sumatorio del peso de todas las partículas Utilizado en: 39
$\hat{x}$	Esperanza del vector $x$ Utilizado en: 30
$\bar{x}$	Media del vector $x$ Utilizado en: 28, 29
$z_t^-$	Vector de estado predicho a priori. Utilizado en: 31
$\bar{z}$	Media del vector $z$ Utilizado en: 29

## Acrónimos

This list of acronyms only specifies the most important ones

- AI (Artificial Intelligence)** Inteligencia Artificial Primer uso en pág. 44
- AIS (Automatic Identification System)** Sistema de identificación automática Primer uso en pág. xi, 46, 54–58, 65, 86, 87
- ANN (Artificial Neural Network)** Redes de neuronas artificiales Primer uso en pág. 16
- CKF (Cubature Kalman Filter)** Filtro de Kalman "Cubature" Primer uso en pág. 32
- COG (Course over Ground)** Curso sobre el mar Primer uso en pág. 56
- DAI-DAO (Data in - Data out)** Entrada de datos - Salida de datos Primer uso en pág. 7
- DAI-FEO (Data in - Feature out)** Entrada de datos - Salida de característica Primer uso en pág. 7
- DEI-DEO (Decision in - Decision out)** Entrada de decisión - Salida de decisión Primer uso en pág. 7
- DFD (Data Feature Decision)** Datos Característica Decisión Primer uso en pág. 6
- DFIG (Data Fusion Information Group)** Grupo de fusión de datos e información Primer uso en pág. 9
- DMA (Danish Maritime Authority)** Autoridad Marítima Danesa Primer uso en pág. 55
- DT (Árbol de decisión)** Decision Tree Primer uso en pág. 16
- EKF (Extended Kalman Filter)** Filtro de Kalman Extendido Primer uso en pág. 26, 28, 32, 37, 40, 60, 74–76, 78, 80–84
- EnKF (Ensemble Kalman Filter)** Filtro de Kalman Ensemble Primer uso en pág. 26, 32, 60, 74–77, 83, 84
- FEI-DEO (Feature in - Decision out)** Entrada de característica - Salida de decisión Primer uso en pág. 7
- FEI-FEO (Feature in - Feature out)** Entrada de característica - Salida de característica Primer uso en pág. 7

- FIM (Fisher Information Matrix)** Matriz de información de Fisher Primer uso en pág. 41
- GHF (Gauss-Hermite Filter)** Filtro de Gauss-Hermite Primer uso en pág. 32
- GHz (Gigahertz)** Gigahercio Primer uso en pág. 45
- GPS (Global Positioning System)** Sistema de Posicionamiento Global Primer uso en pág. 46, 75
- HCI (Human Computer-Interaction)** Interacción humano-computador Primer uso en pág. 9, 18, 55
- ICT (Information and Communication Technologies)** Tecnologías de la información y comunicación Primer uso en pág. 1
- IFF (Identification Friend or Foe)** Identificador amigo-enemigo Primer uso en pág. 46
- IMM (Interacting Multiple Modes)** Filtro de Múltiples modos inter-operantes Primer uso en pág. ii, iii, 34, 35, 37, 60, 69, 74–76, 80–84, 88
- IMO (International Maritime Organization)** Organización Marítima Internacional Primer uso en pág. 46, 55
- JDL (Joint Directors of Laboratories)** Unión de directores de laboratorios Primer uso en pág. 5, 8, 9, 11, 49, 50, 54, 57
- JPDA (Joint Probabilistic Data Association)** Asociación Probabilística Conjunta de Datos Primer uso en pág. 15
- KF (Kalman Filter)** Filtro de Kalman Primer uso en pág. 21, 23, 26–28, 30, 32
- LQE (Linear-Quadratic state Estimator)** Estimador lineal cuadrático Primer uso en pág. 23
- MCKF (Monte-Carlo Kalman Filter)** Filtro de Kalman Monte-Carlo Primer uso en pág. 33, 34
- MHT (Multi-Hypothesis Tracking)** Seguimiento con Hipótesis Múltiples Primer uso en pág. 15
- MLE (Maximum Likelihood Estimation)** Estimación de máxima verosimilitud Primer uso en pág. 23
- MMSE (Minimum Mean Squared Error)** Mínimo error cuadrático medio Primer uso en pág. 23, 26



**MMSI (Maritime Mobile Service Identity)** Identificador de servicio móvil marítimo Primer uso en pág. 46, 58, 64, 71, 72

**MoE (Measures of Effectiveness)** Medidas de Efectividad Primer uso en pág. 18

**MoP (Measures of Performance)** Medida del rendimiento Primer uso en pág. 18

**MSE (Mean Squared Error)** Error cuadrático medio Primer uso en pág. 23

**NASA (National Aeronautics and Space Administration)** Administración Nacional de la Aeronáutica y del Espacio Primer uso en pág. 26

**NED (North East Down)** Norte, este, abajo Primer uso en pág. 59

**NEES (Normalized Estimation Error Squared)** Error cuadrático de estimación normalizado Primer uso en pág. 42

**NIS (Normalized Innovation Squared)** Innovación cuadrática normalizada Primer uso en pág. 42, 83

**NN (Nearest Neighbour)** Vecino más próximo Primer uso en pág. 15

**OSRP (Own Ship Reference Point)** Punto de referencia el propio barco Primer uso en pág. 59

**OSS (Optronic Sensor Systems)** Sistema de sensores optrónicos Primer uso en pág. 44

**PCRLB (Cota inferior posterior de Cramer-Rao)** Posterior Cramer-Rao Low Bound Primer uso en pág. 41

**PDA (Probabilistic Data Association)** Asociación Probabilística de Datos Primer uso en pág. 15

**PDF (Probability Density Function)** Función Densidad de Probabilidad Primer uso en pág. 20, 21, 26, 29, 32, 38, 40, 42, 74

**PF (Particle Filter)** Filtro de partículas Primer uso en pág. 37, 38, 61, 69, 74, 76, 79, 81

**RMSE (Root Mean Square Error)** Raíz del error cuadrático medio Primer uso en pág. 40, 78

**ROT (Rate of Turn)** Ratio de giro Primer uso en pág. 56

**SAR (Synthetic Aperture Radar)** Radar de apertura sintética Primer uso en pág. 44

- SHF (Super High Frequency)** Super alta frecuencia Primer uso en pág. 45
- SIR (Sequential Important Resampling)** Muestreo Secuencial Ponderado con Selección Primer uso en pág. 38, 39
- SOG (Speed over Ground)** Velocidad sobre el mar Primer uso en pág. 56
- SSR (Secondary surveillance radar)** Radar secundario Primer uso en pág. 46
- UKF (Unscented Kalman Filter)** Filtro de Kalman “Unscented” Primer uso en pág. 26, 28, 32, 60, 74–77, 79, 81, 84
- UPF (Unscented Particle Filter)** Filtro de Partículas “Unscented” Primer uso en pág. 32
- UT (Unscented Transformation)** Transformación “Unscented” Primer uso en pág. 28, 30
- VHF (Very High Frequency)** Muy alta frecuencia Primer uso en pág. 46

# Capítulo 1

## Introducción

En este primer apartado de la tesis se presenta, en la sección 1.1 la motivación del proyecto, que sirve a su vez como introducción al tema desde una perspectiva general. A continuación, en la sección 1.2 se exponen los objetivos propuestos a cumplir durante el desarrollo de esta tesis. Finalmente, en la sección 1.3 se realiza una breve descripción de la estructura del resto del documento.

### 1.1 Motivación

Los seres humanos realizamos acciones en función de ciertos parámetros de entrada. Si tengo hambre, quiero comer. Pero las reglas en el mundo no suelen estar definidas de manera tan concisa.

Por ello, buscamos fundamentar las acciones que tomamos con reglas en función de ciertos parámetros de entrada. Esto se conoce como toma de decisiones de manera informada, y es una de las principales ramas de las Tecnologías de la información y comunicación (ICT). Un sistema inteligente de apoyo a la toma de decisiones es el mejor aliado para cualquier tipo de decisión, al tener una máquina la capacidad de retener y procesar información de manera más rápida que un humano, y, además, con menor probabilidad de error. Disminuir el factor humano es una tendencia, tanto para el apoyo a la toma de decisiones, como para ejecutar la decisión en sí.

El problema es que construir un sistema de apoyo a la toma de decisiones adecuado para cada entorno no es trivial y requiere una gran cantidad de trabajo detrás para ir perfeccionando su funcionamiento.

Por otro lado, el funcionamiento de cualquier sistema podemos resumirlo como una serie de valores de entrada al sistema, los cuales son procesados y proporcionan una salida. Idealmente, esta entrada serán valores exactos y fiables, sin probabilidad de error ni variaciones inesperadas.

En problemas desarrollados sobre entornos abiertos, como puede ser la

conducción autónoma o sistemas de vigilancia, los valores de entrada son proporcionados por sensores o cámaras, los cuales no son completamente precisos ni fiables (roturas, interferencias, ruido, fallos de comunicación, etc.), e incluso pueden ser atacados directamente. Además, la precisión y fiabilidad de estos va directamente ligada al coste de los sensores. Por ello, los sistemas que se desenvuelven en el mundo real buscan ser lo más robustos posible para funcionar con garantías ante cualquier tipo de problema.

La literatura ha demostrado que implementar sistemas que parametrizan el entorno en el que se desplazan los vehículos, ya sean marítimos, terrestres, o voladores, les hace herramientas muy útiles para los humanos en múltiples ámbitos. Por ejemplo, en vehículos militares no autónomos, conocer la posición actual y futuras trayectorias de posibles enemigos a media y larga distancia es imprescindible para que el encargado realice una estrategia u otra, mientras que en un vehículo autónomo esta información se puede utilizar tanto para resolver tareas (EJ: un dron que vigile si un depredador va a realizar una trayectoria hacia un rebaño) como para apoyar al propio sistema de conducción autónoma (girar a la derecha y no golpearse con el entorno o con la trayectoria de otro vehículo).

Bajo esta premisa se han desarrollado muchos tipos de sistemas para monitorizar el entorno del vehículo. La implementación más habitual y fiable son los sistemas de fusión de datos multisensor. Estos utilizan varios tipos de sensores para captar el entorno cercano, con características complementarias o redundantes, y unifican toda la información para hacer la interpretación más fiable posible del entorno del vehículo. Además, si son suficientemente potentes, son capaces de extraer conocimiento del escenario, prediciendo el comportamiento de vehículos cercanos y aconsejar sobre acciones a tomar.

De esta manera, aunque un sensor no funcione en unas determinadas condiciones, como puede ser la niebla, los demás sí siguen funcionando y permitiendo un funcionamiento adecuado.

Debido a su importancia, tanto militar como de transporte de mercancías [12] [2], y su amplia superficie para incorporar sensores, el uso de sistemas de fusión multisensor en el entorno marítimo es muy importante para realizar una monitorización y vigilancia sobre amplias distancias.

Este entorno cuenta la mayor diversidad de vehículos, desde motos acuáticas de dos metros de tamaño que alcanzan hasta 65 nudos de velocidad, a portaaviones de más de 300 metros de eslora. Esta diversidad hace que las maniobras que puede realizar cada vehículo sean muy diferentes, y por ello un buen sistema de monitorización marítimo debe ser capaz detectar con precisión cualquier tipo de maniobra que pueda realizar cada tipo de

vehículo.

La detección y seguimiento de los objetivos la realizan denominados estimadores o filtros de posición, que son un subsistema dentro de un sistema de fusión. Estos algoritmos utilizan una serie de fundamentos matemáticos y probabilísticos para, a partir de los contactos obtenidos de varios sensores, y entendiendo que la señal obtenida está corrompida por ruido que le hace dejar de ser precisa, obtener la posición más probable en la que está el objetivo realmente.

Estos algoritmos no son óptimos, al no proporcionar siempre la mejor posición posible, pero cuentan con parámetros que les permiten ajustar su funcionamiento, premiando unas características que le faciliten seguir unas maniobras concretas. La optimización de estos parámetros no es trivial, por lo que se sigue realizando un gran esfuerzo en mejorar los resultados que se pueden obtener de un estimador, ya sea de manera global para cualquier tipo de maniobra, o para solventar un problema específico.

En esta tesis se desarrolla un framework completo con el que realizar análisis entre varias técnicas avanzadas de estimación y filtrado de posición en el entorno marítimo, o en cualquier otro entorno en dos dimensiones. El análisis busca encontrar los mejores estimadores y parámetros de configuración para distintos escenarios marítimos, de manera que un sistema de seguimiento completo pueda ser optimizado.

## 1.2 Objetivos

Partiendo de los puntos tratados en la introducción, se ha definido la siguiente lista de objetivos para este trabajo:

- Analizar y comprender los distintos estimadores y filtros de posición más utilizados en la literatura y compararlos.
- Obtener una serie de contactos de sensores marítimos reales para obtener trazas sobre las que aplicar estimadores de posición.
- Desarrollar un framework de estimación que permita filtrar las trazas anteriormente extraídas.
- Desarrollar un componente del framework que genere gráficos y métricas de comparación entre estimaciones de una misma traza.
- Analizar los resultados obtenidos y concluir las fuerzas y debilidades de cada tipo de filtro para los escenarios propuestos.

## 1.3 Estructura del documento

Una vez realizada la introducción de la tesis, la estructura del documento es expuesta. Mediante dos partes se va a exponer todo el trabajo realizado.

En la primera parte, comenzando por el capítulo 2, se realiza un acercamiento teórico tanto a los sistemas de fusión de manera global, haciendo hincapié en el trabajo a realizar, los filtros de estimación. Además, en el capítulo 3 se explica de manera teórica el sistema a desarrollar para esta tesis y su compatibilidad con un futuro sistema de fusión de datos multisensor.

La segunda parte del trabajo expone el trabajo desarrollado, tanto del framework desarrollado de estimación de posición en el capítulo 4, como de un análisis que demuestra las capacidades del mismo en el capítulo 5.

Finalmente, se realiza una conclusión de la tesis en el capítulo 6, exponiendo las capacidades del sistema desarrollado, los conocimientos aprendidos en el trabajo y las futuras expansiones del sistema.

Recapitulando, la tesis se estructura mediante los siguientes capítulos:

- Capítulo 1. Introducción.
- Capítulo 2. Estado del arte.
- Capítulo 3. Sistema a desarrollar. Diseño.
- Capítulo 4. Sistema a desarrollar. Implementación.
- Capítulo 5. Comparativa.
- Capítulo 6. Conclusiones y líneas futuras.

## Capítulo 2

# Estado del arte

En esta sección vamos a realizar una completa introducción para entender los fundamentos de los temas tratados en el presente documento, garantizando a cualquier lector de este la perfecta comprensión de los futuros apartados.

En la sección 2.1 se introducen los sistemas de fusión de datos de manera teórica. A continuación, en la sección 2.2 se describen los modelos matemáticos que definen los comportamientos de un objetivo.

La estimación del estado se introduce de manera específica en la sección 2.3, haciendo un claro énfasis en su clasificación y las principales técnicas que existen en la actualidad. También se exponen las métricas para analizar su funcionamiento más habituales.

Después, en la sección 2.4 se introducen los tipos de fuentes de datos más comunes a la hora de realizar un seguimiento de objetivos marítimos.

Por último, en la sección 2.5, se comparan las soluciones existentes de fácil acceso para el desarrollo y testeo de múltiples estimadores de posición.

### 2.1 Sistemas de fusión de datos

La fusión de datos, también conocida como integración de datos ha adquirido una mayor relevancia los últimos años debido a que el avance de la tecnología ha disminuido la barrera de entrada, tanto económica como computacional, para permitir su funcionamiento en tiempo real.

A pesar de ello, el término fusión de datos no es nuevo, sino que tiene más de 40 años. En 1985 se creó el U.S. Joint Directors of Laboratories (JDL) Data Fusion Group, grupo formado por expertos de esta área encargado de definir y modelar esta técnica. En 1987 emitieron la considerada primera definición de fusión de datos [13]:

Un proceso multinivel relativo a la asociación, correlación, combinación de datos e información de una o varias fuentes para

obtener una posición refinada, identificar estimaciones y evaluar situaciones, amenazas y su significado, en un instante concreto.

Otra definición popular es la proporcionada por Hall y Llinas [2]:

Las técnicas de fusión de datos combinan datos de múltiples sensores e información asociada para mejorar la precisión y lograr mejores inferencias que las que se podrían lograr utilizando un único sensor.

Como indican ambas definiciones, lo más relevante puede considerarse la combinación de datos provenientes de uno o varios sensores para mejorar la precisión de una medición.

El ejemplo más sencillo para explicar qué es un sistema de fusión de datos es un ser humano. Contamos con multitud de sensores (dos ojos, nariz, boca, tacto, etc.) cuyas medidas se envían al cerebro para que realice una visión global de lo percibido.

### **2.1.1 Clasificación de fusión de datos**

A pesar de ello, este concepto abarca mucho más de la unión de estas definiciones. Es un concepto multidisciplinar por lo que existen diversas clasificaciones en función del enfoque con el que se discretee.

#### **2.1.1.1 Clasificación datos de entrada**

En función de los datos de entrada, Durrant-Whyte [14] propone la siguiente clasificación:

- Complementarios: la información combinada no se solapa en ningún aspecto, aumentando completamente la información que se tiene sobre el escenario global. Por ejemplo: dos cámaras que apuntan en dos direcciones distintas.
- Cooperativas: obtienen la información de un mismo objetivo, pero su fusión produce información nueva. El ejemplo más claro es una videocámara, que fusiona el audio y el vídeo.
- Redundantes: las fuentes de datos aportan información solapada, proporcionando una repetición que aumenta la confianza en el resultado ante posible ruido. Por ejemplo: varias cámaras que apuntan al mismo objetivo.

#### **2.1.1.2 Clasificación entrada salida**

En función del objetivo buscado a partir de la entrada, Dasarathy [15] propone cinco categorías de componentes dentro de su modelo Datos Característica Decisión (DFD). Este modela la propagación de información de manera bidireccional tras la acción de cada modelo (ver figura Fig.2.1).



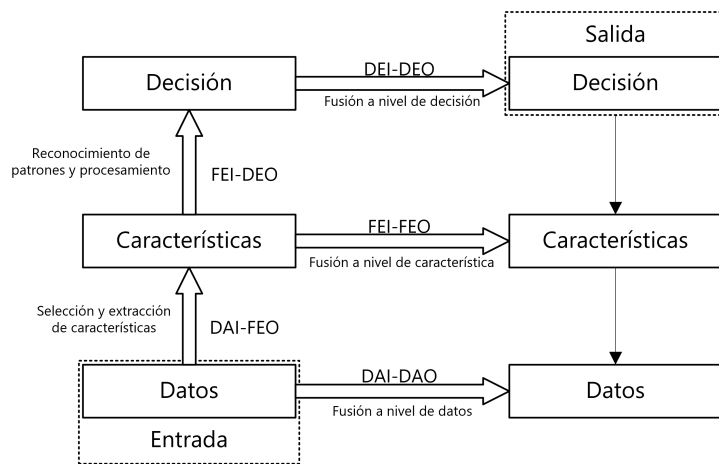


Fig.2.1: Modelo DFD [1]

Los componentes propuestos son los siguientes:

- Entrada de datos - Salida de datos (DAI-DAO): Se reciben los datos en bruto (raw data en inglés) y la salida del sistema de fusión son datos procesados, pero en bruto también. Mantiene una alta dimensionalidad de variables. EJ: se obtiene una posición de varios sensores distintos y se fusiona en una única posición estimada.
- Entrada de datos - Salida de característica (DAI-FEO): Obtiene los datos en bruto y devuelve una serie de características que explican dichos datos. EJ: De las imágenes de varias cámaras se obtiene que no existe amenaza.
- Entrada de característica - Salida de característica (FEI-FEO): se realiza la fusión directamente de características. Por ejemplo, en vez de obtener medidas de los sensores, los sensores filtran su salida, porque no son comparables de manera sencilla entre sí, y te ofrecen una característica, y el sistema de fusión decide las características que explican mejor esta escena.
- Entrada de característica - Salida de decisión (FEI-DEO): A partir de una serie de características obtenidas de cada sensor, se toma la decisión a realizar en dicho escenario. EJ: Cualquier sistema de reconocimiento de patrones, como el reconocimiento facial. Se divide la cara obtenida en cejas, ojos, nariz, boca, etc. y se observa si concuerda con la imagen original. Si concuerdan todas las facetas, se desbloquea.
- Entrada de decisión - Salida de decisión (DEI-DEO): A partir de las decisiones de entrada previas, obtiene mejores decisiones, ratificando las previas u otras nuevas.

### 2.1.1.3 Clasificación del modelo JDL

Esta es la clasificación más popular y utilizada. La propuso el ya mencionado grupo estadounidense, y clasifica los procesos de los sistemas de fusión en cinco niveles en función de su complejidad. Los dos primeros se centran en el bajo nivel mientras que los otros tres utilizan esos resultados para trabajar sobre una visión más general.

Diversos investigadores han propuesto matices y refinamientos del modelo, como Llinas et al. [16] matizando de manera general el modelo, con el fin de que este satisfaga las necesidades de todo tipo de sistema de fusión actual y alejarlo del enfoque militar inicial. Se propone las siguientes categorías que trabajan a bajo nivel:

- Nivel 0 – preprocesado de fuentes de entrada: trabaja a bajo nivel, con las medidas de los sensores. Lo que realiza es puramente gestión de los datos en bruto, permitiendo su proceso al resto de niveles.
- Nivel 1 – refinamiento de objetos: a partir de los datos de los sensores, aplicar técnicas que generan nueva información, ya sean características nuevas (clustering, asociación de contactos), refinamientos de las medidas obtenidas (estimadores de posición), o modifican la existente (eliminación de información incorrecta).

Las otras tres utilizan información de alto nivel:

- Nivel 2 – evaluación de la situación: utilizando los datos resultantes del nivel 1, ser capaz de comparar los objetos o nueva información entre sí para extraer características y evaluar patrones que expliquen el entorno.
- Nivel 3 – evaluación del impacto: a partir de la situación generada en el nivel 2, el sistema debe ser capaz de poner en perspectiva dicha situación e identificar si en el futuro se traducirá en un riesgo o una oportunidad, junto a su probabilidad estimada.
- Nivel 4 – refinamiento del proceso de fusión: a partir de la información obtenida en los procesos de los niveles anteriores, utilizarla para ajustar dichos procesos en el futuro, tratando de ajustar los recursos y optimizar los resultados. Puede simplemente para monitorizar y asesorar al humano, o permitir refinar por sí mismo el sistema.

Para el funcionamiento del sistema de fusión el JDL propone como imprescindibles tres componentes distintos:

- Fuentes de entrada: dependiendo del tipo de entrada que se utilice y su cantidad, tendrá una clasificación mayor o menor. Parámetros introducidos a mano por una persona, sensores, conexiones de bases de datos complejas, etc. son algunos ejemplos de entradas utilizadas.
- Interacción humano-computador (HCI): la capacidad de gestión por parte de un operario sobre el sistema de fusión y sus alertas o notificaciones sobre su funcionamiento afectan a su clasificación de JDL.
- Sistema de gestión de la base de datos: la última característica es la fiabilidad de la base de datos de lectura y escritura del sistema de fusión.

Con estos componentes y los cinco niveles de fusión, se genera el modelo que propone el JDL, mostrado en la figura Fig.2.2 para sistemas de fusión de datos.

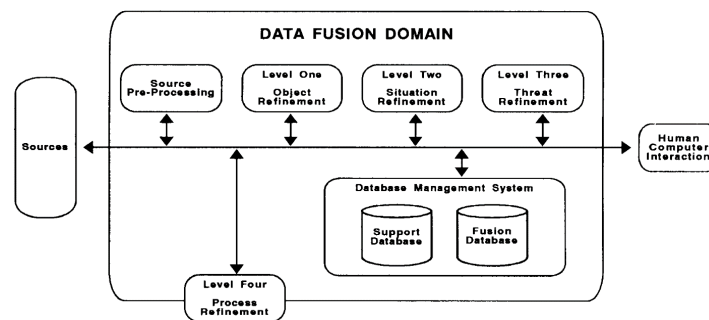


Fig.2.2: Modelo del proceso de fusión JDL [2]

**2.1.1.3.1 Modelo DFIG** Años después de la creación del modelo JDL, Blasch y Plano [1] propusieron un quinto nivel para dotar al sistema de la capacidad de mediación del usuario. Un sistema cuyo objetivo es apoyar la toma de decisiones del humano debe fomentar la interacción entre este y el responsable.

De esta manera, el quinto nivel permite al usuario realizar refinamientos sobre todos los niveles del sistema de fusión de manera cognitiva. Con esta inclusión, se puede ajustar el funcionamiento del sistema según las necesidades de cada tipo de usuario e introducir en el sistema su conocimiento en tiempo real.

Posteriormente, se formó una extensión del grupo JDL, denominada Grupo de fusión de datos e información (DFIG), la cual añadió a su modelo este quinto nivel. Este modelo se muestra en la figura Fig.2.3.

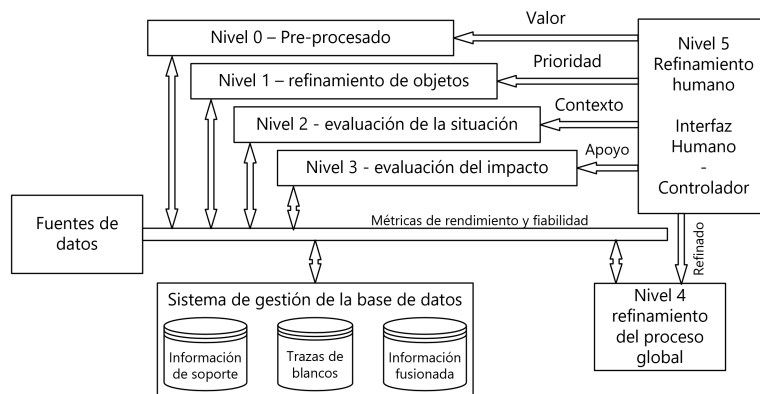


Fig.2.3: Modelo del proceso de fusión JDL-DFIG [1]

#### 2.1.1.4 Clasificación según la arquitectura

Otro tipo de clasificación de sistemas de fusión de datos pueden utilizar el lugar donde se desarrolla el proceso de fusión, es decir, todo el cómputo para obtener los resultados. Se cuentan con cuatro tipos de arquitecturas distintas:

- **Arquitectura centralizada:** se tiene un único lugar de procesamiento, formado por uno o más ordenadores. De esta manera, los sensores deben tener comunicación directa con este para realizar su única tarea, obtener la medición y enviarla al ordenador central y este almacenar toda la información en sus bases de datos.

Este enfoque conlleva claras ventajas, como la necesidad de instalar un único centro de cómputo y la simplicidad a la hora de comunicaciones que adquiere. Eso sí, requiere que todos los sensores y sus comunicaciones sean fiables y estén sincronizadas para evitar problemas en el proceso de fusión. Además, requiere un alto nivel de seguridad al ser el único sistema de que contiene la información.

- **Arquitectura descentralizada:** esta arquitectura utiliza una red de nodos de procesamiento, los cuales procesan su propia fusión de datos. Esta fusión se realiza tanto de los sensores que tiene disponibles dicho nodo, como de la información que le comparten el resto de los nodos mediante un sistema de comunicación interno. Este tipo de arquitecturas utilizan otro tipo de algoritmos para incorporar el conocimiento del resto de nodos al proceso de fusión.

Tiene como principales ventajas el hecho de ser independientes por sí mismo cada nodo, pudiendo funcionar sin necesidad del resto de nodos, pero aumenta drásticamente la complejidad del proceso de fusión y de la comunicación.

- **Arquitectura distribuida:** esta arquitectura realiza un proceso intermedio entre la arquitectura centralizada y descentralizada. Existen nodos intermedios entre los sensores y el centro de fusión que realizan la parte inicial del proceso de fusión, es decir, la parte de bajo nivel, y posteriormente es enviado dicho resultado al centro de fusión para que continúe. De esta manera, cada nodo intermedio es totalmente independiente del resto y únicamente emite la información filtrada.
- **Arquitectura jerárquica:** finalmente, esta arquitectura permite realizar combinaciones de nodos descentralizados y distribuidos mediante una jerarquía que determine su relación y orden en el proceso de fusión para dar lugar a esquemas más complejos y robustos.

Como conclusión sobre las distintas arquitecturas, hay que destacar que ninguna es mejor que las demás, sino que deben utilizarse en función de las necesidades del proyecto sobre el que incorporar el sistema de fusión. Si se cuenta con un grupo de drones, seguramente la mejor solución es una arquitectura descentralizada para que compartan información, pero si es una flota de la armada comandada por un portaaviones, quizá una arquitectura distribuida con un centro de fusión más complejo es recomendable. Hay que buscar el equilibrio recomendado entre las capacidades computacionales de cada nodo, la seguridad y la independencia de cada vehículo (si hay varios). De esta forma, Benaskeur et al. [17], al proponer una arquitectura distribuida, muestra la clasificación de la figura Fig.2.4 según las necesidades:

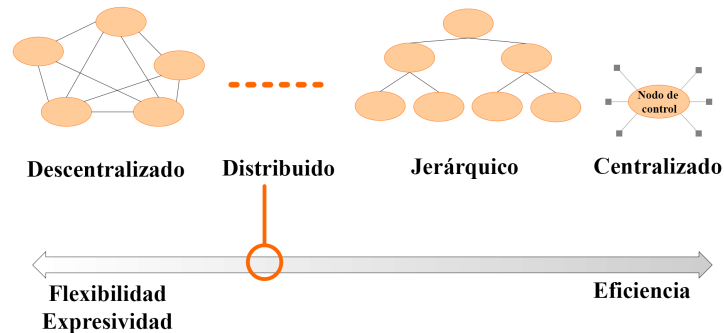


Fig.2.4: Clasificación arquitectura sistemas fusión

### 2.1.2 Técnicas de fusión de datos

Una vez conocidas las distintas clasificaciones sobre, Hall & Llinas [2] propusieron una tabla de técnicas y algoritmos habituales para cada nivel de proceso del modelo JDL. Al tener bastantes años, otros investigadores como

Smith & Singh [18] o Castanedo [19], han realizado nuevos análisis más actualizados en el tiempo. Incluyendo las técnicas que comentan, esta tabla expone perfectamente el tipo de componentes más habitual de un sistema de fusión:

<b>Nivel de proceso JDL</b>	<b>Tipo de técnica</b>	<b>Técnica</b>
<b>Nivel 1 - refinamiento de objetos</b>	Alineación de datos	Transformación de coordenadas Ajustes de unidades
	Estimación de posición y atributos	Técnicas de estimación
	Correlación de datos / objetos	Técnicas de asociación
	Identificación del objeto	Extracción de características
		Reconocimiento de patrones
	Clustering	
<b>Nivel 2 - evaluación de la situación</b>	Agregación de objetos Interpretación de eventos/actividades Interpretación del contexto	Sistemas basados en el conocimiento <ul style="list-style-type: none"> <li>• Sistemas expertos</li> <li>• Ontologías</li> <li>• Inferencia</li> </ul>
<b>Nivel 3 - evaluación del impacto</b>	Estimación agregada Predicción del futuro Evaluación múltiples perspectivas	Dempster-Shafer Redes de neuronas
<b>Nivel 4 - refinamiento del proceso de fusión</b>	Evaluación del rendimiento	Técnicas de optimización
	Control del proceso de fusión	multiobjetivo
	Gestión de la misión	Técnicas de planificación

Tab.2.1: Técnicas de fusión de datos

#### 2.1.2.1 Técnicas nivel 1

Es el nivel más estudiado de todos, al ser totalmente fundamental para la correcta ejecución del resto. Como vemos en Tab.2.1, se descompone en cuatro subcomponentes imprescindibles si se quiere construir un sistema de fusión completo.

**2.1.2.1.1 Alineación de datos** Es imprescindible que los contactos de los radares entren al sistema de fusión preparados para ser comparables con el resto de los sensores. Este paso comúnmente se considera el parte del nivel 0, en función de si se almacenan los contactos originales o se transforman directamente a las unidades y coordenadas utilizadas en el sistema.

En este apartado es importante introducir, en determinados sensores que no proporcionan la identificación del vehículo detectado, un elemento denominado "tracker", que asocia dicho contacto a un identificador unívoco a los contactos de dicho barco. De esta manera, posteriormente, se pueden asociar los contactos de varios sensores entre ellos.

**2.1.2.1.2 Estimación de posición y atributos** Las mediciones de cada sensor detectan objetos reales y devuelven su posición detectada, pero estas están sujetas a ruido y es necesario filtrarlas para obtener su posición real, o al menos, una más fiable que la original, para monitorizar correctamente el escenario. Estos algoritmos, a partir de los contactos asociados a un vehículo concreto, tratan de eliminar o suavizar este ruido de medición utilizando y aprendiendo a medida que llegan las observaciones de posición. Su resultado es una traza que describe el movimiento realizado.

Existen múltiples técnicas para realizar este procesado, las cuales son expuestas en la sección 2.3 en detalle, al ser el objetivo de la tesis.

En este ejemplo, se compara la estimación realizada por una técnica de estimación (UKF, ver sección 2.3.3.4) sobre la posición en el eje X (UKF), respecto a su posición real (clean) y las observaciones realizadas (noisy), en su evolución sobre el tiempo  $k$ .

Como se observa, si hubiera que utilizar las posiciones observadas, se contaría con grandes saltos de posición claramente erróneos respecto a la realidad del blanco, debido al ruido de la medida.

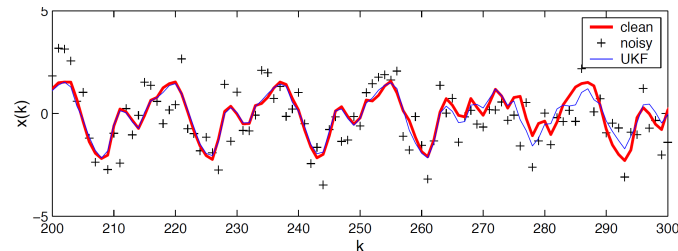


Fig.2.5: Ejemplo resultado algoritmo estimación (UKF) vs posición real vs mediciones [3]

**2.1.2.1.3 Asociación de datos / objetos** Posteriormente (o antes, en función del sistema), si el sistema de fusión cuenta con más de un sensor activo, se realiza el proceso de asociación de las mediciones. Este, trata de unificar trazas de distintos sensores (traza local) que estén siguiendo al mismo objeto, de manera que la unión de las trazas de cada sensor resulte posteriormente en una traza global más fiable. Así, aunque un sensor no funcione correctamente durante un periodo de tiempo, o se rompiese, el

sistema seguiría conociendo la posición y el número de objetivos, en vez de conocer el número de contactos totales, que no representan nada en el escenario real.

Esta etapa es de alta relevancia en un sistema de fusión de datos multisensor, puesto que una mala asociación lacra a todo el sistema (aunque existen mecanismos para revertirlo).

Por ejemplo, en la figura Fig.2.6 se muestra un caso complejo de asociación. En la gráfica b se observan todos los contactos, con su posición y su tiempo, totalmente indistinguibles los vehículos reales que están en el escenario. Sin embargo, como todo el proceso de fusión de nivel 1 funciona correctamente: el tracker etiqueta los contactos a su vehículo correspondiente, cada contacto se estima correctamente y posteriormente se asocian las distintas fuentes entre sí, obtenemos de una nube de contactos, ocho trayectorias separadas y estimadas correctamente.

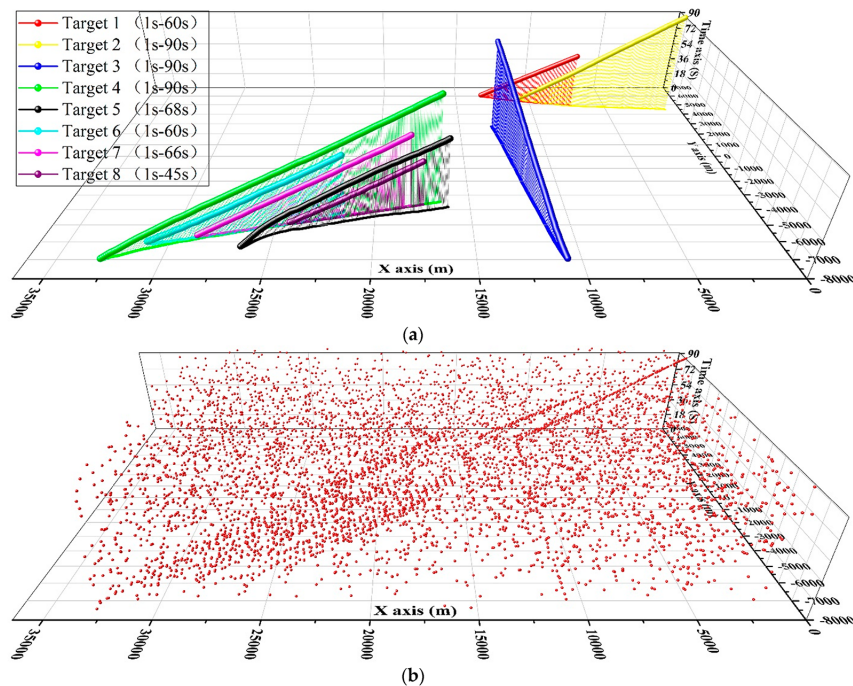


Fig.2.6: Ejemplo contactos reales sin identificador (b) filtrados y asociados en 8 trazas (a) [4]

Las técnicas de asociación de trazas se dividen en dos grupos:

- Técnicas algorítmicas: Su funcionamiento erradica íntegramente el resultado de un algoritmo, sin conocimiento aproximado. Todas las técnicas exponer a continuación,

Para este paso es habitual utilizar la distancia Mahalanobis [20]. Esta distancia estadística pondera la diferencia entre las dos medidas (puede ser



únicamente posición, o incluir más variables, como la velocidad) utilizando una matriz de covarianzas con el ruido de cada variable. De esta manera el valor con el que comparar contempla la confianza de la medida a la hora de realizar un paso muy importante como este.

Las técnicas más conocidas y utilizadas de asociación son:

**Vecino más próximo (NN)** Este grupo de técnicas generan grupos (o clusters) entre todas las trazas locales que existen. De esta manera, cada grupo representa a las trazas más cercanas entre sí, minimizando la distancia total de cada traza a los centroides de los clusters. Estas técnicas son las más simples que se puede desarrollar para asociar las trazas, destacando el algoritmo k-medias [21].

El problema principal es que no proporciona garantías de obtener la solución óptima, al necesitar conocer a priori el número de clusters a realizar. En resumen, le falta contexto para ser un algoritmo de asociación de múltiples contactos.

**Asociación Probabilística de Datos (PDA)** Bar-Shalom y Tse propusieron este famoso algoritmo en 1973 [22] (publicado en 1975) para realizar la asociación.

Su funcionamiento está basado en la regla de Bayes. Inicialmente, se calcula una ventana de probabilidad sobre la traza local en el tiempo  $t$  (podemos definirla como una esfera alrededor de la posición de la traza local que delimita la cercanía o lejanía a la misma de manera probabilística), y sobre los contactos posteriores en tiempo sin asociar a ninguna traza local, se observa si están dentro o no de dicha ventana. Los contactos que estén cerca se considerarán asociables a dicha traza.

Una vez asociado el contacto a la traza, dicha probabilidad calculada es utilizada para ponderar la siguiente actualización de la traza global, es decir, afectar más o menos la posición de este contacto sobre la traza final.

**Asociación Probabilística Conjunta de Datos (JPDA)** Poco tiempo después, los mismos investigadores adaptaron este algoritmo para funcionar para múltiples trazas locales a la vez [23], de manera que en caso de que una medida entre en dos trazas locales, sea capaz de decidir sobre cual asociarla.

**Seguimiento con Hipótesis Múltiples (MHT)** Este algoritmo propuesto por Reid [24] representa otro enfoque para resolver el problema de asociación. Su objetivo es llevar en paralelo todas las posibles combinaciones de asociación desde el instante 0 hasta el instante  $t$ , de manera que en cada

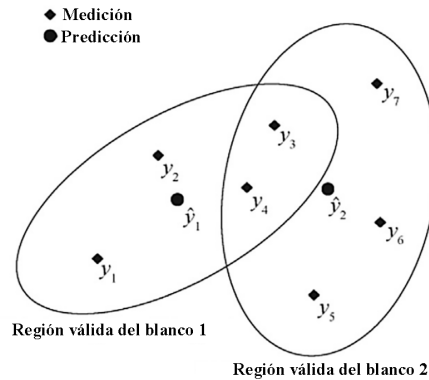


Fig.2.7: Dos blancos con medidas comunes [5]

instante se proporcione siempre la asociación que disminuye el error (o aumenta la verosimilitud).

Su principal problema es la cantidad de posibilidades a calcular, que lo hacen inabarcable en instancias muy tempranas de ejecución. A pesar de ello, existen variantes del algoritmo para hacerlo viable relajando ligeramente el problema.

**Algoritmo de Munkres** Otro algoritmo utilizado para resolver el problema de asociación es el algoritmo de Munkres, también conocido como método húngaro o algoritmo Kuhn-Munkres, al ser ambos los desarrolladores del mismo (primero Kuhn y Harold en 1955 [25] y después Munkres en 1957 [26]).

El algoritmo de Munkres realmente parte de una matriz de asociación. Para generarla en este ámbito, para las trazas globales existentes (formadas por asociaciones de trazas locales) en un sistema de fusión y otras trazas locales sin asignar, se calcula la distancia de cada local sin asignar a cada global, obteniendo como resultado la matriz de asociación. Mediante el algoritmo de Munkres, se resuelve el problema de asociación asignando las trazas de manera que se minimice la distancia mahalanobis total.

**2.1.2.1.4 Identificación del objeto** Por último, a partir de la traza global o de la información recogida por los sensores, tratar de extraer información que identifique el objeto que se está monitorizando. Esta información del contexto del escenario se utiliza en posteriores niveles para sacar conclusiones de un nivel mayor.

Para realizar estas tareas se utiliza todo tipo de técnicas de aprendizaje, para detectar patrones y características concretas. Ejemplos de identificación del objeto pueden ser las Redes de neuronas artificiales (ANN), agrupamientos, Decision Tree (DT), etc. En [27], se utilizan imágenes como entrada de una red neuronal para clasificarla y reconocer el tipo de barco. En [28] se clasifica

el tipo de barco y su tipo de maniobra en función de los parámetros del filtro de estimación.

#### **2.1.2.2 Técnicas nivel 2 y nivel 3**

Con el fin de obtener un contexto que explique el entorno actual, se han hecho algunas propuestas. A pesar de ello, este ámbito no está tan desarrollado como el primer nivel, al carecer de información que evalúe su correcto funcionamiento.

Técnicas de sistemas expertos son comúnmente utilizadas para obtener información de contexto sobre el sistema. Estas técnicas proporcionan una salida comprensible al imitar el razonamiento que realizaría un humano para extraer dichas conclusiones. Por ejemplo, en [29], se describe el uso de ontologías en sistemas de fusión de datos, y concretamente, su utilidad para representar la situación del escenario o de las amenazas de manera comprensible para el operario. En [30] se implementa una ontología sobre un sistema de fusión real basado en cámaras.

Otras, como redes bayesianas [31] o razonamiento borroso utilizan probabilidades para, dado una situación de un barco específico, valorar su estado y obtener su contexto. Su principal ventaja es la capacidad de razonamiento con incertidumbre, lo cual se adapta perfectamente a este problema. Relacionado con estas técnicas tenemos implementaciones basadas en inferencia Dempster-Shafer [32] para cuantificar la confianza de la relación entre dos objetos. Su ventaja respecto al razonamiento bayesiano es que no precisa de ningún conocimiento a priori, como puede ser una distribución de probabilidad.

Las técnicas expuestas también son utilizadas para los procesos del nivel 3. Al ser el objetivo obtener información a más largo plazo, es necesario que dichos procesos cuenten con mecanismos para predecir o extrapolar la posición de los objetos en el futuro. En [33] se realiza este proceso mediante redes neuronales, aunque utilizando los estimadores de posición se puede realizar una extrapolación aceptable.

#### **2.1.2.3 Técnicas nivel 4**

Finalmente, para refinar todos los procesos de fusión, se necesitan extraer los refinamientos que debe realizar cada componente de cada nivel de fusión. Antes de continuar con las técnicas para extraer refinamientos, cabe destacar que los procesos deben ser configurables en tiempo real, para que, una vez extraídos los refinamientos a hacer en el nivel 4, el sistema se pueda ser adaptado. Su problema por resolver lo podemos clasificar como de optimización multiobjetivo, puesto que, con todas las variables a optimizar, existe una configuración que bajo un criterio será la óptima en cada instante. Para ello

debe basarse en Medida del rendimiento (MoP) y Medidas de Efectividad (MoE) de todos los procesos de fusión, tanto de los resultados de los sensores, como de los procesos que infieren conocimiento, para posteriormente decidir los refinamientos.

Actualmente, muchos de los procesos de los niveles anteriores son auto-adaptativos, por lo que su objetivo principal es generar métricas del funcionamiento global del sistema para el humano vía HCI y planificar el tiempo más adecuado para la ejecución de cada proceso de fusión. En [34] se optimiza la precisión de los resultados de fusión utilizando un sistema planificado respecto a un sistema estacionario.

## 2.2 Modelos de los objetivos

Un modelo, en la ciencia, se define como una representación simplificada de la realidad, la cual se puede utilizar como una herramienta para comprender, analizar o simular el funcionamiento de un sistema determinado. Por ello, en el ámbito de este trabajo, el seguimiento de objetivos necesita definir distintos modelos matemáticos que representen correctamente a cada tipo de objetivo como entes sobre el espacio 2D o 3D en el tiempo. Los modelos necesarios para calcular realizar el seguimiento de objetivos son:

- El modelo del proceso (o modelo dinámico) define la evolución del estado del objetivo  $x_t$  en un tiempo  $t$ , a partir del instante de tiempo anterior  $t - 1$ . En este tipo de problemas, define el comportamiento de la cinemática del objetivo a partir de su dinámica y su ruido asociado  $w_t$ . Este ruido representa la diferencia del paso del modelo real al dinámico, puesto que existen aproximaciones o simplificaciones realizadas sobre el problema, como puede ser un ligero viento aleatorio.

$$x_t = f(x_{t-1}, w_{t-1}) \quad (\text{Ec.2.2.1})$$

*Modelo del proceso*

- Los modelos de observación o medición, que definen la forma de relacionar la observación realizada con el estado del objetivo. Estas son funciones  $z_t$  que incorporan a la medida  $x_t$  el ruido asociado que lleva el sensor de la medición  $v_t$ .

$$z_t = h(x_t, v_t) \quad (\text{Ec.2.2.2})$$

*Modelo de observación*

Además, existen otros modelos relacionados, como el cinemático, el cual, a partir de las variables observadas, nos permite obtener el estado. Se obtiene

mediante una operación matricial entre un vector de variables observadas y una matriz jacobiana.

Este modelo permite, por ejemplo, calcular el vector velocidad de un dron si se ejecuta un movimiento de sus rotores concreto. Además, su matriz inversa (o pseudoinversa si no es cuadrada) nos permite hacer el cálculo inverso, es decir, a partir de su velocidad, calcular el movimiento concreto realizado por el dron.

## 2.3 Estimación del estado en sistemas dinámicos

### 2.3.1 Introducción

El estado de un sistema se define como un conjunto de variables representativas (variables de estado), cuyos valores establecen una situación concreta del sistema, que es distinta si los valores difieren. Por otro lado, el sistema será dinámico si los valores de sus variables de estado varían con el paso del tiempo.

Existen casos en los que se cuenta con toda la información y se puede calcular el valor exacto de las variables de estado, pero en muchos sistemas, falta información, ya sea por cantidad o por la calidad de la disponible. En estos casos, es necesario utilizar un estimador, que sea capaz de, utilizando la información disponible, calcular el valor más adecuado para representar el sistema.

Una de las aplicaciones más habituales de los estimadores de estado es el campo de seguimiento de objetivos, al tratar de estimar variables como la posición o velocidad de un vehículo en un instante de tiempo concreto, utilizando en el proceso el valor de otras variables observadas no completamente confiables, al ser medidas con instrumentos de medición precisos, pero no exactos, al estar sometidos a errores de medida, generalmente conocidos. Esto es debido a que para medir las distancias no se puede utilizar un metro tradicional de punto a punto, sino que se utilizan sensores como los explicados en el apartado 2.4.

### 2.3.2 Clasificación

Los estimadores se pueden dividir en función de los dos grandes mundos de la estadística.

#### 2.3.2.1 Inferencia clásica

Por un lado, la parte clásica o frecuentista, donde se entiende que los valores son fijos, pero son desconocidos. Un estimador entonces busca obtener los valores de las variables que maximizan una probabilidad de que suceda.

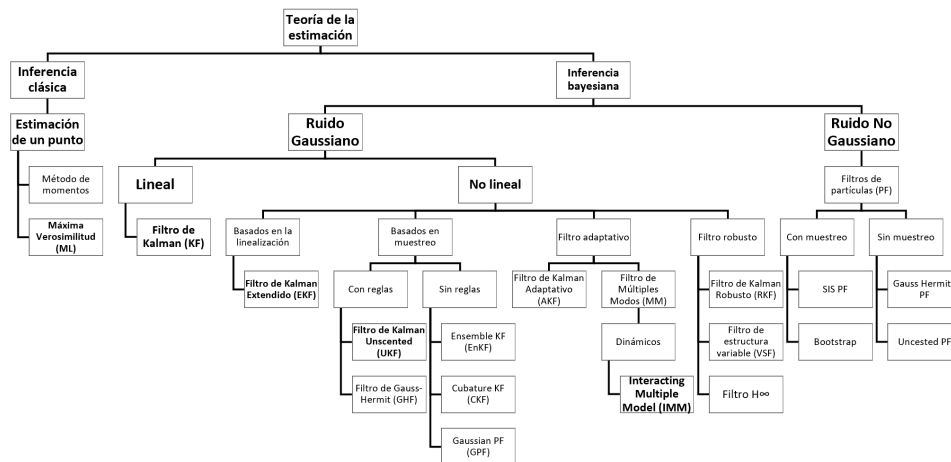


Fig.2.8: Clasificación filtros de estimación

Dentro de esta rama, contamos con los estimadores de un punto, los cuales utilizan inferencia probabilística para representar mediante un único valor (punto) los valores reales de una muestra.

### 2.3.2.2 Inferencia bayesiana

Por el otro lado, la parte bayesiana se fundamenta en el conocimiento de que las variables son aleatorias, regidas por una distribución de probabilidad, la cual es estudiada según el Teorema de Bayes. Esto hace que cualquier posible estimación se realice de manera a priori, utilizando la información disponible hasta dicho momento, pero con el valor real de dicha estimación, se modificaría la información disponible y por tanto la probabilidad de dicha estimación a posteriori.

Esta rama es la más extendida para el seguimiento de objetivos, al adecuarse a la aleatoriedad de la medición de los objetivos. Para su funcionamiento, un estimador bayesiano precisa, al menos, del siguiente conocimiento:

- El modelo del proceso.
- El modelo de la observación.
- Las características asociadas a los ruidos, tanto del proceso como de la observación.
- Un estado inicial de la observación.

**2.3.2.2.1 Asume ruido gaussiano** La siguiente gran distinción es la asunción del uso de una Función Densidad de Probabilidad (PDF) gaussiana,

también conocida como distribución normal o comúnmente como campana de Gauss, para modelar la variable aleatoria. Esta distribución, mostrada en la figura Fig.2.9, es la PDF continua más habitual en cualquier fenómeno real (debido al teorema del límite central) [35] y sus características (media  $\mu$ , moda y mediana centradas) hacen que su aplicación simplifique las ecuaciones de cualquier método de estimación.

Cabe destacar que, en los que utilizan una PDF gaussiana (conocidos como filtros gaussianos), se suele modelar el ruido de la distribución en base a su desviación estándar ( $\sigma$ ). Como se observa en la figura Fig.2.9, un ruido de  $1\sigma$  implica que se consideren como ruidosos valores fuera del 68.27%.

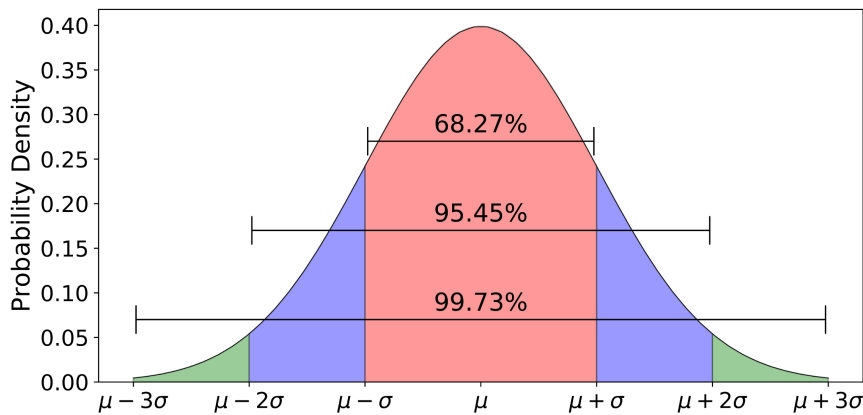


Fig.2.9: Distribución normal [6]

Estos estimadores, a su vez, se dividen según la dinámica del problema que resuelven.

**Problema lineal** Si el problema es lineal, es decir, el modelo dinámico a estimar sigue una ecuación lineal, se cuenta con el KF como solución óptima, que se explica en el apartado 2.3.3.2.

**Problema no lineal** Si el problema no es lineal, el problema aumenta drásticamente de complejidad y no existe ninguna solución óptima para ello. Siguiendo la clasificación de Afshari et al. [31], los estimadores, en función la manera de resolver la no linealidad, se pueden clasificar en distintas categorías:

**Basados en la linealización** El primer caso es realizar una linealización del problema no lineal, de manera que se pueda aplicar el KF original.

**Basados en muestreo** Todos los métodos de esta categoría se apoyan sobre el método de Monte-Carlo. En vez de realizar una linealización para llegar al siguiente instante de tiempo, utilizan una serie de muestras finita (partículas) para aproximar la distribución de probabilidad (en este caso gaussiana) en el próximo instante de tiempo.

Las reglas indican la manera de escoger las muestras a utilizar, si hacerlo de manera informada o aleatoria.

El método de Monte-Carlo fue introducido en la década de los 40 por Ulam, von Neumann y Metropolis [36], permite aproximar resultados matemáticos cuyo cálculo es muy costoso mediante la combinación de los resultados de realizar  $N$  experimentos distintos, variando los parámetros de entrada de manera pseudoaleatoria, para posteriormente agruparlos y obtener un resultado estimado. Desde entonces, este método se volvió muy importante para experimentos en todo tipo de materias.

**Filtros adaptativos** Son capaces de tomar decisiones para ajustarse de mejor manera al problema y obtener mejores resultados. Existen distintas variantes:

- En filtros únicos, se modifican los parámetros variables de configuración del filtro a medida que va realizando el filtrado, tratando de ajustarse.
- En filtros con múltiples modos (siendo un modo un ajuste concreto de un filtro), pueden escoger el mejor modo al comenzar la ejecución (estáticos), o variar en tiempo de ejecución al modo más adecuado (dinámicos).

**Filtros robustos** Finalmente, contamos con los filtros robustos, cuyo principal atractivo es añadir, sobre filtros existentes, una robustez superior a la hora de estimar ruidos o imprecisiones, proporcionando una estimación más adecuada para el problema.

**2.3.2.2 No asume ninguna distribución** Por el contrario, existen problemas donde la variable aleatoria no sigue una distribución de Gauss. En estos casos, puede seguir otra distribución concreta, o directamente ser desconocida o aleatoria la distribución que utiliza. Para estos casos, existen métodos que estiman la posición sin realizar ninguna asunción sobre ella.

Las soluciones más habituales utilizan el concepto del método de Monte Carlo para aproximar la distribución de probabilidad a posteriori, la cual es desconocida en el problema.

Estos filtros se dividen en dos categorías, en función de si en su proceso utilizan una técnica denominada resampling, que busca reutilizar las muestras o partículas del instante anterior (ver sección 2.3.3.7.1.4), o si en cada instante generan de manera completamente nueva todas las partículas.

### 2.3.3 Estimadores principales

A continuación, se presentan y explican los algoritmos más utilizados de estimación y filtrado de estado, haciendo hincapié en su funcionalidad para el seguimiento de objetivos.



### 2.3.3.1 Estimación de máxima verosimilitud (MLE)

Esta técnica, popularizada por Fisher a comienzos del siglo XX, pero empleada previamente por Gauss o Laplace en el siglo XVIII, realiza una estimación basada en probabilidades. El objetivo es obtener, dada una muestra de  $N$  observaciones, cuyos valores se sabe que siguen una determinada distribución mediante el parámetro  $\theta$ , calcular el valor del parámetro  $\theta$  que maximiza la función de verosimilitud (likelihood). Este valor se denomina Estimación de máxima verosimilitud (MLE).

Para la estimación del estado, podemos definir la función de verosimilitud como la función de densidad de probabilidad dadas unas determinadas  $n$  observaciones y el valor del estado real,  $x$ :

$$L(\theta) = p(z|x) \quad (\text{Ec.2.3.1})$$

*Función de verosimilitud*

$$MLE = \operatorname{argmax}_x(L(\theta)) \quad (\text{Ec.2.3.2})$$

*Estimación de máxima verosimilitud*

Cabe destacar que MLE puede no existir para una determinada función. Además, en la práctica, este método se ve limitado al requerir el conocimiento de los modelos de los sensores para conseguir la distribución de las observaciones.

### 2.3.3.2 Filtro de Kalman (KF)

El Filtro de Kalman (KF), también denominado Estimador lineal cuadrático (LQE), surge como mejora del filtro de Wiener (o Wiener-Kolmogorov) en 1960 [37]. Es uno de los principales avances de la teoría de control del último siglo y cuenta con multitud de aplicaciones, destacando en el campo aeroespacial, siendo vital para la misión Apollo 11 [38].

El KF se caracteriza por ofrecer siempre la estimación óptima a partir de toda información que le es proporcionada (medidas, distribución de los ruidos asociados a cada medida, el modelo del sistema, el ruido del sistema, relaciones entre las variables observadas y de estado), devolviendo siempre la estimación que minimiza el Error cuadrático medio (MSE), es decir, el MMSE.

Además, al ser recursivo, no necesita evaluar en cada instante todas las mediciones que ha obtenido, sino que se va ajustando etapa tras etapa, lo cual elimina bastante peso computacional.

Con este filtro, se puede inferir conocimiento de variables que no son directamente medibles, como puede ser la temperatura dentro de un motor,

puesto que se quemaría cualquier instrumento de medida. A partir de las mediciones de temperatura en diversos puntos más alejados y la propagación del calor por los materiales, se puede estimar con bastante precisión este valor.

**2.3.3.2.1 Funcionamiento** Una vez introducidas las variables, pasamos a explicar el algoritmo del filtro. Este, como el resto de filtros bayesianos, funciona mediante un bucle infinito compuesto por tres fases (habitualmente son dos, pero por comprensión del algoritmo base, se explica de esta forma). En la figura Fig.2.10 se muestra el funcionamiento.

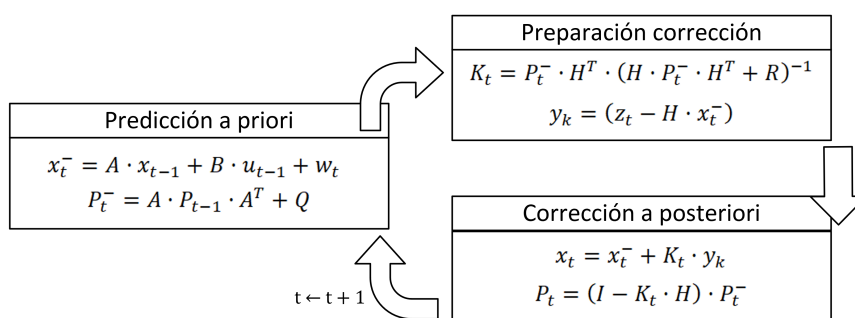


Fig.2.10: Funcionamiento filtro de Kalman

**Fase de predicción** La fase de predicción que genera la predicción a priori  $x_t^-$  a partir de las funciones lineales que describen al modelo,  $f$  y  $h$ . Su funcionamiento utiliza tres ecuaciones:

$$x_t^- = Ax_{t-1} + Bu_{t-1} + w_t \quad (\text{Ec.2.3.3})$$

*KF. Predicción*

En la ecuación Ec.2.3.3 se utiliza el modelo del proceso (ver ecuación Ec.2.2.1) para calcular el vector de estado de manera a priori en el instante de tiempo  $t$  (representado con el superíndice  $x^-$ ). Esto significa que el cálculo se realiza sin apoyarse en observaciones de este instante de tiempo, apoyándose únicamente en los modelos de la última actualización, tanto del proceso ( $x_{t-1}$ ) como de la observación ( $u_{t-1}$ ).

$A$  es una matriz que realiza la transición del estado  $t-1$  al instante de tiempo actual  $t$ .  $B$  es una matriz selecciona qué observaciones se utilizan. Por último,  $w_t$  es un vector que modela el ruido de la medida en esta iteración. Este es modelado por una distribución normal  $N(0, Q)$ .

Cabe destacar que la ecuación tiene una parte del control ( $B, u_{t-1}$ ) se suele eliminar, puesto que no se suele tener conocimiento certero sobre las acciones realizadas sobre el objetivo.

También tenemos la ecuación que calcula la matriz de covarianzas de manera a priori de la estimación del estado, siendo  $P_{t-1}$  la matriz de covarianzas de la iteración anterior y  $Q$  la matriz que modela el ruido del vector de estado (también conocido como ruido de planta), es decir, la incertidumbre sobre la predicción.

$$P_t^- = AP_{t-1}A^T + Q \quad (\text{Ec.2.3.4})$$

*KF. Covarianzas a priori*

**Fase de preparación para la corrección** Una vez terminadas las predicciones, se preparan, con las predicciones, las variables a utilizar en la fase de corrección.

Primero se hace uso de la ecuación Ec.2.3.5 que modela de observación. Esta se compone de la parte correspondiente a las observaciones del vector de estado (para ello se utiliza la matriz  $H$ ) y la observación obtenida en este instante.

$$z_t^- = Hx_t + v_t \quad (\text{Ec.2.3.5})$$

*KF. Ecuación modelo de observación*

Posteriormente, se calculan los valores a utilizar en la corrección. Estos son la ganancia de Kalman  $K_t$  y la innovación  $\tilde{z}_t$ . La ganancia de Kalman indica al modelo si debe apoyarse más en la nueva medida o en el modelo anterior. Su cálculo utiliza la matriz de covarianzas predicha y la matriz  $R$ , que modela el ruido de la medida.

$$K_t = P_t^- H^T S^{-1} \quad (\text{Ec.2.3.6})$$

$$S = HP_t^- H^T + R \quad (\text{Ec.2.3.7})$$

*KF. Ganancia de Kalman*

Por otro lado, la innovación encuentra la diferencia entre la observación y el estado predicho para que el modelo aprenda.

$$\tilde{z}_t = z_t - Hx_t^- \quad (\text{Ec.2.3.8})$$

*KF. Innovación del estado*

**Fase de corrección** Una vez se tienen todos los valores calculados, la fase de corrección genera los valores a posteriori, es decir, a partir de la predicción a priori, utilizar la medida observada para que el modelo aprenda. Tiene dos fórmulas: Primero calculamos el vector de estado a posteriori y final en esta iteración,  $x_t$ . Su cálculo parte de la predicción a priori y se le aplica una corrección. Esta se calcula multiplicando la por la ganancia de Kalman.

$$x_t = x_t^- + K_t \tilde{z}_t \quad (\text{Ec.2.3.9})$$

*KF. Vector de estado a posteriori*

Finalmente, aplicamos el mismo proceso sobre la matriz de covarianzas, aplicando el aprendizaje sobre la estimación a priori.

$$P_t = (I - K_t H) P_t^- \quad (\text{Ec.2.3.10})$$

*KF. Matriz de covarianzas a posteriori*

**2.3.3.2 Conclusiones** El KF exige dos suposiciones sobre el problema, que el modelo del proceso sea lineal y esté definido en forma de ecuación. También necesita el modelo de las observaciones.

Para ambos modelos, necesita que sus distribuciones sean independientes entre sí y sus PDF sean gaussianas. Para problemas con estas características, KF proporciona resultados óptimos de manera estadística al minimizar el MMSE.

Su principal problema es que la mayoría de los problemas en ingeniería cuentan con modelos no lineales, por lo que este filtro cuenta con una usabilidad reducida. A pesar de ello, se han propuesto varias variaciones para adaptarlo a problemas no lineales, como el Filtro de Kalman Extendido, Filtro de Kalman “Unscented” o Filtro de Kalman Ensemble.

**2.3.3.3 Filtro de Kalman Extendido (EKF)**

El Filtro de Kalman Extendido (EKF), como su propio nombre indica, parte de la base creada por Kalman para extender su funcionamiento en problemas no lineales. Fue creado inicialmente por científicos del centro de investigación de la NASA (NASA Ames) [39] [40] [41]. Este algoritmo utiliza la expansión de la serie de Taylor para linealizar el modelo no lineal, de manera que el modelo se adapte y pueda utilizar un KF convencional. Por ello, sigue manteniendo el otro requisito del KF, tener el ruido modelado mediante un proceso gaussiano de media cero.

Además, sus ecuaciones son iguales a las mostradas previamente, excepto los modelos del proceso y (en ocasiones) de observación, que deben ser funciones linealizadas.

**2.3.3.3.1 Funcionamiento** Para resolver esta linealización, se utilizan las matrices jacobianas de cada función, es decir, derivadas parciales de primer orden de la función de cada modelo. Los jacobianos quedan de la siguiente manera (se obvia de la ecuación del jacobiano el tiempo  $t$  para facilitar la comprensión):

$$A_{[i,j]} = \frac{(\delta f_i)}{(\delta x_j)}(x_{t-1}, u_{t-1}, 0) \quad (\text{Ec.2.3.11})$$

*EKF. Jacobiano del modelo de transición de estados*

$$W_{[i,j]} = \frac{(\delta f_i)}{(\delta w_j)}(x_{t-1}, u_{t-1}, 0) \quad (\text{Ec.2.3.12})$$

*EKF. Jacobiano del ruido del modelo del proceso*

$$H_{[i,j]} = \frac{(\delta h_i)}{(\delta x_j)}(x_t^-, 0) \quad (\text{Ec.2.3.13})$$

*EKF. Jacobiano del modelo de espacios observación-estado*

$$V_{[i,j]} = \frac{(\delta h_i)}{(\delta v_j)}(x_t^-, 0) \quad (\text{Ec.2.3.14})$$

*EKF. Jacobiano del ruido del modelo de la observación*

Utilizando los jacobianos y partiendo de las ecuaciones del Filtro de Kalman, se modifican quedando de la siguiente manera:

### Fase de corrección

$$x_t^- = f(x_{t-1}, u_{t-1}, 0) \quad (\text{Ec.2.3.15})$$

*EKF. Predicción*

$$P_t^- = A_t P_{t-1} A_t^T + W_t Q_{t-1} W_t^T \quad (\text{Ec.2.3.16})$$

*EKF. Covarianzas a priori*

$$z_t^- = h(x_{t-1}, 0) \quad (\text{Ec.2.3.17})$$

*EKF. Ecuación modelo de observación*

### Fase de preparación para la corrección

$$K_t = P_t^- H_t^T S^{-1} \quad (\text{Ec.2.3.18})$$

$$S = H_t P_t^- H_t^T + V_t R_t V_t^T \quad (\text{Ec.2.3.19})$$

*EKF. Ganancia de Kalman*

$$\tilde{z}_t = z_t - h(x_t^-, 0) \quad (\text{Ec.2.3.20})$$

*EKF. Innovación del estado*

**Fase de corrección** Una vez se tienen todos los valores calculados, la fase de corrección genera los valores a posteriori, es decir, a partir de la predicción a priori:

$$x_t = x_t^- + K_t \tilde{z}_t \quad (\text{Ec.2.3.21})$$

*EKF. Vector de estado a posteriori*

$$P_t = (I - K_t H_t) P_t^- \quad (\text{Ec.2.3.22})$$

*EKF. Matriz de covarianzas a posteriori*

**2.3.3.3.2 Conclusiones** El EKF basa su funcionamiento en una linealización de primer orden. El problema aparece cuando el modelo del proceso requiera de una linealización de un orden mayor (debido a que los resultados sean erróneos), donde realizar las optimizaciones correspondientes hace muy elevado el coste computacional del filtro.

Por otro lado, aunque se linealice el modelo y pueda dar solución, el filtro deja de ser óptimo como era el KF, al generarse errores al aplicar las derivadas parciales. Además, el cálculo de las derivadas parciales es bastante costoso computacionalmente.

A pesar de ello, es uno de los estimadores no lineales más populares, al obtener buenos resultados en determinados casos, al estar ya muy testeados y por ello su información a priori (matrices  $Q$  y  $R$ ) optimizada.

#### 2.3.3.4 Filtro de Kalman “Unscented” (UKF)

El UKF es propuesto en 1997 por Julier y Uhlmann [42] y perfeccionado por Wan y van der Merwe en el año 2000 [3]. Proponen un filtro que resuelve los principales problemas del EKF, sus malos resultados con modelos altamente no lineales y su alto coste computacional.

Su funcionamiento parte del EKF, pero modificando el método de cálculo de la estimación, pasando de derivadas parciales y jacobianos a la Transformación “Unscented” (UT).

**2.3.3.4.1 Transformación Unscented** Esta transformación UT se basa la intuición de que “es más fácil aproximar la distribución de probabilidad que aproximar una transformación no lineal arbitraria” [43].

Su funcionamiento aproxima la estadística de una variable aleatoria (media y covarianza) tras haber sido sometida a una función no lineal, sin necesitar realizar la operación matemática utilizando jacobianos. Consideremos la función  $z = f(x)$  como una función no lineal donde:

- $x$  es una variable aleatoria con media  $\bar{x}$  y matriz de covarianzas  $P_x$ .

- $z$  es una variable aleatoria con media  $\bar{z}$  y matriz de covarianzas  $P_z$ .

Para obtener la estadística de la variable  $z$  se utiliza la matriz  $\chi$ , compuesta por un conjunto de sigma points. Estos son valores representan la media y covarianza del vector  $x$ , de longitud  $L$ , antes de ser introducidos en la función no lineal. Los  $2L+1$  puntos se obtienen siguiendo las siguientes ecuaciones:

$$\mathcal{X}_0 = \bar{x} \quad (\text{Ec.2.3.23})$$

$$\mathcal{X}_i = \bar{x} + (\sqrt{(L+\lambda)P_x})_i \quad i = 1, \dots, L \quad (\text{Ec.2.3.24})$$

$$\mathcal{X}_i = \bar{x} - (\sqrt{(L+\lambda)P_x})_{i-L} \quad i = L+1, \dots, 2L \quad (\text{Ec.2.3.25})$$

*UT. Cálculo de los sigma points*

El parámetro  $\lambda$  se rige por la siguiente ecuación  $\lambda = \alpha^2(L + \kappa) - L$ , siendo  $\alpha$  una constante que precisa la dispersión de los valores alrededor de la media. Su valor se encuentra entre 1 y  $10^{-4}$ .

El parámetro  $\kappa$  suele utilizar los valores 0 o  $3 - L$ . Ambos se utilizan como parámetros de escala para modelar la covarianza del punto.

Una vez tenemos los sigma points, los propagamos a través de la función no lineal:

$$\mathcal{Z}_i = f(\mathcal{X}_i) \quad i = 0, \dots, 2L \quad (\text{Ec.2.3.26})$$

*UT. Propagación sigma points*

Con estos puntos transformados, que no siguen la PDF gaussiana que tenían en  $\mathcal{X}$ , es necesario aplicarles una aproximación para obtener la media  $\bar{z}$  y su covarianza  $P_z$ .

$$\bar{z} \approx \sum_{i=0}^{2L} W_i^{(m)} \mathcal{Z}_i \quad (\text{Ec.2.3.27})$$

$$P_z \approx \sum_{i=0}^{2L} W_i^{(c)} (\mathcal{Z}_i - \bar{z})(\mathcal{Z}_i - \bar{z})^T \quad (\text{Ec.2.3.28})$$

*UT. Aproximación gaussiana función no lineal*

Siendo los pesos  $W_i^{(m)}$  y  $W_i^{(c)}$  los relativos a la media y la covarianza de cada sigma point. Se calculan con la siguiente ecuación:

$$W_0^{(c)} = \frac{\lambda}{L + \lambda} + (1 - \alpha^2 + \beta) \quad (\text{Ec.2.3.29})$$

$$W_0^{(m)} = \frac{\lambda}{L + \lambda} \quad (\text{Ec.2.3.30})$$

$$W_0^{(c)} = W_0^{(m)} = \frac{1}{2(L + \lambda)} \quad i = 1, \dots, 2L \quad (\text{Ec.2.3.31})$$

*UT. Pesos aproximación gaussiana*

$\beta$  es un parámetro que permite incorporar conocimiento a priori de la distribución que sigue  $x$ . Al ser gaussiana, se recomienda utilizar  $\beta = 2$

**2.3.3.4.2 Funcionamiento** Una vez entendida la Transformación “Unscented” (UT), se procede a explicar el funcionamiento del filtro de estimación. Este filtro se compone de las dos fases habituales de un KF. Predicción del siguiente instante, y después corrección del mismo con las medidas.

**Fase de predicción** Antes de comenzar, hay que destacar que el vector de estado debe ser modificado, para incluir las dos variables aleatorias y permitir su propagación al siguiente instante de tiempo vía Transformación “Unscented” (UT). Sigue estando presente el estado del proceso, pero se incluye el ruido del proceso  $w$  y el de la medida  $v$ . Por ende, los sigma points se componen de la misma estructura.

$$x = [x^T w^T v^T]^T \quad (\text{Ec.2.3.32})$$

$$\mathcal{X} = [(\mathcal{X}^x)^T (\mathcal{X}^w)^T (\mathcal{X}^v)^T]^T \quad (\text{Ec.2.3.33})$$

*UKF. Vector de estado*

La predicción comienza calculando los estadísticos del vector de estado a utilizar en la transformación UT. Debido al tipo de problema que tenemos, con probabilidades, no se puede utilizar la media aritmética, sino que hay que utilizar la esperanza matemática. Este estadístico una variable aleatoria representa el valor medio de un fenómeno aleatorio, teniendo en cuenta las probabilidades de dicho fenómeno. EJ: Esperanza de un dado de 6 caras, 3.5.

$$\hat{x}_{t-1} = E[x_{t-1}] \quad (\text{Ec.2.3.34})$$

$$P_{t-1} = E[(x_{t-1} - \hat{x}_{t-1})(x_{t-1} - \hat{x}_{t-1})^T] \quad (\text{Ec.2.3.35})$$

*UKF. Estadísticos iniciales*

A continuación se calculan los puntos sigma:

$$\mathcal{X}_{t-1} = \hat{x}_{t-1} \hat{x}_{t-1} \pm \sqrt{(L + \lambda)P_{t-1}} \quad (\text{Ec.2.3.36})$$

*UKF. Cálculo sigma points*

Al igual que en la UT, se propagan los sigma points mediante la función no lineal:

$$\mathcal{X}_{t|t-1}^x = f(\mathcal{X}_{t|t-1}^x, \mathcal{X}_{t|t-1}^w) \quad (\text{Ec.2.3.37})$$

*UKF. Propagación sigma points*



Una vez propagados los puntos, se utilizan junto a los pesos (ver Ec.2.3.29) para realizar la predicción del vector de estado y de la matriz de covarianza:

$$x_t^- = \sum_{i=0}^{2L} W_i^{(m)} \mathcal{X}_{i,t|t-1} \quad (\text{Ec.2.3.38})$$

$$P_t^- = \sum_{i=0}^{2L} W_i^{(c)} (\mathcal{X}_{i,t|t-1} - x_t^-) (\mathcal{X}_{i,t|t-1} - x_t^-)^T + Q \quad (\text{Ec.2.3.39})$$

*UKF. Predicción vector de estado y covarianzas*

**Fase de corrección** Una vez se ha predicho el estado, es necesario recalcular los sigma points con el vector de estado predicho, para que incorpore el efecto de la matriz Q sobre los puntos.

$$\mathcal{X}_{t|t-1} = x_t^- x_t^- \pm \sqrt{(L + \lambda) P_t^-} \quad (\text{Ec.2.3.40})$$

*UKF. Recálculo puntos sigma*

A continuación, se coloca la predicción en el formato de la observación para compararla con la real. Para ello se somete a la predicción a la función de observación y se obtiene su estado.

$$\mathcal{Z}_{t|t-1} = h(\mathcal{X}_{t|t-1}^x, \mathcal{X}_{t|t-1}^v) \quad (\text{Ec.2.3.41})$$

$$z_t^- = \sum_{i=0}^{2L} W_i^{(m)} \mathcal{Z}_{i,t|t-1} \quad (\text{Ec.2.3.42})$$

*UKF. Predicción de la observación*

Finalmente, generamos la matriz de covarianzas de la observación  $P_{y_t y_t}$  y la matriz de covarianzas cruzada entre el vector de estado y el de observación  $P_{x_t y_t}$ .

$$P_{z_t z_t} = \sum_{i=0}^{2L} W_i^{(c)} (\mathcal{Z}_{i,t|t-1} - z_t^-) (\mathcal{Z}_{i,t|t-1} - z_t^-)^T + R \quad (\text{Ec.2.3.43})$$

$$P_{x_t z_t} = \sum_{i=0}^{2L} W_i^{(c)} (\mathcal{X}_{i,t|t-1} - x_t^-) (\mathcal{Z}_{i,t|t-1} - z_t^-)^T \quad (\text{Ec.2.3.44})$$

$$K_t = P_{z_t z_t} (P_{x_t z_t})^{-1} \quad (\text{Ec.2.3.45})$$

*UKF. Ganancia de Kalman*

Por último, una vez se obtiene la medida  $z_t$ , se calcula la innovación  $\tilde{z}_t$  para aplicar la corrección sobre la predicción, obteniendo la posición final estimada y terminando la ejecución de esta iteración del filtro.

$$\tilde{z}_t = z_t - z_t^- \quad (\text{Ec.2.3.46})$$

$$x_t = x_t^- + K_t \tilde{z}_t \quad (\text{Ec.2.3.47})$$

$$P_t = P_t^- - K_t P_{z_t z_t} K_t^T \quad (\text{Ec.2.3.48})$$

**2.3.3.4.3 Conclusiones** Se puede concluir que el filtro UKF ofrece mejor precisión en la estimación en todo tipo de escenarios respecto al EKF, con una complejidad computacional similar. Su implementación es mucho más sencilla, al dejar de lado los cálculos de jacobianos. Aun así, sigue contando con algunas suposiciones, como la necesidad de una PDF gaussiana, aunque se puede modificar para eliminarla, con el Filtro de Partículas “Unscented” (UPF).

Existen variaciones sobre este filtro, como el Filtro de Gauss-Hermite (GHF) o el Filtro de Kalman "Cubature" (CKF). Estas se pueden resumir como ligeras variaciones del UKF (los parámetros de configuración, etc.) y proporcionan buenos resultados en distintos tipos de problemas.

### 2.3.3.5 Filtro de Kalman Ensemble (EnKF)

Este filtro, desarrollado por Evensen [44] [45], funciona como alternativa al EKF, aplicando el KF sobre un conjunto (ensemble) de muestras aleatorias dadas a partir del estado actual. Para generar dicho conjunto utiliza el conocido método de Monte-Carlo. A diferencia del UKF que propaga un número fijo de sigma points, aquí se generan tantas muestras como sean posibles (debido a la carga computacional), de manera totalmente aleatoria, para propagarlas al siguiente instante de tiempo.

Su resultado conlleva varias distribuciones gaussianas con distintas medias y covarianzas, que son unificadas para dar lugar a la distribución final y al vector de estado de dicha iteración.

**2.3.3.5.1 Funcionamiento** Como en todas las versiones del KF, este filtro se compone de dos fases principales.

**Fase de predicción** El primer paso es generar el conjunto de muestras a utilizar. Para generar N muestras, se utilizan el error de la observación  $R$  y la estimación en el instante  $t$ ,  $x_t$ . Existen distintos métodos para generar el conjunto aleatorio.

A partir del siguiente conjunto:  $X_{t-1} = [x_{t-1}^1, \dots, x_{t-1}^i \ \forall i = 1, \dots, n]$  se realiza la propagación de todas las muestras utilizando la función no lineal.

$$X_t^- = [x_t^{i-} \ \forall i = 1, \dots, n] = f(x_{t-1}^i, w_t) \quad (\text{Ec.2.3.49})$$

*EnKF. Predicción muestras*

Una vez propagadas, se puede obtener la predicción creada por todas las muestras mediante la siguiente fórmula:

$$x_t^- = \frac{1}{n} \sum_{i=1}^n x_t^{i-} \quad (\text{Ec.2.3.50})$$

*EnKF. Predicción*

**Fase de corrección** Con la predicción de todas las muestras se realiza la ecuación de corrección, la cual . Primero es necesario calcular la ganancia de Kalman. En este caso, no se cuenta con una matriz  $Q$  del ruido del vector, al ser el propio vector calculado a partir de un conjunto de muestras. Para solventarlo, se utiliza la denominada matriz de covarianzas "ensemble"  $C$ .

$$C = \frac{1}{n} X_t^- (X_t^-)^{-1} \quad (\text{Ec.2.3.51})$$

*EnKF. Matriz de covarianzas "ensemble"*

$$K_t = CH^T (HCH^T + R)^{-1} \quad (\text{Ec.2.3.52})$$

*EnKF. Ganancia de Kalman*

También es necesario ajustar la observación para que en la ecuación de corrección corrija a todas las medidas. Para ello, se genera un vector que contiene, de manera repetida, la observación, variando únicamente el ruido de observación.  $Z_t = [z_t^i]$ , siendo  $z_t^i = z_t + v_t^i$

A continuación, tras preparar toda la información, se puede aplicar la ecuación de corrección para obtener los estados en el instante  $t$  de cada una de las muestras:

$$X_t = X_t^- + K_t(Z_t - HX_t^-) \quad (\text{Ec.2.3.53})$$

*EnKF. Ecuación de corrección*

Finalmente, se calcula el valor del vector de estado resultante de la unión de todas las muestras corregidas.

$$x_t = \frac{1}{n} \sum X_t = \frac{1}{n} \sum_{i=1}^n x_t^i \quad (\text{Ec.2.3.54})$$

*EnKF. Ecuación de corrección*

**2.3.3.5.2 Conclusiones** Esta solución ofrece una gran estimación cuando la cantidad de muestras es elevada. Su principal problema es este, que su rendimiento depende de manera directa del número de muestras, y de este depende el coste computacional. Por ello, todas las técnicas basadas en Monte-Carlo están muy ligadas a las técnicas de computación de altas prestaciones para maximizar el número de muestras.

Por otro lado, este filtro es muy similar al Filtro de Kalman Monte-Carlo

(MCKF). Concretamente, su única diferencia es que MCKF propaga la media y la varianza en la fase de actualización [46].

Para ilustrar el funcionamiento de los filtros vistos hasta el momento, en la figura Fig.2.11 se resume el funcionamiento de los tres últimos filtros, observando sus diferencias a la hora de calcular el siguiente instante de tiempo.

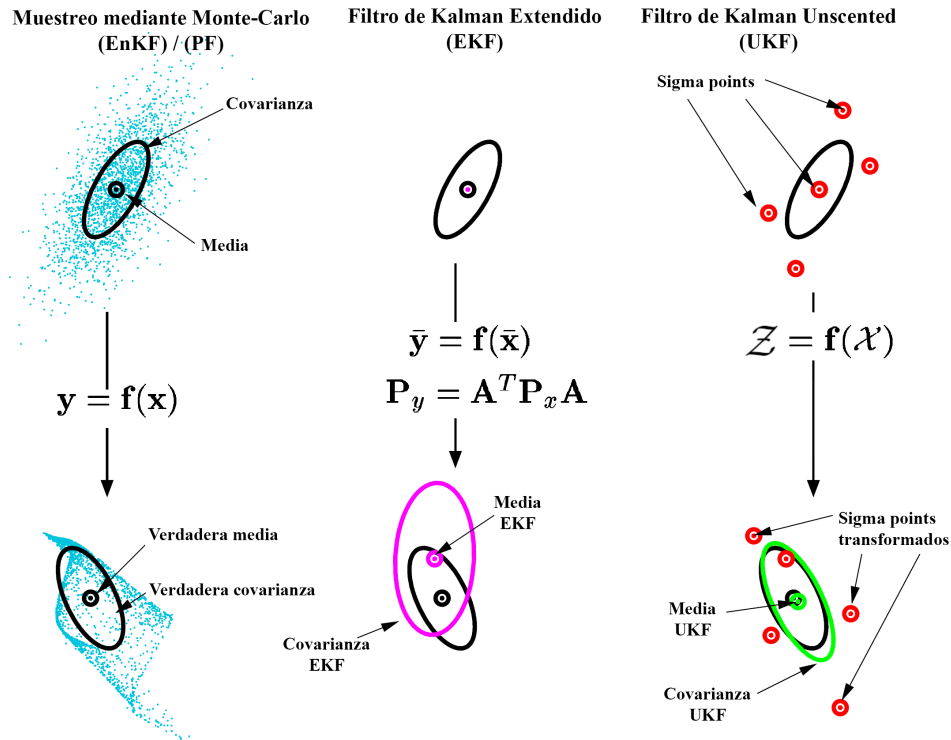


Fig.2.11: Comparativa EKF, UKF, EnKF/PF [3]

### 2.3.3.6 Filtro de Múltiples modos interoperantes (IMM)

Propuesto por primera vez en 1985 [47], el filtro IMM se ha convertido en uno de los filtros más populares para realizar seguimiento de objetivo con alta maniobrabilidad y seguimiento de múltiples objetivos. Esto es debido al funcionamiento cooperativo entre un número finito de modos de seguimiento para combinar sus estimaciones de posición y obtener una estimación más ajustada.

Cada modo se compone de un filtro de seguimiento que abarca un conjunto de tipos de movimientos que puede realizar el objetivo. De esta manera, y en función de los contactos de los sensores, el filtro IMM detecta la impor-

tancia de cada modo en cada momento de la trayectoria y se adapta a ello, proporcionándole peso concreto en la estimación final. Esta transición entre estados es modelada mediante una cadena de Markov.

Para su funcionamiento se puede utilizar cualquier tipo de filtro mientras sea bayesiano.

**2.3.3.6.1 Funcionamiento** Como siempre, partimos de un instante de tiempo  $t$ , con un vector de estado concreto  $x_t$ , una matriz de covarianza  $P_t$  y varias variables para gestionar los  $N$  modos.

La matriz de transiciones  $\Pi$ , cuyo conocimiento debe conocerse a priori, es una matriz cuadrada de dimensiones  $N \times N$  que representa para cada celda, la probabilidad de que el blanco esté en el modo  $i$  y transicione al modo  $j$   $p_t^{ij}$ . Sus valores pueden variar a medida que avanza la ejecución.

$$\Pi = \begin{pmatrix} p_t^{11} & \dots & p_t^{1N} \\ \dots & \dots & \dots \\ p_t^{N1} & \dots & p_t^{NN} \end{pmatrix} \quad (\text{Ec.2.3.55})$$

*IMM. Matriz de transición*

Para cada modo, el filtro IMM utiliza un filtro de seguimiento (puede repetirse entre modos) que representa un modo de movimiento concreto (movimiento a velocidad constante, aceleración constante, etc.). Estos funcionan de manera totalmente independiente al filtro IMM. La única información que comparten es su vector de estado  $x_t^i$ , su matriz de covarianzas  $P_t^i$  iniciales en cada actualización.

Además, el filtro IMM tiene  $N$  probabilidades  $\mu_t^i$  que representan la probabilidad de que el objetivo en un instante de tiempo  $t$  esté operando en el modo  $i$ . Por tanto, en dicho instante, la suma de probabilidades de todos los modos para ese blanco será 1.

Un filtro IMM se compone de cuatro fases.

**Interacción / Mezclado** En esta fase se mezclan las estimaciones de los distintos modos, teniendo en cuenta la probabilidad predicha de cada modo. Esta representa la probabilidad de que el blanco se mueva conforme al modo  $j$  en el instante  $t$ . Para ello, se suman las probabilidades de, desde cualquier modo  $i$  en el instante  $t - 1$ , pasar al modo  $j$ .

$$\mu_t^{j-} = \sum_{i=1}^N \mu_{t-1}^i p_{t-1}^{ij} \quad (\text{Ec.2.3.56})$$

*IMM. Probabilidad predicha*

Con esta probabilidad, calculamos la probabilidad de mezclado, es decir, la probabilidad de que el blanco estuviera en el modo  $i$  en la actualización anterior, dada la hipótesis de que el modo actual es el  $j$ .

$$\mu_t^{i|j} = \frac{p_{t-1}^{ij} \mu_{t-1}^{i|j}}{\mu_t^{j-}} \quad (\text{Ec.2.3.57})$$

*IMM. Probabilidad de mezclado*

Con estas dos probabilidades, ya podemos calcular el vector de estado  $x_{t-1}^{0j}$  y matriz de covarianzas  $P_{t-1}^{0j}$  a utilizar como entrada en el filtro de cada modo. Para esta última, se utiliza un término de corrección de la desviación de cada modo  $X_{t-1}^j$ .

$$x_{t-1}^{0j} = \sum_{i=1}^N \mu_{t-1}^{i|j} x_{t-1}^i \quad (\text{Ec.2.3.58})$$

*IMM. Entrada vector de estado cada modo*

$$P_{t-1}^{0j} = \sum_{i=1}^N \mu_{t-1}^{i|j} P_{t-1}^i + X_{t-1}^j \quad (\text{Ec.2.3.59})$$

$$X_{t-1}^j = \sum_{i=1}^N \mu_t^{i|j} (x_{t-1}^i x_{t-1}^{0j}) (x_{t-1}^i x_{t-1}^{0j})^T \quad (\text{Ec.2.3.60})$$

*IMM. Entrada matriz de covarianzas cada modo*

**Predicción** Se utilizan los filtros de cada modo para obtener su estimación de vector de estado y matriz de covarianzas, a partir de la estimación mezclada en el apartado anterior. Esta fase incluye la corrección que realice cada filtro con la observación.

**Actualización de los modelos** Una vez los filtros de cada modo han terminado, se calcula la verosimilitud del residuo de cada modo  $\Lambda_t^i$ . Es decir, la innovación del error.

$$\Lambda_t^i = \left| S_t^i \right|^{1/2} \exp\left(-\frac{1}{2} v_t^i (S_t^i)^{-1} v_t^i\right) \quad i = 1, \dots, N \quad (\text{Ec.2.3.61})$$

*IMM. Verosimilitud*

Donde  $\tilde{z}_t$  es la innovación de la medida (residuo) y  $S_t^i$  la covarianza del residuo.

También se realiza la actualización de las probabilidades de modo, una vez tenemos la estimación de cada modo y podemos observar su comportamiento. Para ello, a partir de la regla de Bayes con las probabilidades a priori y las verosimilitudes, se obtiene la siguiente ecuación.

$$\mu_t^i = \frac{\Lambda_t^i \mu_t^{j-}}{\sum_{i=1}^N \Lambda_t^i \mu_t^{j-}} \quad i = 1, \dots, N \quad (\text{Ec.2.3.62})$$

*IMM. Actualizada probabilidad modo*

**Combinación** Una vez terminada la iteración, con las probabilidades actualizadas y las estimaciones de cada modo, se combinan los resultados de todos los modos de manera ponderada en la estimación y matriz de covarianzas final.

$$x_t = \sum_{i=1}^N \mu_t^i x_t^i \quad (\text{Ec.2.3.63})$$

*IMM. Vector de estado final*

$$P_t = \sum_{i=1}^N \mu_t^i P_t^i + X \quad (\text{Ec.2.3.64})$$

$$X = \sum_{i=1}^N \mu_t^i (x_t^i x_t^i)^T \quad (\text{Ec.2.3.65})$$

*IMM. Matriz de covarianzas final*

**2.3.3.6.2 Conclusiones** El filtro IMM, así como el resto de filtros de múltiples modos o variaciones de éste, son grandes alternativas para el filtrado de blancos de alta maniobrabilidad [48], al permitir una rápida respuesta a cualquier cambio con el salto a un modo específico para el mismo, incluso con un filtro diferente.

A pesar de ello, como el resto de filtros actuales, no es una buena solución cuando la densidad de blancos es elevada.

### 2.3.3.7 Filtro de partículas (PF)

Esta rama de filtros de estimación, busca mejorar el rendimiento del EKF. La implementación más simple de un algoritmo PF es la primera que se hizo. Esta técnica es conocida como el filtro Bootstrap, desarrollado por

Gordon [49]. También se conoce como Muestreo Secuencial Ponderado con Selección (SIR). A pesar de ello, no fue hasta 1996 que se les denominó como Filtro de partículas (PF) [50].

Estas soluciones proponen aproximar la Función Densidad de Probabilidad (PDF) del estado en cualquier tipo de problema, ya sea lineal, donde se conozca que la PDF será lineal, o en un sistema no lineal cuya distribución es desconocida. Esta es la principal diferencia con los filtros de Kalman, puesto que no depende una distribución gaussiana, sino que es capaz de estimar cualquier PDF.

Su funcionamiento, a grandes rasgos, se observa en la figura Fig.2.12. Basándose en el método de Monte-Carlo, se generan  $N$  partículas aleatorias cuyo objetivo es representar la PDF de la siguiente iteración, de manera que, una vez conocida, se pueda calcular el estado de dicha iteración. Esto hace que el resultado de las variables aleatorias sea de vital importancia. Por ello, se trata de maximizar el número de estas partículas utilizando técnicas como la computación de altas prestaciones, con el fin de abarcar un mayor espacio de búsqueda y obtener mejores resultados. Idealmente, si se tuvieran infinitas partículas, la estimación sería óptima.

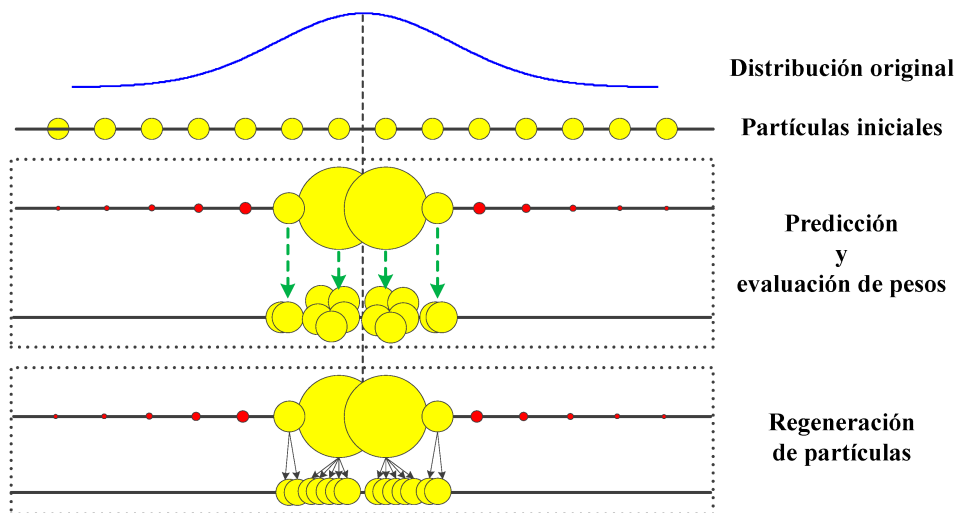


Fig.2.12: Funcionamiento filtro de partículas [7]

**2.3.3.7.1 Funcionamiento** El filtro funciona a partir de las  $N$  partículas que muestrean el espacio de búsqueda (cada partícula es un posible estado) y sus  $N$  pesos asociados (la probabilidad de certidumbre de esa muestra dentro de la PDF global). El algoritmo cuenta con cuatro fases.



**Inicialización** El primer paso, que se realiza únicamente una vez, consiste en poner las  $N$  partículas, basándose en el vector de estado inicial y modificando su incertidumbre, sobre el espacio del estado. Su incertidumbre puede realizarse de manera completamente aleatoria, o siguiendo algún tipo de patrón, como una posición inicial aproximada.

El peso de cada partícula en este instante es ecuánime,  $\frac{1}{N}$ .

El bucle, repetido para cada nueva medición observada, es el siguiente:

**Predicción** Se estima la posición en el siguiente instante de tiempo introduciendo cada partícula en el modelo del proceso.

$$x_t^{i-} = f(x_{t-1}^i) + w_t^i \quad (\text{Ec.2.3.66})$$

*PF. Predicción cada partícula*

**Actualización de pesos** Utilizando las partículas y sus pesos asociados de la iteración anterior  $w_{t-1}^i$ , se calcula su nuevo peso en función a su semejanza sobre el estado obtenido por la observación  $z$ . También se recalcula el sumatorio de peso  $W$ .

$$w_t^i = \frac{\exp(-\frac{1}{2}(z_t - h_t(x_t^{i-}))^{-1}R_t^{-1}(z_t - h_t(x_t^{i-})))}{\sqrt{(2\pi)^m |R_t|}} \quad (\text{Ec.2.3.67})$$

$$W_t = \sum_{i=1}^N w_{t-1}^i \quad (\text{Ec.2.3.68})$$

*PF. Actualización*

Dicho peso debe ser normalizado. Para ello, se calcula cada peso normalizado  $\alpha$

$$\alpha_t = \frac{w_t^i}{W_t} \quad (\text{Ec.2.3.69})$$

*PF. Actualización*

**Remuestreo** Con el fin de mantener la información de las mejores muestras del instante anterior, una vez ya calculada la a posteriori, se utiliza el remuestreo (resampling) para descartar la mayoría de las muestras que tengan bajo peso y potenciar la exploración de las partículas que tienen un alto peso. De manera, se abarca mejor el espacio de la distribución de probabilidad desconocida.

Existen varios métodos para este paso, siendo uno de ellos el conocido como SIR. El inicialmente propuesto está basado en probabilidades de esperanza, de manera que a mayor peso normalizado, mayor esperanza de obtener más partículas en la siguiente iteración basadas en dicha partícula  $i$ .

$$E(n_t^i) = N\alpha_t^i \quad (\text{Ec.2.3.70})$$

#### *PF. Resampling SIR*

Cabe destacar que es importante añadir ruido a las nuevas partículas, ya que sino se terminaría convergiendo a un único punto.

**2.3.3.7.2 Conclusiones** Esta rama de filtros se ha convertido en una solución muy popular los últimos años, con multitud de bibliografía aplicada a casos muy diversos. Esto es debido a la mejora computacional de los sistemas, permitiendo al filtro emitir una cantidad de partículas muy elevada y por ende obteniendo resultados bastante más acordes a un EKF en situaciones cuya PDF es desconocida.

Uno de sus usos más comunes es la visión por computador (expuesta en la sección 2.4.1), al ser muy sencilla la implementación visual de las partículas sobre la imagen 2D.

### **2.3.4 Métricas de funcionamiento**

Una vez aplicado un estimador de posición, es primordial conocer si su resultado es aceptable bajo una serie de condiciones. Estas pueden ser respecto a la precisión de la estimación realizada, a su capacidad de funcionamiento en tiempo real (también denominado como on-line), o la consistencia de los resultados estimados.

Cabe destacar que, en función del escenario, la existencia de ground truth (conocimiento real y preciso del vector de estado), se pueden utilizar unas métricas de rendimiento (MOP) u otras.

- Si se cuenta con ground truth, se puede utilizar el error de la estimación del filtro respecto al ground truth:  $\tilde{X}_t = x_t - x_t^-$
- Si no se cuenta con ground truth, se utiliza la innovación del filtro a partir de la medida.  $\tilde{z}_t = z_t - Hx_t^-$

#### **2.3.4.1 Precisión**

Para medir la precisión del filtro, se utiliza la Raíz del error cuadrático medio (RMSE): Esta mide la diferencia entre las predicciones realizadas y el estado real del blanco. Se suele realizar por variables o grupos de ellas, como por ejemplo posición (suma de X e Y), o velocidad. En caso de no contar con ground truth, se pueden realizar M iteraciones mediante el método de

Monte-Carlo para aproximarlos.

$$RMSE_t^{Var} = \sqrt{\frac{1}{N} \sum_{i=1}^N \tilde{X}_t^i(Var)} \quad (\text{Ec.2.3.71})$$

*Métricas. RMSE con vector de estado*

En el caso de no contar con ground truth, se modifica la fórmula ligeramente utilizando la innovación de cada parámetro de la observación. En estos escenarios, la posición x e y. Habitualmente, con casos reales sin ground truth, al no ser una simulación, únicamente se tiene una medición, luego  $N = 1$

$$RMSE_t^{Var} = \sqrt{\frac{1}{N} \sum_{i=1}^N \tilde{z}_t^i(Var)} \quad (\text{Ec.2.3.72})$$

*Métricas. RMSE con innovación*

También se utiliza la Posterior Cramer-Rao Low Bound (PCRLB) como indicador de precisión, mostrando un límite inferior sobre el MSE de la estimación. Su funcionamiento utiliza la inversa de la Matriz de información de Fisher (FIM) para proporcionar un límite inferior en la varianza del estimador utilizado.

Para ello, se establece la hipótesis de ser consistente si el valor se encuentra dentro de los límites superior  $s(\alpha, N)$  e inferior  $i(\alpha, N)$  delimitados por el parámetro  $\alpha$  (ratio de error).

$$J_t = D_{t-1}^{22} - D_{t-1}^{21}(J_{t-1} + D_{t-1}^{11})^{-1}D_{t-1}^{12} \quad (\text{Ec.2.3.73})$$

$$J_0 = Q_0^{-1} \quad (\text{Ec.2.3.74})$$

*Métricas. Matriz de información de Fisher*

$$D_{t-1}^{11} = A_{t-1}^T Q_t - 1^{-1} A_{t-1} \quad (\text{Ec.2.3.75})$$

$$D_{t-1}^{12} = D_t - 1^{21} = -A_{t-1}^T Q_t - 1^{-1} \quad (\text{Ec.2.3.76})$$

$$D_{t-1}^{22} = Q_t - 1^{-1} + H_t^T R_t^{-1} H_t \quad (\text{Ec.2.3.77})$$

*Métricas. Valores para calcular la FIM*

### 2.3.4.2 Consistencia

Por último, se mide la consistencia del filtro. Esto quiere decir que los valores estimados se encuentran dentro de un margen de error razonable.

Para ello, se utilizan dos estadísticos, en función de la disponibilidad de ground truth:

Primero se muestra el Error cuadrático de estimación normalizado (NEES): idealmente se aplica cuando se cuenta con ground truth, pero en la práctica, solo experimentos muy concretos permiten medir las variables del estado con la suficiente precisión. Por ello, se suele utilizar NEES aproximando el valor real mediante simulaciones Monte-Carlo a posteriori, con la siguiente ecuación.

$$NEES_t = \frac{1}{N} \sum_{i=1}^N (\tilde{X}_t^T P_t^{-1} \tilde{X}_t) \quad (\text{Ec.2.3.78})$$

*Métricas. NEES*

El otro estadístico es el Innovación cuadrática normalizada (NIS). Esta otra métrica se puede utilizar también de manera a posteriori con simulaciones Monte-Carlo, pero también, al utilizar las mediciones, se puede comprobar en tiempo real, con  $N=1$  y cada única observación  $z_t$  en el tiempo.

$$NIS_t = \frac{1}{N} \sum_{i=1}^N (\tilde{z}_t^T S_t^{-1} \tilde{z}_t) \quad (\text{Ec.2.3.79})$$

*Métricas. NIS*

Con los valores hallados, se establece la hipótesis de que el filtro utilizado es consistente y su PDF gaussiana. Con estas condiciones, el valor NIS/NEES es una PDF chi-cuadrado que cuenta con  $V$  grados de libertad, siendo  $V$  la dimensión del vector de la observación (EJ: Posición XY,  $V = 2$ ). En este caso, si el valor NIS (promedio) es inferior a  $V$ , se considera que el filtro es consistente.

### 2.3.4.3 Tiempo de ejecución

Por último, para comprobar que el filtro pueda funcionar en tiempo real, el proceso completo del filtro debe tardar menos tiempo que el periodo de refresco habitual de los sensores.

Para comprobarlo, se suele medir el tiempo de computación del mismo vía software.

## 2.4 Fuentes de información

Para realizar cualquier estimación de posición, primero necesitamos encontrar las medidas o fuentes de datos de donde extraer la posición ruidosa. Para ello, en el entorno marítimo, se utilizan las siguientes tecnologías:

### 2.4.1 Visión por computador

Computer vision, o machine vision, es un campo de la inteligencia artificial centrado en obtener información de imágenes captadas por cámaras. Sobre estas imágenes del mundo real, mediante técnicas de IA se pueden detectar los objetos de una escena, procesarlos y extraer información de ellos.

Uno de los temas más investigados de esta rama de la IA busca utilizar las imágenes captadas para desarrollar sistemas de vigilancia que detecten objetos o comportamientos concretos, para, en caso de detectarlo, alertar sobre dicho suceso [51]. En el ámbito marítimo es una técnica habitual, como se ve en la figura Fig.2.13, para monitorizar el entorno de cualquier barco o costa [52], observando el horizonte y detectando cualquier figura que se observe sobre él.

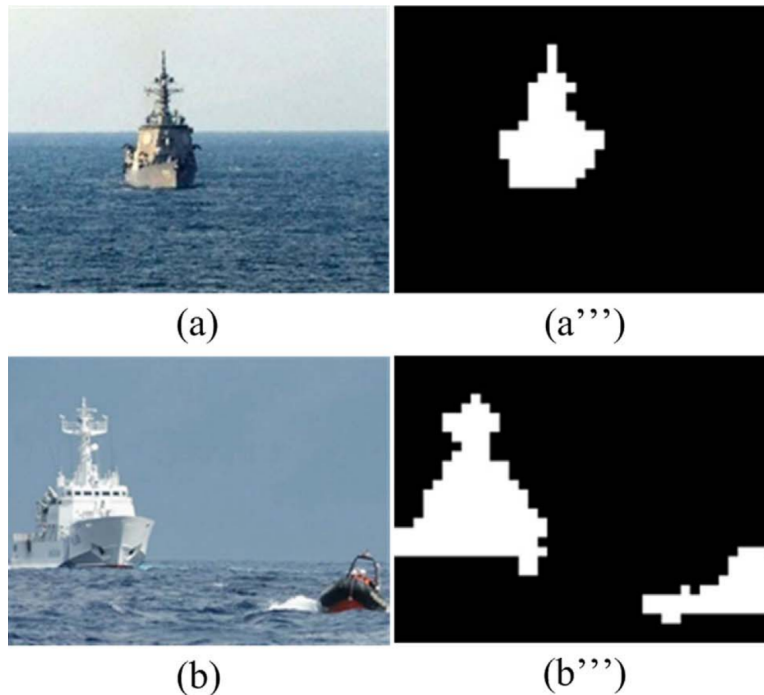


Fig.2.13: Detección de barcos mediante visión por computador [8]

Además, se puede ir un paso más allá y realizar, con las detecciones de los vehículos en el tiempo y el movimiento del propio buque (si está en movimiento), inferir su posición y rumbo y más conocimiento. A pesar de ello, aún quedan varios retos que resolver en entornos marítimos [53]. Aparte de las cámaras de vídeo habituales, existen varias implementaciones dedicadas para la monitorización del entorno marítimo.

#### 2.4.1.1 Sistema de sensores optrónicos (OSS)

Con el fin de hacer más robustos los sistemas de visión por ordenador, se han desarrollado estos sensores, principalmente para buques de guerra. Un OSS cuenta con diversos sensores de visión que complementan la cámara convencional con, por ejemplo, cámaras infrarrojas o térmicas. Además, cuentan con la Inteligencia Artificial (AI) integrada para reconocer y detectar objetivos. En la figura Fig.2.14 observamos un sistema OSS, de la marca Chess-Dynamics, instalado en un barco.



Fig.2.14: Sistema OSS de la marca Chess-Dynamics

#### 2.4.1.2 Radar de apertura sintética (SAR)

Se denomina Radar de apertura sintética (SAR) a satélites que son capaces de realizar imágenes de alta calidad y resolución sobre una zona en concreto. En los últimos años se han conseguido disminuir costes para lanzar más satélites SAR, de manera que este tipo de información es más accesible que nunca. A pesar de ello, queda reducido (a día de hoy) a análisis a posteriori de escenarios más que a seguimiento en tiempo real, puesto que la información en directo es privada.

Como se puede observar en la figura Fig.2.15, es posible a partir de estas imágenes de muy alta resolución, reconocer los vehículos que se encuentren en una determinada zona, siendo muy útil para complementar la información obtenida por otros radares, al ser imposible no ser detectado.

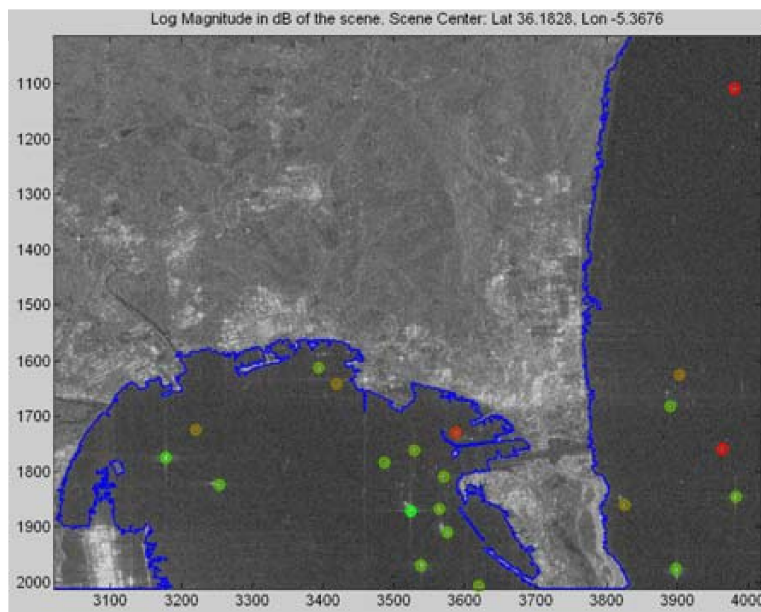


Fig.2.15: Detección de vehículos mediante SAR [8]

## 2.4.2 Radares marítimos

La opción más extendida en el entorno marítimo es instalar un radar en el propio barco para detectar la posición de los vehículos cercanos en cualquier dirección. Un radar funciona emitiendo ondas de alta frecuencia (varía en función del radar) que rebotan contra los objetos que encuentren por el camino y vuelven al radar, detectando la energía con la que vuelve la onda. De esta manera, un barco puede conocer la posición de los distintos vehículos que detecte cercanos, utilizando la pantalla que suelen proporcionar para ello. Además, en función de las características del radar instalado y su colocación en el barco se puede obtener un mayor alcance.

Principalmente se utilizan dos tipos de bandas distintas para transmitir las ondas, ambas dentro del rango Super alta frecuencia (SHF). Estas son la banda S, emitida entre las frecuencias 2 y 4GHz y con dimensión de onda entre 7.5 y 15cm; y la banda X, de frecuencias entre 5.2 y 10.9GHz, y dimensión de onda entre 2.5 y 3.75cm.

Las principales ventajas que aportan los radares son su funcionamiento en cualquier tipo de condición, incluso con lluvia o niebla densa. Además, funciona en todo caso, aunque el barco no se identifique.

#### **2.4.2.1 Identificador amigo-enemigo (IFF)**

Una de las implementaciones más conocidas utilizando radares son los Identificador amigo-enemigo (IFF). Estos sistemas utilizan dos radares con distintos objetivos. El principal detecta el objeto como se explicó anteriormente, mientras que el Radar secundario (SSR) busca realizar una “conversación” con dicho objetivo. Mediante la onda emitida va un mensaje cifrado, que, si el objeto destino es aliado, sabrá descifrar y responder al emisor, de manera que ambos se identifiquen. En caso de no encontrar respuesta, se detectará a dicho objetivo como enemigo (foe).

#### **2.4.3 Sistema de identificación automática (AIS)**

La Organización Marítima Internacional (IMO) añadió en el año 2000 una nueva regulación que obliga a la mayoría de los vehículos marítimos a incorporar sistemas para proporcionar información del barco al resto de barcos cercanos y autoridades costeras [54].

La tecnología detrás de AIS es similar al funcionamiento de un radar. Utiliza un transpondedor (transmisor y receptor) de ondas Muy alta frecuencia (VHF), cuya frecuencia va de 30 a 300 MHz. Mediante estas ondas se envía, entre otros parámetros, información de la posición del barco, captada vía GPS, su orientación, su Identificador de servicio móvil marítimo (MMSI) y el tipo de barco y uso del mismo.

El principal problema de AIS es que no obliga a todas las embarcaciones a llevarlo incorporado. En concreto, no se obliga a buques militares, pesqueros o embarcaciones de recreo de menos de 45 metros, por lo que la visión del entorno es limitada y por ende no puede confiarse únicamente en dicha información. Por último, destacar que hay dos tipos de transpondedores AIS: la clase A para barcos grandes y el B, más barato, para vehículos de menores dimensiones.

Su principal ventaja erradica en el hecho de ser un sistema de obligado uso e identificativo del vehículo detectado, a diferencia del resto de sensores que no identifican al vehículo.

### **2.5 Entornos de estimación existentes**

Una vez comprendida de manera global los componentes principales de la temática de la tesis, se realiza una breve comparativa entre las soluciones existentes encontradas para probar y testear filtros de estimación.



### 2.5.1 MATLAB Toolbox

MATLAB es un entorno y lenguaje de programación, creado en 1984 y de ámbito privado, que ofrece una cantidad inmensa y muy potente de herramientas software de cálculo. Es muy habitual su uso en centros de investigación y universidades. La herramienta es de pago, aunque ofrece licencias gratuitas a estudiantes de múltiples universidades.

Dentro de la herramienta ofrecen la capacidad de incorporar grandes paquetes de programación con funciones específicas de un área de investigación, denominadas Toolbox, permitiendo que en cualquier momento se expandan las funcionalidades del framework desarrollado.

Existe una Toolbox orientada a la temática de este trabajo, denominada "Sensor Fusion and Tracking Toolbox", que proporciona herramientas desarrolladas por la empresa MathWorks, para simular y comprender sistemas de fusión.

Con esta herramienta se puede generar un sistema de fusión completo, desde la detección de los sensores hasta la retroalimentación una vez extraído el contexto.

Su principal problema es que, al ser un software comercial, su manipulación y distribución es limitada, siendo así incapaces de modificar, por ejemplo, un comportamiento concreto de un algoritmo de estimación.

### 2.5.2 Tracker Component Library

Por otro lado, se ha encontrado otra librería implementada en el lenguaje de programación MATLAB. Esta, desarrollada por Crouse [9] integra una gran cantidad de estimadores de posición, así como componentes de otras materias, como de astronomía, predicción atmosférica, etc.

La librería además cuenta con ciertas funciones implementadas en el lenguaje de programación C, para maximizar el rendimiento en las tareas más importantes.

Su principal problema es similar al de MATLAB. La entidad que publica este software es el departamento de la armada de Estados Unidos. Por ello, aunque se proporciona el código fuente, este es poco manejable al estar altamente encapsulado su funcionamiento, siendo poco legible el funcionamiento del mismo. Además, como se observa en la figura Fig.2.16, no parecen estar bien optimizados los filtros para el seguimiento.

No obstante, es una solución e iniciativa maravillosa para la comparación de una alta cantidad de filtros como tiene implementados.

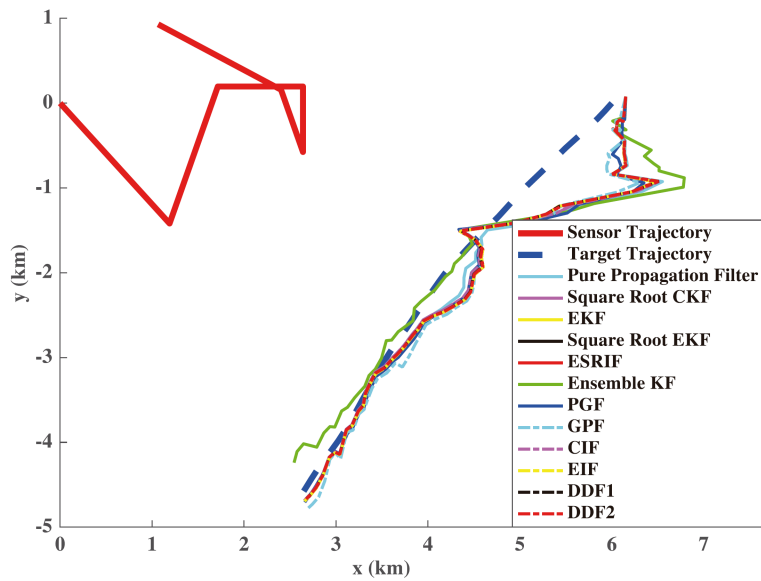


Fig.2.16: Estimación Tracker Component Library [9]

### 2.5.3 Conclusión

Únicamente se han encontrado estas dos soluciones de acceso "sencillo" que permitan realizar una comparativa de estimadores de manera simple. Esto indica que la rama de investigación tiene una lacra en este aspecto, al realizar cada investigador su framework, del cual no se conocen las casuísticas de su implementación, haciendo que su comparabilidad y fiabilidad en los resultados sea dudosa.

Aun así, la existencia de estas dos librerías no inhabilitan el objetivo de esta tesis, al buscar comprender en profundidad el funcionamiento de los algoritmos de estimación, desarrollarlos desde cero todos los algoritmos, entendiendo el código y permitiendo así modificarlos y generar variaciones sobre ellos en el futuro.

## Capítulo 3

# Sistema a desarrollar. Diseño

Una vez analizado el estado del arte de la temática de la tesis, en este capítulo se expone la estructura conceptual del sistema de estimación de posición a desarrollar y su futuro encaje en un futuro sistema de seguimiento global.

Las secciones de este capítulo se ordenan de la siguiente forma: la sección 3.1 expone con mayor detalle el funcionamiento de un sistema de seguimiento global que utiliza fusión de datos, en concreto del nivel 1 del modelo JDL. Después, en la sección 3.2, se expone conceptualmente el objetivo de este trabajo, el funcionamiento del sistema de estimación de posición a desarrollar. Se hace hincapié en el flujo de información percibida por los sensores y su proceso por parte de los estimadores de posición.

### 3.1 Sistema de seguimiento global

#### 3.1.1 Introducción

Tal y como se ha expuesto en la sección 2.1, un sistema de fusión de datos precisa de multitud de componentes o subsistemas perfectamente integrados para funcionar de manera adecuada. Incluso el sistema más simple requiere de varios componentes funcionando en paralelo para proporcionar utilidad. Por ello, para la comprensión del trabajo a realizar en esta tesis y las conclusiones extraídas, es necesario antes comprender internamente el funcionamiento y la complejidad de un sistema de fusión. Tanto la complejidad de sus componentes como la relación unidireccional entre niveles la bidireccional para mejorar el rendimiento del sistema global.

Un sistema de fusión, como se ha observado en el modelo JDL (ver 2.1.1.3), dista de ser un algoritmo con una entrada y una salida. Es el centro de una máquina que genera información y conocimiento constantemente, retroalimentándose de la misma para mejorar su funcionamiento. Como se

puede ver en la figura Fig.3.1, todos los subsistemas son importantes, al influir de manera directa en el funcionamiento de todos los demás. Es por ello que cualquier mejora que se pueda realizar en un sistema de fusión, en cualquiera de sus niveles, es importante para el sistema general.

A partir de la información de los sensores, se realiza el seguimiento de los objetos. Pero es utilizando esa información, tanto procesada como sin procesar, mediante la cual las capas superiores informan a las capas inferiores de como modificarse o trabajar para mejorar la eficiencia y resultados de cada nivel.

Idealmente, la información extraída en los niveles superiores, permitirán indicar a las capas inferiores que los estimadores funcionen en una cierta configuración, obtenida en esta tesis, para estimar de mejor manera el movimiento de dicho blanco.

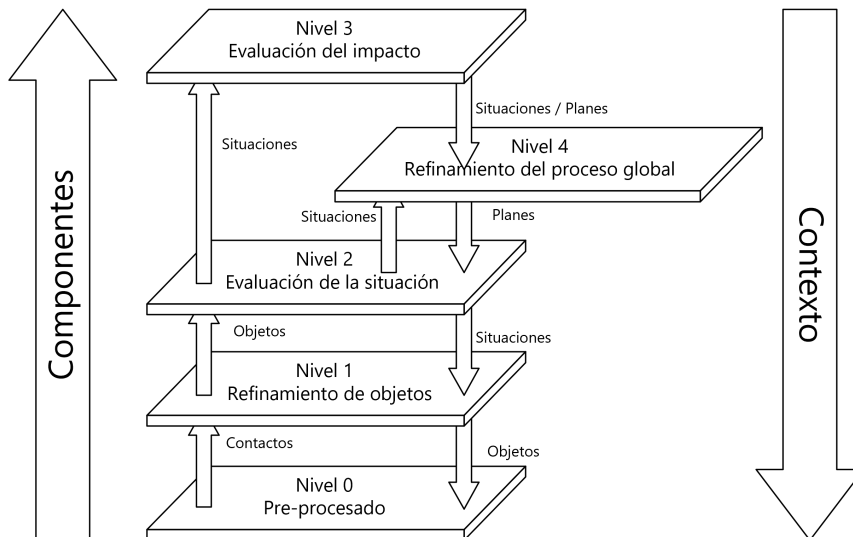


Fig.3.1: Flujo de información de un sistema de fusión [10]

### 3.1.2 Sistema de fusión básico

Esta tesis se centra únicamente en la faceta de estimación de posición, realizando un framework que permite una rápida comparación entre varios filtros distintos. Dicho componente se encuentra en el nivel 1 del modelo JDL.

Obviando la parte de extracción de conocimiento y su retroalimentación (es decir, la parte autónoma de un nodo en una arquitectura descentralizada), un nodo básico de fusión de datos tiene los siguientes componentes: Estos componentes cubren totalmente el nivel 0, y parcialmente el nivel 1 del modelo Unión de directores de laboratorios.

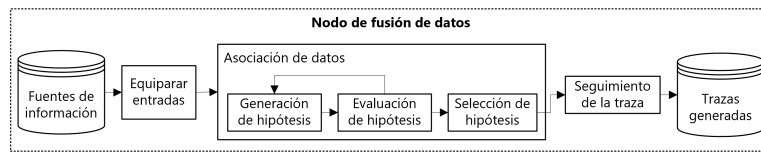


Fig.3.2: Componentes nodo de fusión [11]

Como se puede observar, se tiene en cuenta la necesidad de realizar un pre-procesado, tanto de unidades como de variables, con el objetivo de dejar completamente comparables y utilizables las entradas de cualquier sensor disponible. Sobre el nivel 1 únicamente se realiza el seguimiento de la traza y la asociación de varias fuentes distintas sobre el mismo objeto, dejando de lado facetas como la extracción de características e identificación del blanco. Respecto al trabajo de la tesis, la única diferencia es que este sistema cuenta con asociación de varias fuentes de datos y el propuesto no realiza asociación.

### 3.1.2.1 Nivel 0

Si se profundiza en este sistema de fusión, su nivel 0 puede definirse mediante la figura Fig.3.3.

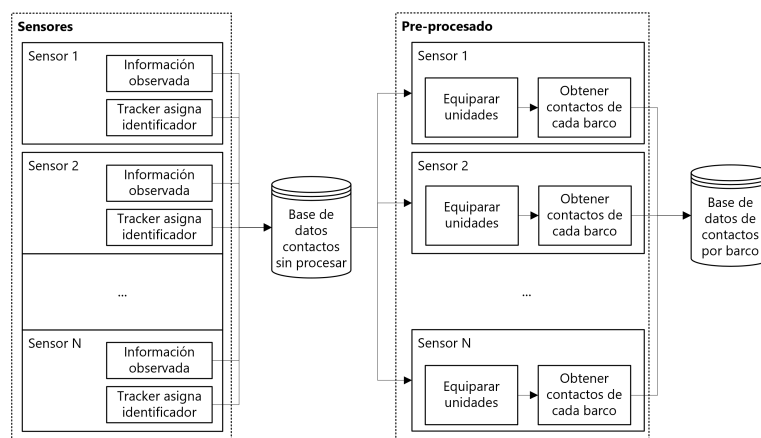


Fig.3.3: Componentes nodo de fusión. Nivel 0

Cada sensor proporciona la información que observa, y, además, provee el identificador del contacto. Estos van a parar directamente a la base de datos del sistema. Una vez almacenados, se les aplica una conversión de unidades, para que todos los contactos estén en las mismas unidades y sean comparables entre sí. Además, se obtienen los contactos de cada barco, de manera que se puedan ordenar cronológicamente (algunos contactos vienen desordenados al tener un mayor retardo para generar la salida su sensor) e introducirlos por orden al filtro de estimación.

### 3.1.2.2 Nivel 1

**3.1.2.2.1 Estimación** El nivel 1 comienza con el filtrado de las trazas que se acaban de obtener. En la figura Fig.3.4 se realiza una representación del proceso a realizar. Cada traza sin filtrar se introduce al subsistema de

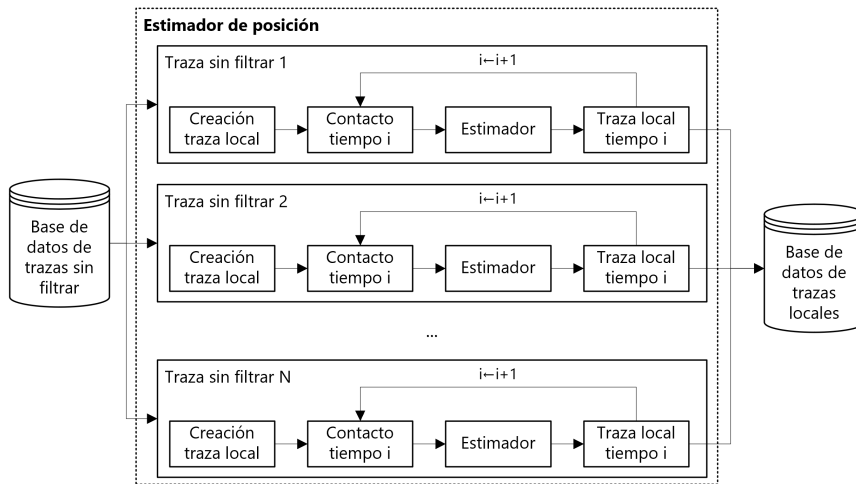


Fig.3.4: Componentes nodo de fusión. Nivel 1 - Estimación

estimación. Al entrar, se le asigna un identificador unívoco a dicha traza, y se comienza su filtrado. Aquí, se realiza la estimación, utilizando uno de los filtros de estimación expuestos en 2.3 sobre cada uno de los contactos de la traza. Su resultado, sobre cada contacto, almacena tanto la estimación producida (traza local) como la información del estimador para la siguiente iteración.

**3.1.2.2.2 Asociación** Por último, se cuenta con la figura Fig.3.5, que representa la fase de asociación de datos del nodo de fusión.

En esta fase se asocian las trazas locales, que filtran por separado a un mismo blanco, en una traza global que se construye mediante la combinación de todas las asociadas. Las asociaciones entre trazas globales y locales se reflejan en otra base de datos denominada la tabla de asociación, la cual es consultada y modificada constantemente.

Para cada actualización de una traza local, se sigue el siguiente proceso. Si aún no está asociada, se comprueba si existe una traza global que cumpla las condiciones para asociarla.

La primera condición es que el sensor de la traza local aún no esté presente en dicha traza global. Esta condición evita que dos barcos se introduzcan en una misma traza global, lo cual sería erróneo. En caso de que sean compatibles los sensores, hay que hacer ambas comparables. Un blanco seguido por sensores distintos, para ser comparables, deben estar en el mismo instante de tiempo.

Y aunque no se tenga el contacto exacto de ambas trazas en ese instante de tiempo, se puede utilizar la información de la estimación para realizar una predicción a priori y propagar el punto más atrasado al mismo tiempo.

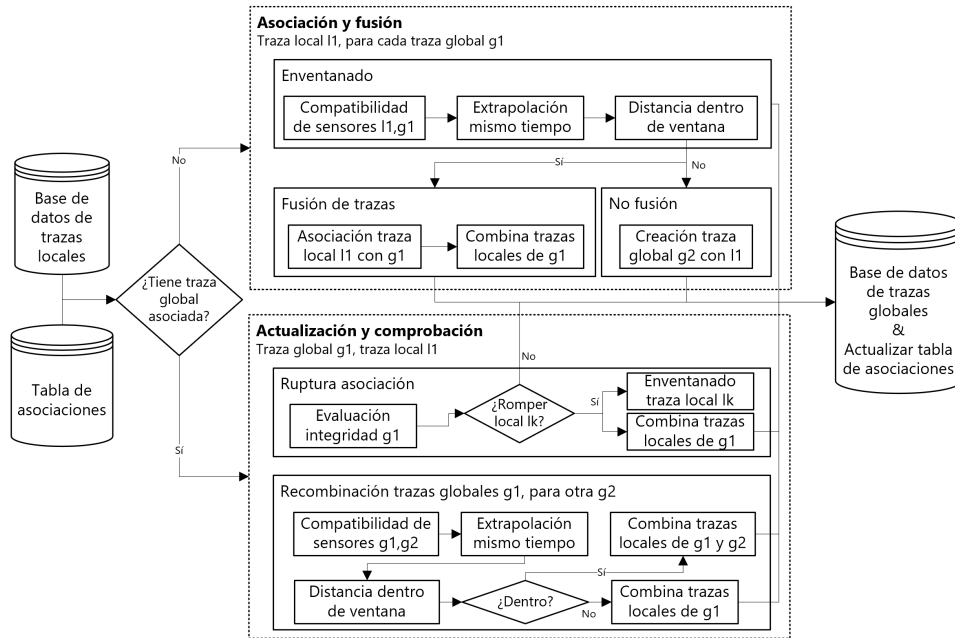


Fig.3.5: Componentes nodo de fusión. Nivel 1 - Asociación

Una vez tienen el mismo tiempo, hay que comprobar que de verdad, en el mismo instante de tiempo, las trazas representen el mismo blanco. Para ello existen varias técnicas, como las expuestas en la sección 2.1.2.1.3. En caso de sí ser asociables las trazas, se apunta dicha asociación en la tabla de asociación, pero si no lo es con ninguna traza global, se concluye que dicha traza debe formar una traza global por sí misma, otorgándole un identificador de traza global y apuntando dicha nueva asociación en la tabla. Por otro lado, en el caso de que dicha traza local ya estuviera asociada, con este nuevo contacto hay que actualizar su traza global. Para ello, primero se comprueba si la asociación sigue siendo correcta, puesto que el sistema puede haber fallado y la asociación realizada ya no sea válida. La primera comprobación que se realiza es si este nuevo contacto sigue la progresión esperada por el barco, extrapolando la traza global al nuevo tiempo y comprobando su distancia. Si sigue dentro, se actualiza la traza global fusionando el nuevo contacto y se termina el proceso. Si no está dentro de la ventana de integridad, se elimina la traza local de dicha traza global, al dejar de parecerse su comportamiento.

La segunda comprobación, evalúa si existe alguna traza global (g2) que represente el mismo barco que la seleccionada (g1). Este caso es frecuente cuando al inicio el contacto es muy ruidoso, no se asocia a la traza global

que le correspondería y genera una nueva. Su funcionamiento es similar al inventariado. Se comprueba que entre ambas trazas no coincidan sensores y si las trazas están lo suficientemente cerca, se unen en una sola traza global, cambiando de dicha manera la tabla de asociación. En caso contrario, se realiza una fusión de trazas locales de la traza global seleccionada (g1).

## 3.2 Sistema a desarrollar

El sistema a desarrollar, a diferencia de un sistema implementado en un barco, no funciona en tiempo real. Esto es debido a la falta del hardware necesario. Por ello, el funcionamiento del sistema va a ser simulado, realizando para ello ligeras variaciones respecto a un sistema que funcione en tiempo real.

El sistema obtendrá directamente los contactos preparados de un único barco, que representen una trayectoria concreta. Con este procedimiento se facilita alcanzar los objetivos de la tesis, tanto el desarrollo del sistema de estimación, al tener que tratar con objetos más simples, como la demostración de las capacidades de análisis de la estimación de posición de un barco ofrecida por los distintos filtros.

Para obtener las trayectorias deseadas, se han definido los siguientes criterios que debe tener una traza que represente un movimiento real de un barco:

- El contacto AIS debe ser emitido por un barco.
- El barco debe notificar su tipo de barco.
- El barco debe notificar la maniobra que está realizando.
- El barco debe estar en movimiento.
- Deben existir mínimo  $N = 50$  contactos del mismo barco para conformar una trayectoria.
- Los contactos deben tener una separación máxima de  $\Delta t = 30\text{seg}$  entre sí.

Una vez introducido el sistema a desarrollar en esta tesis, se pasa a exponer los componentes a desarrollar. Partiendo del modelo de fusión de JDL (ver 2.1.1.3), se requiere generar, como mínimo, los siguientes componentes:

- Fuentes de datos. Sensores
- Base de datos de soporte
- Nivel 0 - Pre-procesado de las fuentes de datos



- Nivel 1 - Control de objetivos
- Interacción humano-computador (HCI)

### 3.2.1 Fuentes de datos. Sensores

Las fuentes de datos a utilizar deben proporcionar información de vehículos marítimos, a poder ser, utilizando sensores habituales en dicho entorno, como los especificados en la sección 2.4. Utilizar datos reales, medidos en el entorno marítimo, dotaría a este trabajo de una fiabilidad mayor en los resultados frente a si se trabajase con datos que simulen el funcionamiento de los sensores.

Cabe destacar que, por la temática de esta tesis, la obtención de información de determinados sensores es compleja o incluso imposible, al ser tecnologías en su mayoría enfocadas a los departamentos de defensa de gobiernos o empresas privadas.

Existe poca información pública accesible para realizar trabajos de investigación en estos ámbitos. Además, el problema aumenta de complejidad si se desea asociar la información de dos sensores. Por ello, en esta tesis y como punto inicial para un sistema global, el foco es únicamente la información obtenida por sensores AIS (más información en la sección 2.4.3).

Tras analizar fuentes de datos AIS disponibles de manera gratuita y observar artículos que exploran varias fuentes de datos AIS [55], se ha decidido utilizar la información de distribuida de manera pública y gratuita por el la Autoridad Marítima Danesa (DMA) [56]. Esta fuente ofrece información de contactos AIS, detectados a lo largo de toda la costa de Dinamarca, gracias a los más de 30 transpondedores [57] distribuidos por la península de Jutlandia y el resto de islas danesas.

La información se proporciona en forma de conjuntos, divididos por días. De esta manera, en cada fichero se encuentran todos los contactos registrados por los transpondedores de cualquier barco que circulase por la costa de Dinamarca.

Haciendo énfasis en la información que proporcionan los contactos AIS [55], sus parámetros más relevantes son los siguientes:

Parámetro	Descripción	Unidades
AIS	Identificador unívoco del barco de AIS.	String
IMO*	Identificador del barco de IMO.	String
Latitud	Coordenadas que representan la posición de un lugar en dirección Norte-Sur.	Coordenadas

Parámetro	Descripción	Unidades
Longitud	Coordenadas que representan la posición de un lugar en dirección Oeste-Este.	Coordenadas
Ratio de giro (ROT)*	Valor que indica el número de grados que gira cada 30 segundos.	Número entre -127 (izq) a 127 (der)
Velocidad sobre el mar (SOG)	Velocidad producida por el barco, sin tener en cuenta el viento ni oleaje.	1/10 nudos
Curso sobre el mar (COG)	Dirección hacia donde se esperaba que apunte el barco para llegar al siguiente punto.	Grados
Heading	Dirección hacia donde apunta el barco actualmente (no hacia donde se mueve).	Grados
Eslora	Metros de punta a punta del barco.	Metros
Manga máxima	Metros de ancho del barco en el punto de máxima anchura.	Metros
Calado*	Metros de profundidad en el agua del barco.	Metros
Destino*	Nombre del destino del barco.	String
Tiempo de llegada estimado*	Tiempo restante estimado para llegada al destino indicado.	Tiempo UTC
Tiempo actual	Instante de tiempo actual.	Tiempo UTC
Estatus de navegación*	Valor que indica el tipo de maniobra que está realizando el barco. Se configura a mano. Por ejemplo: Movimiento utilizando motor, barco anclado (con el ancla) o amarrado a puerto, etc.	Número
Tipo de barco	Valor que identifica el tipo de barco. Por ejemplo: carguero, pesquero, militar, Barco de pasajeros, barco personal, barco cisterna, remolcador, etc.	Número
Tipo de cargo	Concreta específicamente el tipo de barco en caso de ser tipo carguero	Número

Tab.3.2: Parámetros de AIS

\*No es obligatorio en transpondedores de clase B. En los de clase A sí.

Esta gran cantidad de información es recibida en cada uno de los contactos de cada barco. De media, un fichero diario de la costa de Dinamarca contiene diez millones y medio de contactos AIS repartidos en más de 11000 barcos distintos.

Esta información, aparte de para el trabajo a realizar en esta tesis, tiene potencial para ser aprovechada en el resto de niveles del modelo de fusión JDL, como se realiza en [28].

### 3.2.2 Base de datos de soporte

Otra parte esencial es almacenar la información, tanto los contactos originales de los sensores, como las posiciones filtradas y el resto de tareas de posteriores niveles de fusión (asociación de trazas, conocimiento, etc.). Ofrecer sistemas de seguridad sobre la información es prioritario en un sistema final, tanto de integridad y tolerancia a fallos, como de autenticación para el acceso a la información.

Esta labor de seguridad, pese a importante, se descarta para esta tesis, al no buscar ofrecer un servicio de fusión de datos final. Por tanto, el almacenaje de la información, por facilidad de uso, se realizará en diversos ficheros de texto plano sin codificar.

### 3.2.3 Nivel 0 - Pre-procesado de las fuentes de datos

Los datos propuestos teóricamente son válidos para realizar una estimación de posición con ella, pero debido a su formato (fichero de texto) y orden (ordenados vía Tiempo actual), es necesario aplicar sobre estos datos reales un profundo pre-procesado para obtener la información necesaria para estimar una trayectoria de un barco.

Todos los estimadores de posición explicados en la sección 2.3.3 precisan de información de tiempo, latitud y longitud para funcionar, así como un identificador del vehículo a filtrar. Los filtros aún así cuentan con ciertas restricciones, principalmente de tiempo.

Al estar la precisión de la estimación directamente asociada a los tiempos de los contactos, es necesario realizar un pre-procesado limitando la distancia temporal máxima entre los contactos de una traza, y, por otro lado, establecer una cantidad mínima de contactos seguidos de cada barco para conformar una traza. De esta manera, se eliminan contactos sueltos (fruto de utilizar datos reales) y se garantiza que las trazas a filtrar tengan unas características esperables de barcos reales, con una periodicidad de contactos alta y realicen una trayectoria suficientemente larga.

### 3.2.3.1 Separación por MMSI

Debido a la necesidad de realizar esta comprobación, el primer paso del pre-procesado es separar las trazas directamente por identificador de barco MMSI. De esta manera, el proceso para comprobar la diferencia temporal entre un contacto y el siguiente es más simple.

Aun así, este paso tiene más complejidad, al no estar perfectamente documentados los contactos. Como un mismo barco puede ser detectado por distintos receptores y posteriormente estos contactos han sido unificados, en función de la clase del receptor, algunos contactos tienen más información que otros. Por ello, en este mismo paso, se realiza un replicado de la información estática de cada barco, y se eliminan los barcos cuya información no es detallada, para facilitar la selección de trazas y su posterior análisis.

En la figura Fig.3.6 se realiza un diagrama de flujo que representa este paso.

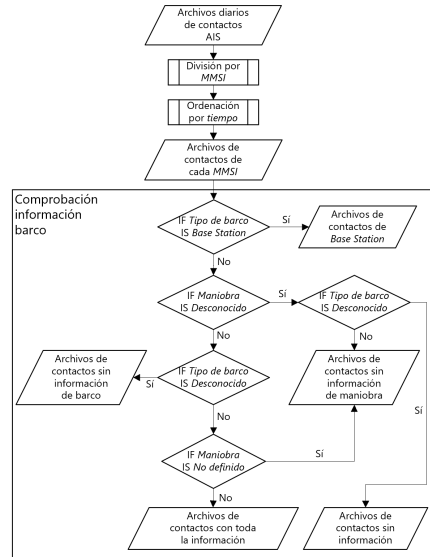


Fig.3.6: Sistema a desarrollar. Nivel 0 - Pre-procesado

### 3.2.3.2 Extracción de trazas

Una vez están los contactos divididos por MMSI con toda la información posible de AIS, el siguiente paso es, como se ha comentado, extraer trayectorias útiles para introducirlas en los filtros de estimación.

De esta manera, un mismo barco, mediante un mismo sensor, puede contar con más de una traza. Todas ellas se almacenan en otra base de datos distinta, al estar ya preparadas para ser filtradas. En la figura Fig.3.7 se observa el flujo del algoritmo que extrae las trazas.

Este algoritmo recorre todos los ficheros de barcos con toda la informa-

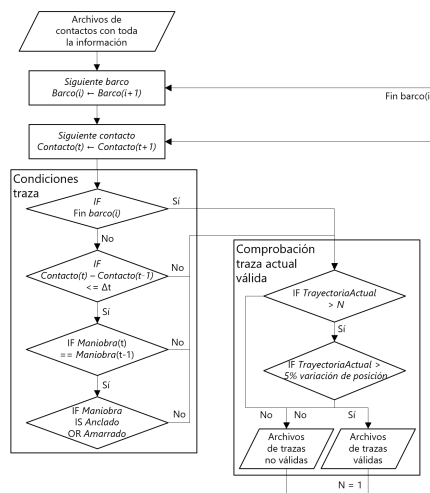


Fig.3.7: Sistema a desarrollar. Nivel 0 - Extracción de trazas

ción deseada y extrae comprueba que la traza cumple las tres últimas condiciones indicadas al comienzo de este apartado. El proceso que sigue es sencillo, sobre dos bucles que recorren los barcos y los contactos, se concatenan varias condiciones que si se cumplen, permiten añadir contactos a una lista de contactos que potencialmente sea una traza válida.

Cuando una de las condiciones se incumpla, se evalúa si la traza almacenada es hasta este instante es válida. El primer paso es si cuenta con más de 50 contactos. La segunda condición garantiza que el barco realice un mínimo desplazamiento sobre su latitud y longitud, evitando así que el parámetro de maniobra esté mal configurado (se recuerda que este se configura manualmente por el operario del barco).

Una vez se completa una traza, tanto validada como descartada (en función de la condición se almacena en una localización diferente), se utiliza este último contacto como el primero de la siguiente posible trayectoria.

### 3.2.3.3 Sistema de referencia

Otro aspecto importante a la hora de realizar un sistema con tantos subcomponentes y sensores distintos, es fijar el sistema de referencia la localización de los blancos detectados. En la figura Fig.3.8 se ilustra una simulación mostrando como medir distancias con el sistema de coordenadas utilizado.

Este es un sistema cartesiano (XYZ), cuyo centro (0,0,0) siempre es el centro de fusión, es decir, el propio vehículo que implemente este sistema de fusión. Este sistema de referencia centrado en el vehículo se denomina Punto de referencia el propio barco (OSRP).

Los ejes son definidos mediante el sistema Norte, este, abajo (NED), proporcionando al norte el eje X, al eje Y el este y la altura según la elevación sobre el mar.

En los escenarios a desarrollar en esta tesis, al no haber ningún punto de referencia concreto sobre el que realizar las mediciones, se fija en un punto central de la costa de Dinamarca, de manera que no haya que moverlo en ningún momento. Esta decisión no perjudica el rendimiento, únicamente se tendrán distancias mucho más elevadas que las encontradas en un caso real. En concreto, se fija el punto en las coordenadas latitud y longitud: 56.054349, 10.989258.

**3.2.3.3.1 Unidades** Las unidades del sistema de referencia son metros, de manera que la distancia entre el propio barco y los contactos se mida en metros en el eje X (este positivo, oeste negativo), Y (norte positivo, sur negativo) y Z (elevado positivo, submarino negativo).

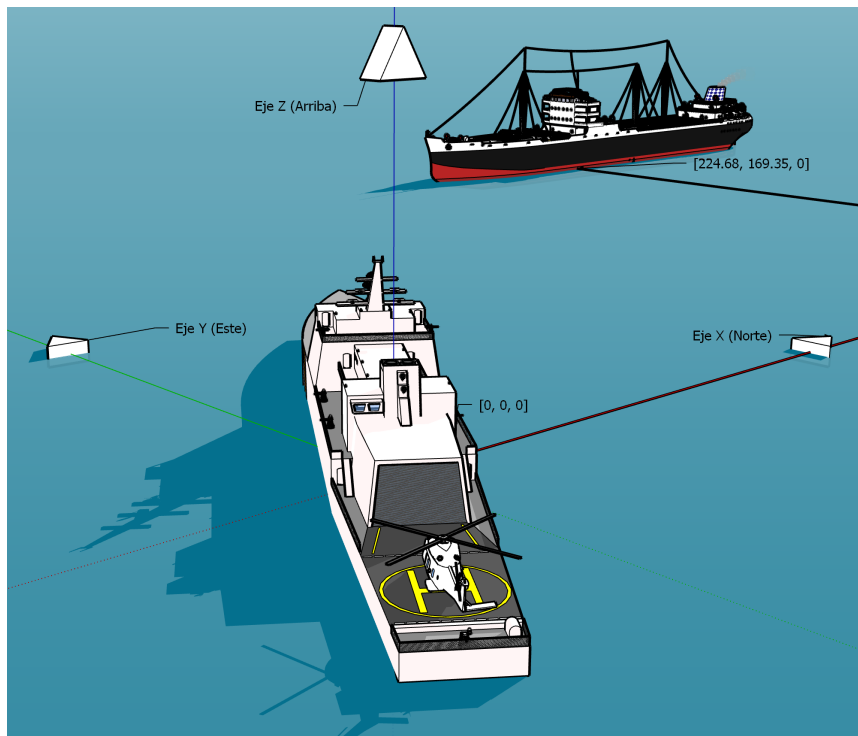


Fig.3.8: Sistema de coordenadas OSRP-NED

Para la velocidad, se utilizarán metros segundo, para facilitar las operaciones internas del sistema con las posiciones de los blancos.

### 3.2.4 Nivel 1 - Control de objetivos

Una vez se tienen una serie de trayectorias de barcos en movimiento preparadas para ser analizadas, es el momento de realizar el filtrado de estimación. Para este objetivo, se han seleccionado los siguientes filtros:

- Filtro de Kalman Extendido (EKF) (ver 2.3.3.3).
- Filtro de Kalman “Unscented” (UKF) (ver 2.3.3.4).
- Filtro de Kalman Ensemble (EnKF) (ver 2.3.3.5).
- Filtro de Múltiples modos interop-erantes (IMM) (ver 2.3.3.6).

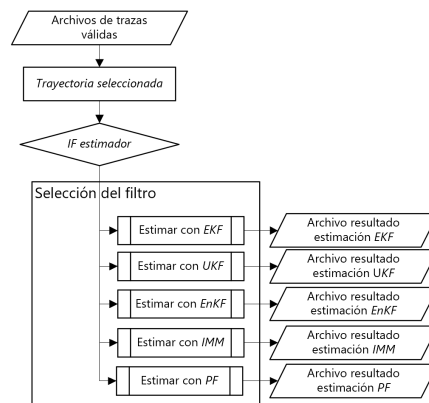


Fig.3.9: Sistema a desarrollar.

- Filtro de partículas (PF) (ver 2.3.3.7).

El proceso de este paso es simple (descrito en la figura Fig.3.9). A partir de los contactos de una trayectoria del paso anterior, se selecciona el filtro a utilizar y sus resultados se almacenan en su fichero correspondiente. El proceso de cada filtro está explicado anteriormente en el estado del arte.

### 3.2.5 Interfaz humano-computador

Finalmente, contamos con la interfaz que permite al humano observar y comparar la información obtenida para realizar un análisis de los resultados de los filtros. Este componente cuenta con varios modos, en función de la información deseada:

- Acceso a los ficheros de resultados de cada filtro por separado.
- Acceso a las métricas específicas de la estimación realizada por un filtro.
- Gráficas específicas de la estimación de cada filtro (o los contactos).
- Gráficas generales comparativas de la variación de un filtro en función de la configuración utilizada.
- Gráficas generales comparativas de la estimación de varios filtros.

De esta manera, se proporcionan tanto acceso a la información más específica, como abstracciones más generales para observar en un vistazo toda la información.

Las gráficas escogidas para comparar los distintos filtros muestran los valores de posición y velocidad a lo largo del tiempo. También se genera otra gráfica donde se muestra la posición en los ejes XY, observando el movimiento realizado por el blanco.

Respecto a las métricas utilizadas, al no contar con ground truth en los datos seleccionados, ni tener implementado un sistema Monte-Carlo para aproximarlos, se proporcionan resultados de:

- RMSE con innovación de posición (ver 2.3.4.1).
- PCRLB (ver 2.3.4.1).
- NIS (ver 2.3.4.2).
- Tiempo de ejecución del filtro (ver 2.3.4.3).

## Capítulo 4

# Sistema a desarrollar. Implementación

En este capítulo se exponen las características del desarrollo del framework a realizar. La sección 4.1 expone el utilizado para todo el desarrollo y pruebas. La sección 4.2 indaga en las características de la programación del framework, desde los datos hasta la impresión de resultados.

### 4.1 Entorno de desarrollo

A continuación, se expone todas las características del entorno de desarrollo utilizado para la capacidad de replicación del trabajo realizado.

#### 4.1.1 Lenguaje de programación

Para el desarrollo del framework de estimación se ha escogido el programa MATLAB r2019a, tanto por su facilidad para el desarrollo de operaciones matriciales, muy presentes en los estimadores de posición, como por sus funcionalidades pensadas por y para científicos que proporcionan una facilidad a la hora de realizar pruebas y extraer resultados. Además, estas están totalmente optimizadas y cuentan con una gran documentación.

Destacar sobre manera la capacidad de incorporar grandes paquetes de programación con funciones específicas de un área de investigación, denominadas Toolbox, permitiendo que en cualquier momento se expandan las funcionalidades del framework desarrollado. En este framework se utiliza la Toolbox de computación de altas prestaciones, cuyo funcionamiento es clave para desarrollar las pruebas en menor tiempo, aprovechando las capacidades del servidor utilizado (ver Tab.4.2).



### 4.1.2 Computador utilizado

El entorno donde se ha desarrollado el sistema y realizado todas las pruebas es un servidor, cuyas características se muestran en la tabla Tab.4.2. En este caso es de especial relevancia la alta capacidad del sistema de almacenamiento, al realizar gran cantidad de operaciones de lectura / escritura. También destacar el procesador, que junto a la memoria RAM, permite al sistema añadir paralelización para filtrar varias trazas a la vez, o lanzar una cantidad de partículas mayor.

Componente	Marca	Modelo	Capacidad
Procesador	AMD	Threadripper 2950X	16 núcleos
RAM	Corsair	DDR4 3000	32GB
Almacenamiento	Samsung	970 EVO	1TB
Tarjeta gráfica	Gigabyte	RTX 2070	8GB

Tab.4.2: Componentes PC utilizado

## 4.2 Estructura del proyecto

Para la programación en MATLAB se ha optado por un paradigma similar lo más similar a la orientación a objetos posible, para facilitar su comprensión y permitir el cambio de cualquier sub-sistema por otro en cualquier momento sin tener que modificar apenas el código. Para ello, se han desarrollado todos los sub-sistemas utilizando funciones, con entradas y salidas de variables fijas. La estructura del proyecto se muestra en Fig.4.1.

Las carpetas del framework son las siguientes:

- Configuration: se proporciona mediante ficheros externos la configuración de los filtros de estimación a utilizar. De esta manera, se puede cambiar de configuración sin tener que modificar el código. También se cuenta con una configuración global del framework para proporcionar variables de control del script, y la posición del centro de fusión.
- Data: los datos de entrada de cada proceso se encuentra en esta carpeta. Se cuenta con una carpeta de los datos de Dinamarca, tanto todos como

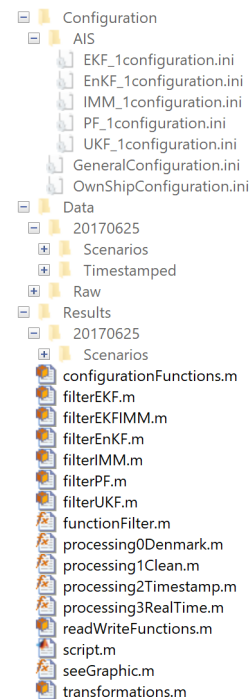


Fig.4.1: Estructura Matlab

los escenarios a probar, y los contactos puros del día escogido.

- Results: por último, se tiene la carpeta de resultados, donde se almacena la información generada por cada filtro, así como las gráficas comparativas de cada escenario.

A continuación, se muestra, de manera resumida, el objetivo de cada uno de los ficheros del framework:

- configurationFunctions: proporciona métodos de lectura para la configuración del filtro de estimación a utilizar. También lee configuración global del framework, y la posición del centro de fusión.
- filterX: los siguientes seis ficheros incorporan las funciones necesarias para realizar el proceso de filtrado de cada estimador. Como mínimo cuentan con un método de inicialización, otro de predicción y otro de corrección con la medida.
- functionFilter: proporciona al filtro deseado los contactos para filtrar y gestiona las variables del mismo para la siguiente iteración.
- processing0Denmark: a partir de los contactos puros, se dividen por Identificador de servicio móvil marítimo.
- processing1Clean: separa los barcos con información de los que no la tienen. Además, trata de completarla en todos sus contactos.
- processing2Timestamp: obtiene trayectorias de cada barco con ciertas condiciones temporales y espaciales.
- processing3RealTime: para cada trayectoria seleccionada (escenario), se recorre sus contactos y los filtra. Permite su expansión para más componentes de sistemas de fusión.
- readWriteFunctions: realiza la gestión de los ficheros de contactos y trazas. Implementa métodos de lectura y escritura en función del filtro, del contacto o del posterior uso de lo leído.
- script: orquesta todo el cómputo de la tesis, realizando llamadas a los métodos processing y seeGraphic.
- seeGraphic: función que, en función de los contactos recibidos y el estimador utilizado, genera el gráfico deseado, permitiendo comparar distintos filtros entre sí.
- transformations: lista de funciones implementadas para realizar la transformación de unidades a las unidades fijadas en el sistema.

### 4.2.1 Extracción de trazas

Como se ha introducido anteriormente, en esta versión inicial del framework se introducen trazas preparadas de barcos moviéndose, puesto que contactos muy separados temporalmente o barcos quietos no son el interesantes para su análisis.

De esta manera, a partir de un fichero con contactos AIS de Dinamarca (en concreto del 25 de julio de 2017). De sus 10565262 contactos de 3720 barcos distintos, al final del proceso se han extraído 12416 trazas distintas de 1618 barcos distintos, mediante el proceso de la figura Fig.4.2.

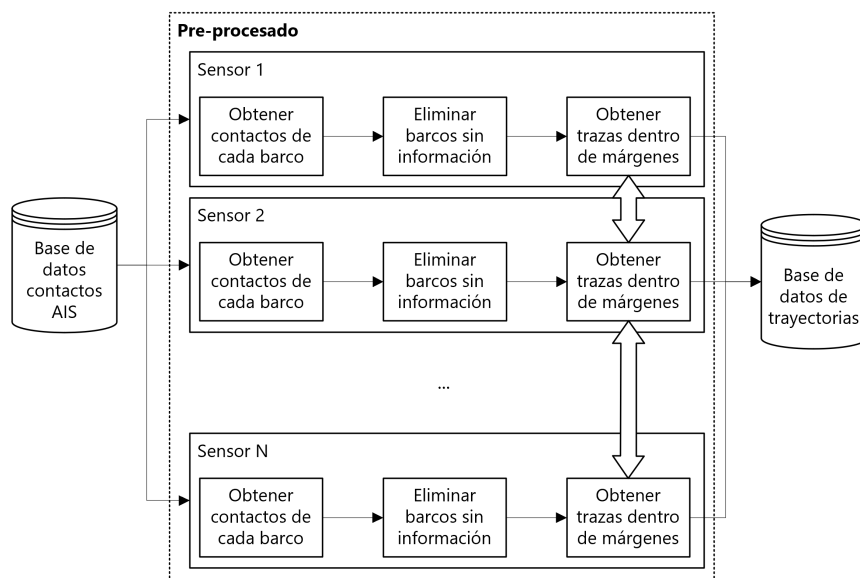


Fig.4.2: Extracción de trazas

Este proceso, en el código, se realiza mediante el algoritmo Alg.4.1.

---

#### Alg.4.1 Extracción de trazas

---

```
1: pathRawFiles = processing0Denmark('aisdk20170625')
2: pathCleanedFiles = processing1Clean(pathRawFiles)
3: maxPeriod = 30, minNumTracks = 50, thresholdMovement = 5
4: pathTimestampedFiles = processing2Timestamp(pathCleanedFiles,
    maxPeriod, minNumTracks, thresholdMovement)
```

---

### 4.2.2 Script tiempo real

Una vez extraídas las trazas de barcos con movimiento, se realiza el filtrado de las mismas. Este proceso, preparado para futuras ampliaciones,

permite realizar el filtrado de varios sensores con distintos blancos a la vez. Su pseudocódigo se muestra en el algoritmo Alg.4.2. Para cada fichero de trayectorias, se le aplican los N filtros de estimación que se escojan, siendo el último, ‘Plots’, para añadirlo en la gráfica comparativa.

---

**Alg.4.2** Filtrado todas las trazas

---

```

1: allFiles = dir(path)
2: for k=1:length(allFiles) do                                ▷ Para todos los archivos
3:   nameFile = allFiles(k).name
4:   filters = ['EKF-1', 'EnKF-1', 'IMM-1', 'PF-1', 'UKF-1', 'Plots']
5:   graphs = ['verXY', 'verXTime', 'verYTime', 'verVxTime', 'verVy-
   Time', 'computeTime']                                     ▷ Se generan las siguientes
   gráficas
6:   plotsFigure = size(filters, graphs)                       ▷ Almacena las trayectorias de
   cada filtro y gráfica
7:   sensor = 'AIS'
8:   for i=1:length(filters) do                               ▷ Para todos los estimadores
9:     processing3RealTime(path, nameFile, filters(i), sensor)
10:  full = false                                             ▷ El gráfico con todos los items
11:  for i=1:length(graphs) do                                 ▷ Para todos los gráficos
12:    for j=1:length(filters) do                             ▷ Para todos los estimadores
13:      processing3RealTime(path, nameFile, filters(i), sensor)
14:      plotsFigure(i,j) = seeGraphic(path, nameFile, filters(j), sen-
   sor, graphs(i), full)                                     ▷ Imprime la trayectoria en una
   figura
15:      legend(plotsFigure(i,:), filters(i,:))               ▷ Añade en la leyenda cada
   filtro al gráfico i saveAsImage()                       ▷ Guarda el gráfico como una imagen

```

---

#### 4.2.2.1 Función tiempo real

Para realizar el filtrado, se llama a una función auxiliar que recibe la información del escenario a procesar (uno o más barcos) y los sensores que tiene el mismo, y los procesa. En este caso al método únicamente le llegan contactos AIS de un mismo barco. Además, se le pasa el filtro para que lo utilice.

Una simplificación de este método se puede ver en Alg.4.3. Primero se prepara toda la información: contactos, configuración, etc. y se inicializan las variables a utilizar que almacenen el proceso. Una vez se tiene la información, se llama a la función de cada filtro (en el código: functionEKF, functionEnKF, functionIMM, functionPF o functionUKF) según corresponda.

---

**Alg.4.3** processing3RealTime

---

```
1: procedure PROCESSING3REALTIME(path, nameFile, filter, sensor)
2:   plots = readPlots(path, nameFile, sensor)
3:   confFilter = readConf(filter)
4:   confGlobal = readConf('Global')
5:   ownPosition = readOwnPos('Dinamarca')
6:   pathWrite = string(path, nameFile, filter)
7:   filtState = vector(plots)    ▷ Almacena la iteración previa del filtro
8:   filtState(i) = functionFilter(filter, confFilter, filtState(i-1), plots(i),
   pathWrite, ownPosition)    ▷ Se filtra el contacto en el filtro seleccionado
```

---

#### 4.2.2.2 Función filtrado

Una vez dentro de la función que orquesta el funcionamiento de cada filtro, los cuales tienen la misma estructura, mostrada en Alg.4.4. Con para contacto, lo primero es transformar, mediante la función `posWGS84toCar`, las coordenadas geográficas a las posiciones cartesianas relativas al centro de fusión (ver Fig.3.8).

A continuación, según la posición del contacto respecto a la trayectoria tiene un comportamiento distinto.

- El primer contacto se utiliza para apoyar a la creación de la traza local de este filtro en el segundo. Esto se realiza para inicializar de manera precisa el vector de estado utilizando las medidas.
- Este segundo contacto pone en marcha el filtro, inicializando todas las matrices de apoyo ( $R, H$ ) y la matriz de covarianza con el ruido esperado a priori  $P$ .
- Una vez inicializado el filtro, en las siguientes iteraciones se realiza el mismo proceso. Con las variables del filtro de la iteración anterior, el incremento de tiempo se realiza la predicción de la posición esperada por el filtro en dicho instante de tiempo. Después, utilizando la medida de la observación, se corrige dicha predicción, dando lugar a la posición final estimada por el filtro. Finalmente, se utilizan las variables anteriores para extraer las métricas de este proceso, incluido el tiempo de computación de esta iteración.

#### 4.2.2.3 Función impresión resultados

Una vez completado el filtrado de la traza, se ha escrito toda la estimación y sus métricas asociadas en ficheros de texto, se generan las gráficas deseadas, tanto del vector de estado como de las métricas. En el algoritmo Alg.4.5 se muestra una simplificación del código real de la función `seeGraphic`.

---

**Alg.4.4** functionFilter

---

```
1: procedure FUNCTIONFILTER(filter, confFilter, filtStatePrev, plots,
  pathWrite, ownPosition)
2:   for i=1:length(plots) do                                ▷ Para todos los contactos
3:     [x,y,z] = posWGS84toCar(plots(i).Lat, plots(i).Lon, plots(i).Hei,
  ownPosition)
4:     if i == 0 then                                       ▷ Se almacena para la siguiente
5:       stateVector(x,y,0,0)
6:       filtStatePrev = save(stateVector, plots.Time)
7:     else if i == 1 then                                   ▷ Se inicializa el filtro
8:       [xLast,yLast] = filtStatePrev.StateVector
9:       period = plots(i).time(1) - filtStatePrev.time
10:      vx = (x - xLast) / period
11:      vy = (y - yLast) / period
12:      [stateVector, Pk, R, H] = filter.initialize(x,y,vx,vy,confFilter)
13:      filtStatePrev = save(stateVector, plots.Time, Pk, R, H)
14:      writeFilter(filtStatePrev)
15:     else                                                 ▷ Se itera el filtro
16:       period = plots(i).time(1) - filtStatePrev.time
17:       [SVPred,PkPred] = filter.predict(period,confFilter, filt-
  StatePrev)                                               ▷ Puede variar ligeramente en función del filtro
  utilizado
18:       [SVFinal,PkFinal] = filter.update(x,y,confFilter,filtStatePrev,
  SVPred,PkPred)
19:       filtStatePrev = save(SVFinal, PkFinal, plots.Time)
20:       [metrics] = filter.metrics(period,SVPred,PkPred, filt-
  StatePrev)                                               ▷ Se extraen las métricas de cada filtro y
  escenario
21:       writeFilter(filtStatePrev, metrics)
```

---

---

**Alg.4.5** seeGraphic

---

```
1: procedure SEEGRAPHIC(path, nameFile, filter, sensor, graph, full)
2:   conf = tipoFiltro(filter)                                ▷ Color, marcador, etc.
3:   tracks = readTracks(path, nameFile, filter)
4:   [varAxisX, varAxisY] = getVariables(track, graph)
5:   graph = generateGraph(varAxisX, varAxisY)
6:   graph.title(nameFile, filter, graph)
7:   if full == true then                                    ▷ Se añade más items del filtro al gráfico
8:     items = extraItems(filter)
9:     graph.add(items)
```

---

Este código genera la gráfica con las características indicadas por parámetro de un estimador (o contactos) indicado.

Inicialmente obtiene, a partir del filtro, las características del trazo a realizar (grosor, color, marcador, etc.). También se lee el fichero con los contactos filtrados e información del filtro.

El siguiente paso selecciona de la traza del escenario, las dos columnas del fichero a utilizar para generar el gráfico. Finalmente, si se ha indicado, según el filtro se permite escoger unos ítems extra u otros para la impresión. Por ejemplo, el PF tiene partículas, pero no tiene la probabilidad de modo del IMM. Los ítems son los siguientes:

- Vector velocidad
- Matriz de covarianzas
- Partículas
- Probabilidades de modo (solo IMM). En gráfico aparte.

## Capítulo 5

# Comparativa

En este capítulo se realiza un análisis utilizando como única herramienta el framework desarrollado en los capítulos anteriores. El objetivo de este análisis es demostrar las capacidades que otorga este software para comparar distintos filtros y características de los mismos en cualquier tipo de escenario.

La sección 5.1 indica los aspectos que se buscan en escenarios para realizar el análisis, así como una explicación del escenario antes del filtrado. La sección 5.3 realiza el análisis, comparando las trayectorias filtradas con los filtros estudiados y variantes de los mismos.

### 5.1 Tipos de escenarios

Para demostrar el correcto funcionamiento de un filtro, es necesario someterlo a situaciones complejas que comprometan su capacidad de seguimiento, tanto en el tiempo como en la posición. Por ello, se buscan cuatro tipos de escenarios.

- El primero será el movimiento más habitual, dirección y velocidad constante, sin apenas cambios de dirección.
- El segundo realizando un giro suave a lo largo del tiempo, manteniendo una velocidad constante.
- El tercer escenario debe completar un giro brusco de dirección.
- Finalmente el cuarto espera con giros constantes de distinto viraje.

Con estos cuatro tipos de escenario propuestos, se cuenta con una variedad suficiente para analizar los filtros implementados.



### 5.1.1 Selección de escenario

Para contar con imágenes que faciliten la selección de las trazas a escoger, se utiliza la función seeGraphic con los contactos (plots), utilizando el gráfico XY.

Tras observar las imágenes generadas, se observa que algunas trazas sin apenas movimiento se han filtrado por el criterio introducido para evitarlas. A pesar de ello, al verlas, se evitan fácilmente.

#### 5.1.1.1 Escenario 1

El escenario 1, sin apenas giros, es el conformado por un carguero, con identificador MMSI: 209207000. Esta trayectoria cuenta con 65 contactos esparcidos a lo largo de 11 minutos y 30 segundos, categorizada como ‘movimiento utilizando motor’.

El gráfico XY del escenario 1 se muestra en la figura Fig.5.1.

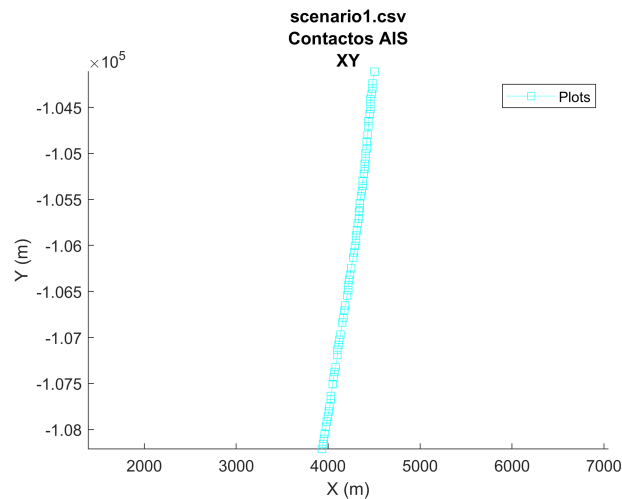


Fig.5.1: Escenario 1. Contactos

#### 5.1.1.2 Escenario 2

La segunda trayectoria seleccionada para su análisis es la realizada por MMSI 211346180, un velero entre las 17:55 y 18:13 horas.

Como se observa en la figura Fig.5.2, el barco realiza un ligero giro hacia la derecha que finalmente corrige. Este comportamiento es común en los veleros para aprovechar el viento.

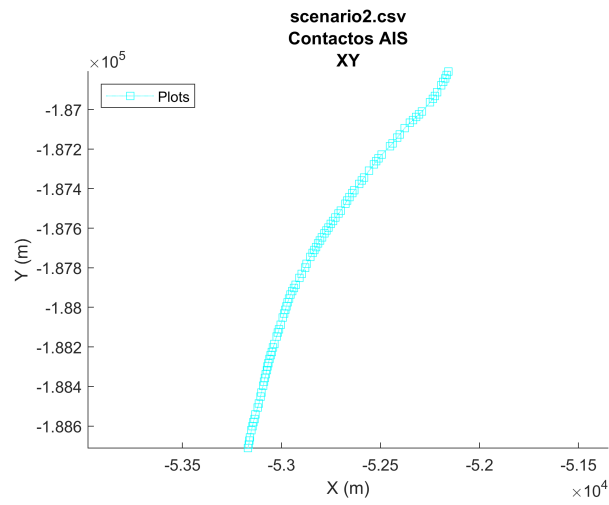


Fig.5.2: Escenario 2. Contactos

### 5.1.1.3 Escenario 3

El tercer escenario lo completa un barco de pasajeros (MMSI 211255580), realizando una serie de zig-zag durante 12 minutos. Cabe destacar que el concepto barcos de pasajeros es muy ambiguo, y abarca tanto vehículos personales y yates como cruceros inmensos. En este caso se trata de un ferry mediano, de 50 metros de eslora.

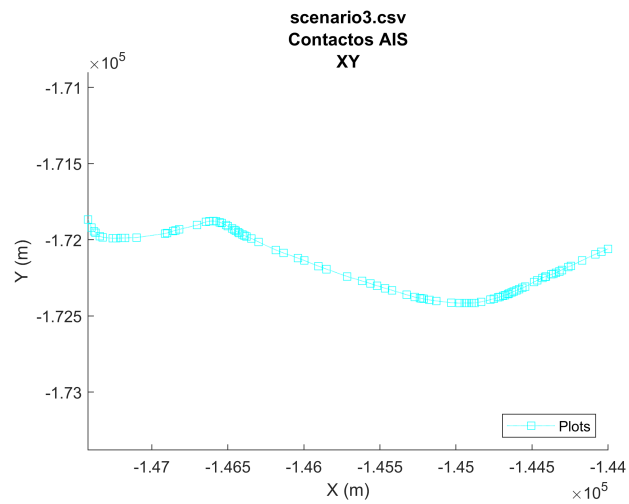


Fig.5.3: Escenario 3. Contactos

#### 5.1.1.4 Escenario 4

Finalmente, contamos con el barco con identificador 211178260, este es un barco de pasajeros que realiza un paseo marítimo por la ciudad alemana Flensburg. Debido a ello, esta trayectoria es tan peculiar, durando dos horas, 10 minutos y 50 segundos y terminando exactamente en el punto de comienzo.

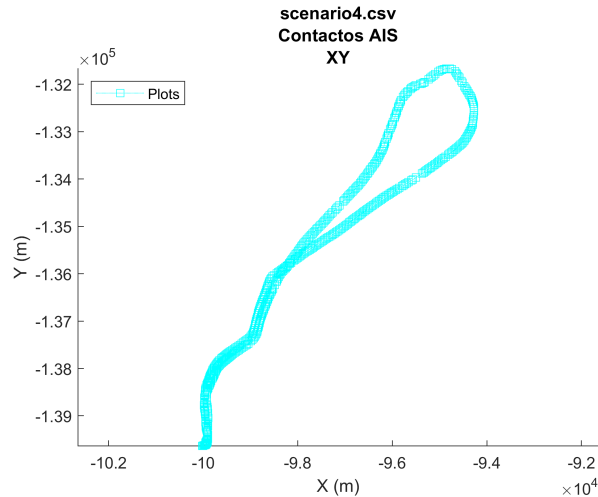


Fig.5.4: Escenario 4. Contactos

#### 5.1.1.5 Características de los contactos

A continuación, en la tabla Tab.5.2 se realiza un breve análisis de la temporalidad de los contactos, para comprender los mismos y su posible implicación en el filtrado.

Escenario	Media	Desviación estándar
Escenario 1	10,95 seg	3,69 seg
Escenario 2	12,06 seg	4,94 seg
Escenario 3	08,07 seg	6,21 seg
Escenario 4	10,49 seg	3,12 seg

Tab.5.2: Características periodo escenarios

Como se puede observar, los escenarios tienen una tasa de refresco centrada entorno a 10 segundos, contando con una alta variación respecto a la media en los escenarios 2 y 3.

Esto puede afectar al funcionamiento de los filtros, haciendo que el ruido asociado disminuya si se reciben muchos contactos seguidos y lo contrario si vienen muy separados. Este ruido afecta a la confianza en la predicción realizada, apoyándose en mayor medida en la observación a mayor ruido.

## 5.2 Configuración de cada filtro

Es necesario para garantizar la replicabilidad de las pruebas y para comprender las mismas, definir los parámetros de configuración utilizados en cada filtro.

De manera global, al utilizar contactos de un sensor AIS, se asigna un error de posición de 10 metros, y de velocidad de 5 metros/segundo. Estos valores se utilizan a la hora de generar la matriz de covarianzas  $P$  y el modelar el ruido de la observación  $R$ .

Los filtros EKF e IMM incorporan un parámetro que amplía o disminuye el ruido del vector de estado de la  $Q$ . El utilizado en EKF y en el modo de velocidad constante del IMM es 0.05, mientras que para el modo ruidoso es 0.4.

El filtro IMM a su vez cuenta con más parámetros:

- Un número de modos a utilizar. Como ya se ha comentado, se utilizan dos. Uno de velocidad constante y otro ruidoso. Ambos son filtros EKF.
- Una matriz de transición por defecto, que es la mostrada en la ecuación Ec.5.2.1.
- La probabilidad de cada modo a priori. Por defecto, ambos modos tienen la misma probabilidad, luego 50%.

$$\Pi = \begin{pmatrix} 0.999 & 0.001 \\ 0.010 & 0.990 \end{pmatrix} \quad (\text{Ec.5.2.1})$$

*IMM. Matriz de transición utilizada*

Los filtros EnKF e PF cuentan con un parámetro que define el número de partículas a generar. Por defecto se ha configurado con 200 partículas.

Por último, el filtro UKF cuenta con los tres parámetros explicados en su sección 2.3.3.4. Se utilizan los valores que se indican por defecto para PDF gaussianas.

- $\alpha = 0.001$ , para el cálculo de alpha.
- $\beta = 2$ , conocimiento apriori de la PDF de  $x$ . (Gaussiana=2).
- $\kappa = 0$ , valor de escala. Suele ser 0

## 5.3 Análisis entre filtros

A continuación, se comparan los cinco filtros implementados entre sí, observando también los contactos medidos.

### 5.3.1 Escenario 1

En la figura Fig.5.5 se muestra la gráfica XY del escenario 1. Como vemos, todos los filtros siguen adecuadamente la posición de los contactos, siendo el UKF el filtro que está algo más lejos de los demás. Se observa claramente al final de la ejecución.

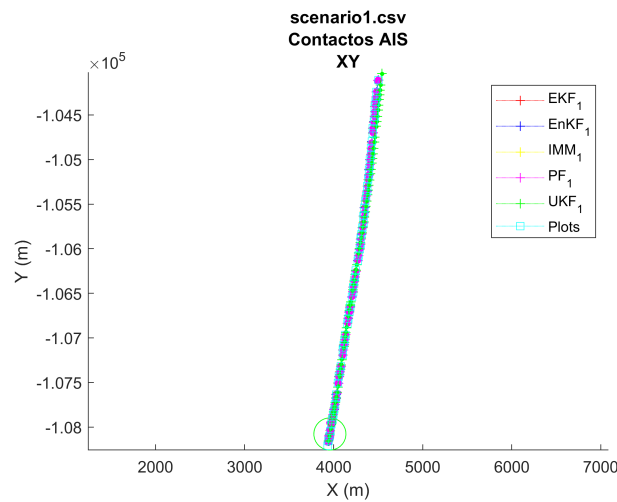


Fig.5.5: Escenario 1. Gráfico XY

Si observamos la predicción de la velocidad (figura Fig.5.6), comparada con la medida por el GPS del barco (y transformada desde nudos), vemos que el UKF inicializa para ambas velocidades a muy por encima de la velocidad real, y posteriormente se ajusta. En la velocidad de X, finalmente no es capaz de seguir correctamente la disminución de velocidad en dicho eje, haciéndolo de manera más progresiva.

Cabe destacar que el resto de filtros tienen un comportamiento similar, bastante ruidoso entre contactos. El funcionamiento de los filtros IMM y EKF es prácticamente idéntico, al no variar en ningún caso al modo ruidoso, siendo el por defecto el mismo que el utilizado para el EKF. Por último, destacar el filtro EnKF, cuya predicción de velocidad es extremadamente sensible a las variaciones reales. La variación en el eje Y parece un claro error del filtro.

Respecto a las métricas, como se observa en la figura Fig.5.7, el tiempo de cómputo de los filtros es bastante similar entre ellos, así como ínfimos en

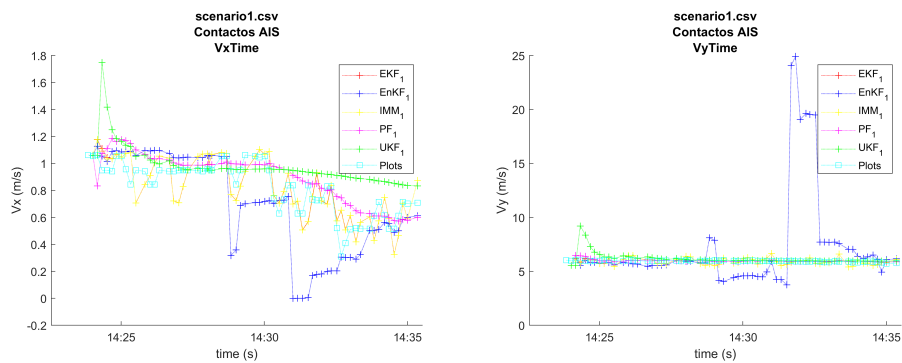


Fig.5.6: Escenario 1. Velocidad en cada eje

general, tardando cada filtro menos de una centésima de segundo en cada iteración. Decir que este tiempo incluye el cálculo de las métricas, lo cual podría considerarse fuera del filtro.

Se observan dos claras categorías lógicas en el tiempo. Los filtros más rápidos son el EKF y UKF, al utilizar el primero un único filtro, respecto a los dos que tiene el IMM, mientras que el UKF utiliza 13 sigma points en vez de 200 partículas que utilizan el EnKF y PF.

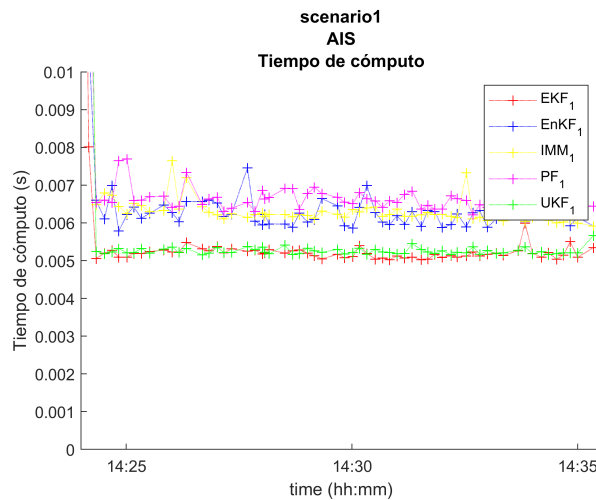


Fig.5.7: Escenario 1. Tiempo de cómputo

### 5.3.2 Escenario 2

En la figura Fig.5.8 se muestra la gráfica XY del escenario 2. En este escenario también siguen correctamente los filtros a los contactos, siendo el UKF el filtro que más se desvía otra vez, especialmente al final.

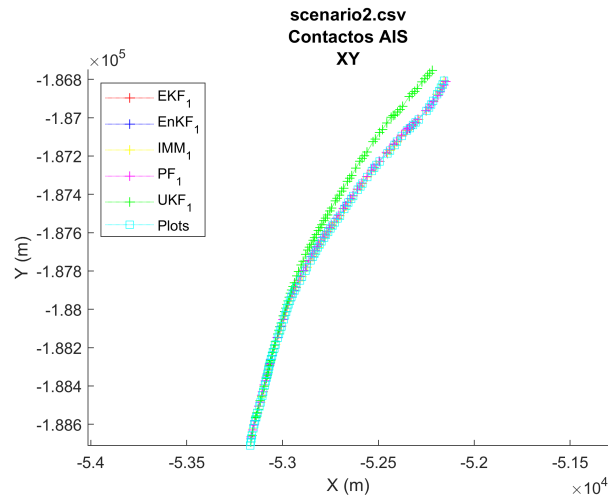


Fig.5.8: Escenario 2. Gráfico XY

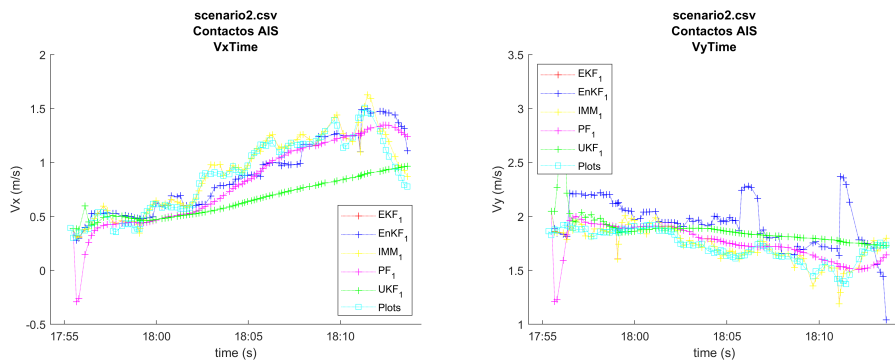


Fig.5.9: Escenario 2. Velocidad en cada eje

En este escenario también la predicción de la velocidad del UKF (figura Fig.5.9) es incapaz de seguir de manera adecuada la subida de velocidad.

Cabe destacar que el resto de filtros tienen un comportamiento similar al escenario previo, consiguiendo resultados prácticamente idénticos. La velocidad varía bastante en el tiempo, sobre todo en su componente Y con el filtro EnKF.

### 5.3.2.1 Análisis específico del UKF

Dado el mal comportamiento del filtro UKF al final de los dos escenarios analizados hasta ahora, se va a realizar un análisis específico sobre el comportamiento de este filtro. Inicialmente, se muestra en la figura Fig.5.10 toda la información del filtro (covarianzas, vector velocidad y puntos sigma).

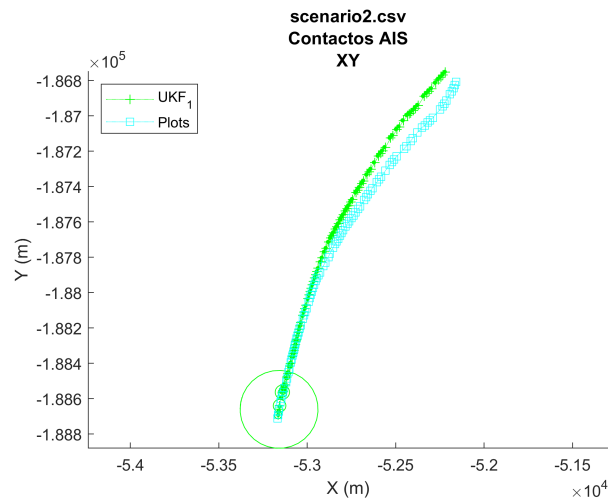


Fig.5.10: Escenario 2. UKF

Se detecta claramente que la matriz de covarianza (la elipse) se inicializa de manera muy elevada, y disminuye rápidamente a una escala que apenas se puede discernir. Se observa que en la zona final trata de corregir el giro, por lo que sí lo detecta, pero la matriz de covarianzas es tan reducida que la corrección realizada con la medida es insuficiente para el giro del barco.

Este es uno de los problemas de utilizar un filtro básico para un problema cambiante y complejo como este. El filtro en sí funciona correctamente pero no está ajustado al problema, puesto que la matriz de covarianzas disminuye constantemente en el tiempo, y la no linealidad del problema permite que en cualquier instante el barco gire, haciendo que el comportamiento, como se observa, no sea adecuado.

Cabe recordar que este filtro trataba de mejorar el rendimiento del EKF cuando la no linealidad era elevada, pero en este problema no se da esta casuística.

Para demostrar la comprensión del problema del filtro, una solución rápida es introducir una covarianza mínima, de manera que el filtro no pueda reducirla infinitamente. Si se realiza esta modificación, el funcionamiento general es más adecuado, como se ve en las figuras Fig.5.11. En los momentos que aumenta la covarianza, el filtro permite corregir la posición aproximándose en gran cantidad a la posición real.

De esta manera, si comparamos el RMSE de ambos filtrados (con y sin la covarianza mínima), observamos como el error de la predicción ha disminuido de manera significativa (figura Fig.5.12).

En cuanto al NIS, se observa como el filtro apenas cumple la consistencia



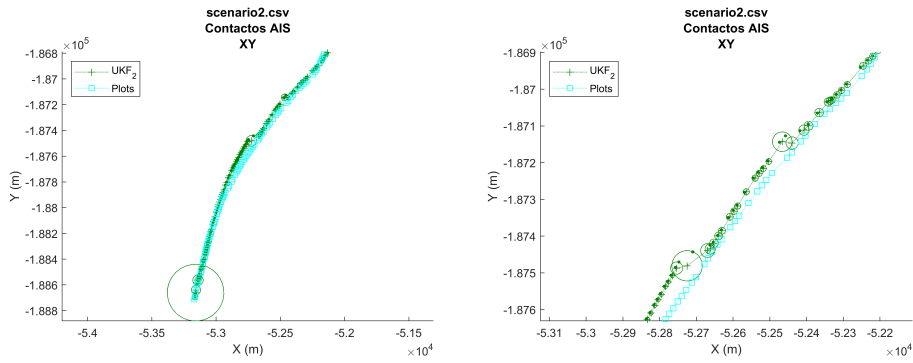


Fig.5.11: Escenario 2. UKF modificado

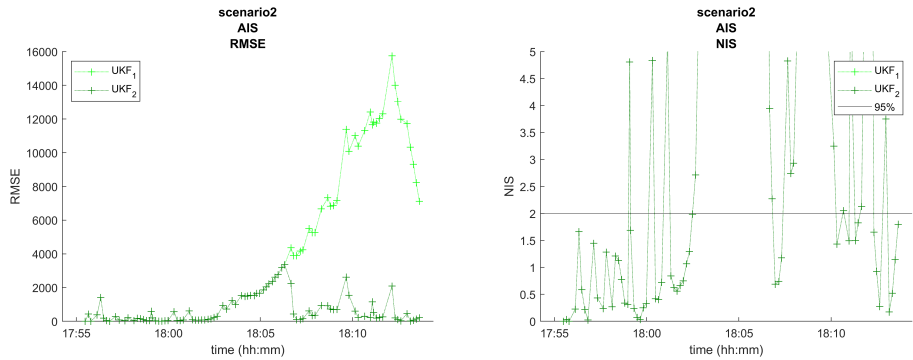


Fig.5.12: Escenario 2. UKF comparativa RMSE

con la hipótesis de un 95% de confianza. Esto nos indica que los resultados del filtro distan de ser fiables.

### 5.3.3 Escenario 3

En la figura Fig.5.13 se muestra la gráfica XY del escenario 3.

Como se ve, todos los filtros menos el PF tienen su comportamiento habitual. El UKF sin la modificación sigue proporcionando una estimación muy vaga. Se observan patrones de seguimiento pero muy tardíos. Los demás filtros siguen teniendo unos resultados muy correctos, que posteriormente se compararán en detalle.

#### 5.3.3.1 Análisis específico del PF

Si se observan las partículas del PF en cada iteración (figura Fig.5.14), centrándose en el comienzo donde se pierde, se ve como al propagar las partículas, estas, como el UKF tiende a converger. En este caso, son las partículas en vez de la covarianza, pero terminan representando lo mismo. En este, el primer filtro de partículas creado (y por ende el más sencillo),

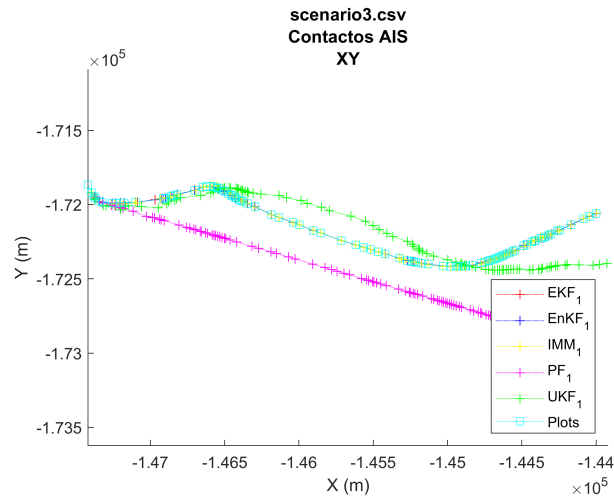


Fig.5.13: Escenario 3. Gráfico XY

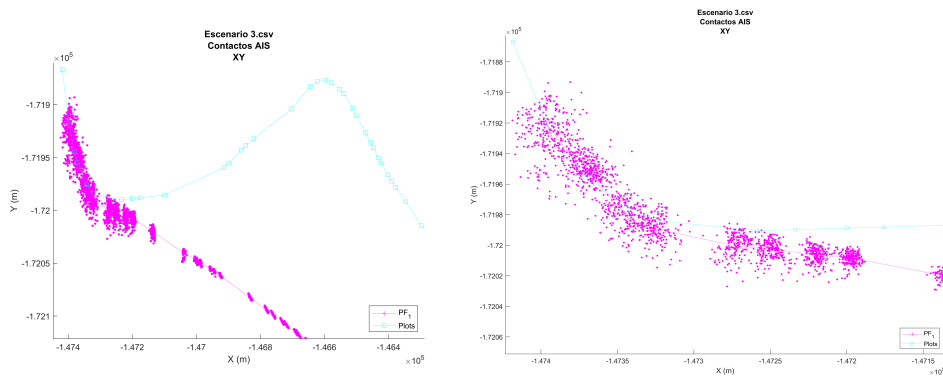


Fig.5.14: Escenario 3. Gráfico XY PF zoom

existen este tipo de problemas debido a la dependencia de las partículas y su estancamiento. Esta rama de filtros ha evolucionado mucho en los últimos años, proponiendo múltiples variantes que mejorarían el resultado obtenido por este.

### 5.3.3.2 Filtro IMM más ruidoso

Como se ha indicado, los filtros IMM y EKF producen unas predicciones claramente similares. Por ello, se procede a comparar su funcionamiento. El filtro IMM, como se ve en la figura Fig.5.15, realizan sus probabilidades de modo algunos cambios al comienzo de la traza.

Estas variaciones son producidas por las curvas que realiza el barco, ha-

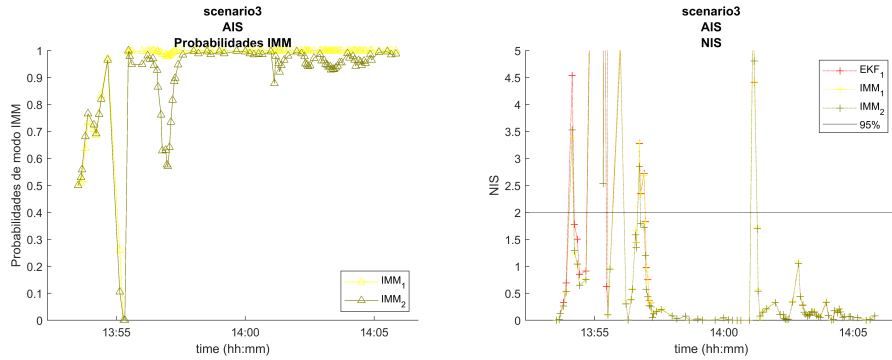


Fig.5.15: Escenario 3. Probabilidades de modo IMM y NIS

ciendo que por primera vez en todos los escenarios el filtro pase al modo 2, diferenciándose así ligeramente del filtro EKF.

Por ello, se prueba una nueva variante de filtro IMM con una nueva configuración ligeramente más ruidosa, para tratar de obtener una mejor adaptación a los giros del barco. En concreto, el cambio realizado es el uso de la matriz de transición con una mayor probabilidad de cambio (Ec.5.3.1).

$$\Pi = \begin{pmatrix} 0.95 & 0.05 \\ 0.10 & 0.90 \end{pmatrix} \quad (\text{Ec.5.3.1})$$

*IMM. Matriz de transición ruidosa*

En la figura de la izquierda se ve como el filtro más ruidoso cumple su función, transicionando en mayor medida al modo ruidoso. A pesar de ello, sus resultados son muy similares. El NIS se dispara durante los cinco primeros minutos en los tres estimadores. Esto nos indica que, a pesar de los modos y los filtros, los contactos permiten predecir con consistencia. Como se ve en la figura Fig.5.16, existe alrededor del minuto las 13:55 un contacto muy ruidoso y con un periodo muy amplio a diferencia del resto de la traza. Estos saltos son los que hacen aumentar drásticamente el ruido del filtro y errar en su predicción inicial.

### 5.3.4 Escenario 4

En la figura Fig.5.17 se muestra la gráfica XY del escenario 4.

Como vemos, todos los filtros siguen adecuadamente la posición de los contactos, menos el PF, que ante un giro con un ángulo mayor a lo que sus partículas son capaces de predecir sigue una trayectoria rectilínea uniforme, totalmente incorrecta. El UKF a su vez sigue esbozando la forma real de manera vaga.

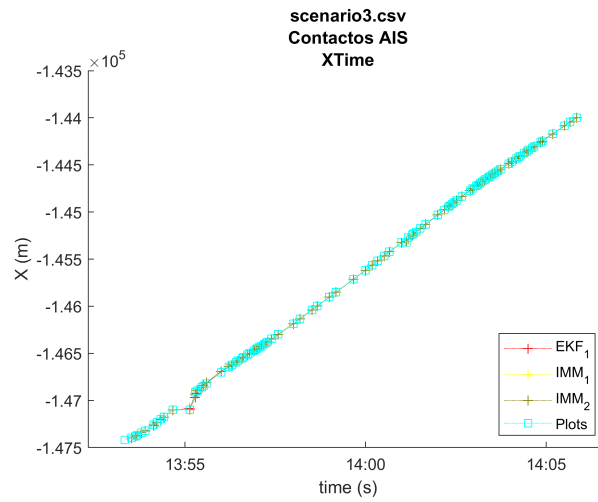


Fig.5.16: Escenario 4. Comparación IMM Gráfico XTime

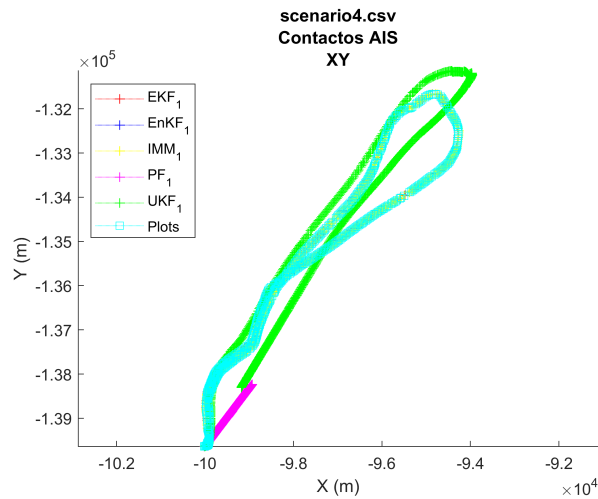


Fig.5.17: Escenario 4. Gráfico XY

Con el fin de analizar este escenario en profundidad, al ser tan grande tanto espacial como temporalmente, en la figura Fig.5.18 se amplía la gráfica en la zona de la curva más cerrada y en la figura Fig.5.19 se hace lo propio con la velocidad. Del gráfico se eliminan los dos filtros con pobres resultados al no aportar información.

Se observa que los resultados siguen siendo muy similares, por lo que las predicciones que realizan estos tres filtros son todas ellas muy correctas. En las figuras de la velocidad se observa lo habitual en los filtros hasta ahora. El EKF e IMM coinciden bastante en la velocidad, excepto en un instante en el que el IMM transita de modo (ver figura Fig.5.20).

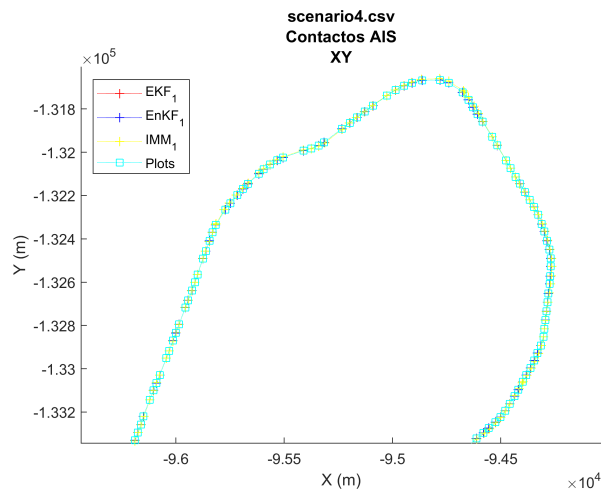


Fig.5.18: Escenario 4. Gráfico XY curva

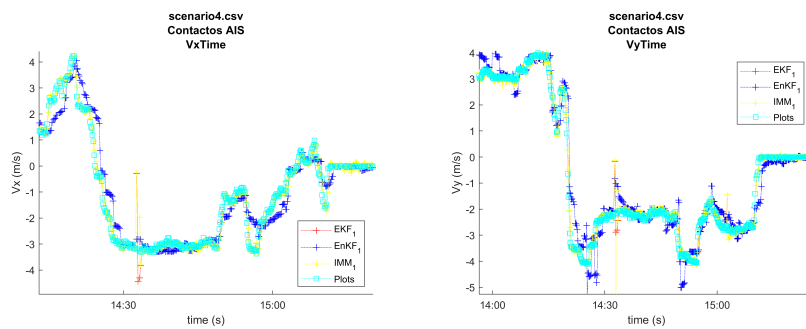


Fig.5.19: Escenario 4. Velocidad en cada eje ampliada

Esta única transición en ambas configuraciones del filtro IMM nos indican que permanecen constantemente en el modo no ruidoso, al no contar con un giro excesivamente drástico (recordar que el escenario dura más de dos horas).

Sin embargo, ahora el NIS sí se cumple la hipótesis de ser consistente con un 95% de confianza en la mayoría de instantes, luego el problema del escenario anterior era claramente debido a la mala frecuencia y ruido de los contactos.

### 5.3.5 Conclusión

Debido a las altas variaciones de velocidad del filtro EnKF y el hecho de que el filtro IMM tiene la capacidad de realizar la misma predicción que el EKF, pero a su vez de ajustarse a transiciones más ruidosas, se puede concluir que el filtro IMM es el mejor filtro para estos cuatro escenarios.

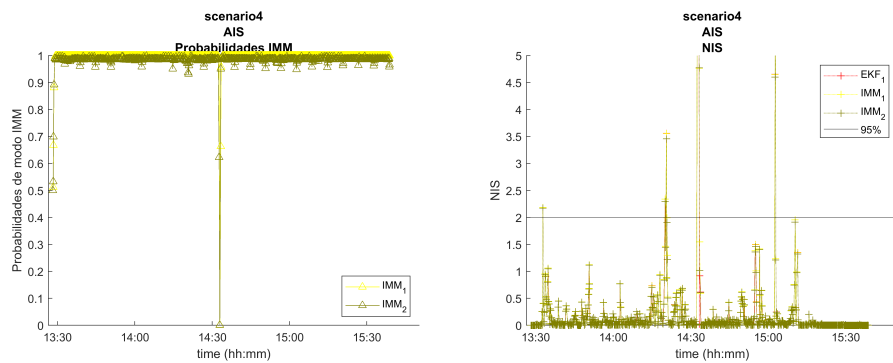


Fig.5.20: Escenario 4 Probabilidades de modo IMM y NIS

Esta conclusión no es sorprendente, al ser uno de los filtros más utilizados para estas tareas al permitir ajustarse perfectamente utilizando distintos modos.

Cabe destacar que el filtro IMM es el mejor utilizando en sus modos dos EKF, tras llevar desarrollándolo, testeándolo y optimizándolo durante varios meses. Será interesante en el futuro hacerlo funcionar con otro tipo de modos, ya sean del mismo filtro o de varios a la vez.

Respecto al resto de filtros, es muy posible que puedan ofrecer mejores resultados en el futuro, ajustándolos mejor al problema.

El filtro de partículas implementado actualmente tiene un funcionamiento correcto cuando el filtro no tiene demasiados giros, pero tiene tendencia a perderse, haciéndolo completamente inviable para cualquier causa.

En cuanto al filtro EnKF, este sí es capaz de funcionar de manera adecuada con el mismo número de partículas. Es sorprendente estos resultados, al no ser muy utilizado para realizar tracking de objetivos, sino más en entornos de predicción meteorológica. Es posible que los buenos resultados sean acordes al uso de la configuración del filtro EKF que sí está ajustado tras el trabajo con el IMM.

Por último, los resultados del filtro UKF han sido una decepción, puesto que en muchas investigaciones suele proporcionar buenos resultados. Es bastante probable que con una mejor configuración y ajustes sea capaz de ofrecer unos resultados acordes al resto de filtros.

## Capítulo 6

# Conclusiones y líneas futuras

En este último capítulo se realiza un resumen global de la tesis, comprobando el alcance de los objetivos marcados al comienzo del mismo (sección 6.2).

A su vez, también se introducen posibles vías de desarrollo, tanto paralelas como continuistas al framework realizado.

### 6.1 Conclusiones

Este trabajo se ha centrado en ofrecer una solución software sencilla para la comparación de la gran cantidad de filtros de estimación, enfocados al seguimiento de vehículos. Esta tarea ha sido completada con éxito, contando con un software potente capaz de integrar de manera sencilla cualquier filtro, únicamente fijando las entradas y salidas del mismo. De esta manera, cualquier filtro se beneficia de una abstracción sobre los contactos de los sensores y contando con una serie de métricas y gráficas integradas para su comparación y testeo en cualquier tipo de escenario.

#### 6.1.1 Cumplimiento de los objetivos

La tesis comenzó definiendo los siguientes puntos a cumplir:

- Analizar y comprender los distintos estimadores y filtros de posición más utilizados en la literatura y compararlos.
- Obtener una serie de contactos de sensores marítimos reales para obtener trazas sobre las que aplicar estimadores de posición.
- Desarrollar un framework de estimación que permita filtrar las trazas anteriormente extraídas.

- Desarrollar un componente del framework que genere gráficos y métricas de comparación entre estimaciones de una misma traza.
- Analizar los resultados obtenidos y concluir las fuerzas y debilidades de cada tipo de filtro para los escenarios propuestos.

Esta tesis comenzó de realizando un análisis puramente teórico, comprendiendo la gran dimensión de esta rama de investigación de tracking y fusión de datos y encontrando el lugar donde aportar con este trabajo. Tras comprender todos los componentes que precisa este tipo de sistemas tan complejos, en una de las áreas más importantes se realiza una aportación práctica concreta y tangible, que puede ser utilizada como base para proyectos por cualquier investigador.

El framework desarrollado proporciona el conjunto de filtros más famosos implementados, al estar basado en los principales artículos de la bibliografía. Tras realizar una importante comprensión de los mismos, quedar documentados en esta memoria y estableciendo una misma nomenclatura para todos ellos, han sido implementados siguiendo todos ellos el mismo procedimiento, facilitando así su comprensión. De esta manera, se cumple el primer objetivo definido en la introducción.

Con el fin de desarrollar un sistema que permita comparar los filtros analizados teóricamente, se diseñó y desarrolló un sistema altamente configurable y con capacidad de expansión que itere sobre una serie de contactos de entrada, sean del tipo de sensor que sean. Con este fin, a la par que se desarrollaba el sistema, se comenzó a desarrollar los cinco filtros analizados basándose en los pasos expuestos en esta memoria. De esta manera, se cumple el objetivo número tres.

En este instante, con el sistema de filtrado ya implementado se comenzó a testear su funcionamiento. Para ello se buscaron fuentes de datos, y se obtuvo un conjunto de contactos AIS. De ellos, se realizó un proceso para extraer de todo el conjunto una serie de trayectorias de interés y de las todas ellas se escogieron a mano varias de ellas. En este proceso se cumplió el objetivo número dos.

Una vez comprobado que los filtros funcionan correctamente con los cuatro escenarios, se comenzó a cumplir el siguiente objetivo. Tras buscar múltiples artículos sobre métricas de comparación de filtros, se incluyeron las más habituales dentro del framework y se explicó de manera teórica en esta memoria. También se realizó el desarrollo de una generación de gráficas a partir de cualquier tipo de información producida por el estimador, y también a partir de la información de los contactos utilizados. Con esto,



se permite realizar tanto análisis de un propio filtro, observando su funcionamiento de manera interna, como compararlo respecto a otros filtros.

Por último, una vez desarrollado todo el framework, utilizando únicamente herramientas del mismo, se acomete un análisis con los cinco filtros, y diversas variantes entre ellas sobre los cuatro escenarios. En este análisis se identifica el comportamiento del filtro y las causas de este para realizar dicho comportamiento.

Finalmente, se han cumplido todos los objetivos marcados para esta tesis, obteniendo unos resultados exitosos en todos ellos.

### **6.1.2 Conclusiones personales**

A nivel personal, este proyecto ha sido muy interesante. Conseguir desarrollar un sistema desde cero, que es capaz de realizar una comparación entre cualquier tipo de filtrado y analizarlos únicamente con esta herramienta es muy satisfactorio.

El mayor inconveniente de este tipo de proyectos, bajo mi opinión, es no poder fiarte de los avances que han hecho los demás investigadores. Esta herramienta sienta las bases para que un sistema, bajo las mismas condiciones, pueda ser evaluado contra el resto de filtros.

Por otro lado, todo el estado del arte escrito me ha aportado una comprensión muy elevada del trabajo realizado y sobre todo, de lo ínfimo que es en comparación con un sistema de fusión completo. A pesar de ello, es una de las labores más complejas, al ser prácticamente la base del sistema.

## **6.2 Líneas futuras**

Finalmente, esta tesis deja camino a varias líneas de trabajo futuras. De manera muy clara se observan dos ramas principales.

La primera de ellas se enfoca en potenciar el framework desarrollado, haciendo uso de él para futuros proyectos o ampliando las características de este con más cantidad. Con este enfoque, se definen los siguientes posibles trabajos:

- La opción más directa es utilizar el entorno tal y como está para tratar de optimizar los filtros implementados, ya sea en estos mismos escenarios de prueba, o en otros.
- También sería interesante tratar de probar estos estimadores con otra serie de sensores, puesto que para esta tesis únicamente se han probado contactos AIS.

- Como se ha comentado en la comparativa de estimadores, estos no son más que una muestra mínima de todas las aportaciones realizadas hasta el día de hoy. Adaptar nuevos filtros al framework, compararlos con los originales, modificar su configuración, etc. será necesario para estar al día con las novedades de esta rama de investigación.
- De igual forma, existen más métricas para comparar el funcionamiento de los filtros. Incluirlas, entenderlas y utilizarlas es una tarea muy interesante para ser capaz de detectar puntos de mejora de los estimadores.
- Lo mismo se puede realizar con las gráficas. Al igual que se han implementado comparaciones internas entre filtros, como las probabilidades de modo del filtro IMM, existen más variables en otros filtros cuyo análisis puede ser relevante para una estimación satisfactoria.

Por otro lado, se tiene la posibilidad de realizar trabajos paralelos en los que el framework juegue un papel secundario. Con este objetivo se definen las siguientes posibles ramas a seguir:

- Integrar este framework como subcomponente de un sistema de fusión de datos más amplio, o utilizar este para realizar esa ampliación. De esta manera, se pueden utilizar las características del framework para dar soporte al sistema completo.
- Utilizando el contexto extraído en niveles superiores de un sistema de fusión y las métricas de rendimiento de este trabajo, utilizarlo para variar la estimación de un blanco entre configuraciones, e incluso entre distintos filtros, para obtener mejores resultados en tiempo real.
- Desarrollar un simulador de generación de contactos que puedan servir para someter a los estimadores a escenarios más drásticos, pensados para evaluar unas características concretas.

# Bibliografía

- [1] E. P. Blasch and S. Plano, “JDL Level 5 Fusion Model “User Refinement” Issues and Applications in Group Tracking,” *Signal Processing, Sensor Fusion, and Target Recognition XI*, vol. 4729, no. July 2002, pp. 270–279, 2003.
- [2] D. L. Hall and J. Llinas, “An introduction to multisensor data fusion,” *Proceedings of the IEEE*, vol. 85, no. 1, pp. 6–23, 1997.
- [3] E. A. Wan and R. Van Der Merwe, “The unscented Kalman filter for nonlinear estimation,” *IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium, AS-SPCC 2000*, pp. 153–158, 2000.
- [4] B. Yan, N. Xu, W. B. Zhao, and L. P. Xu, “A three-dimensional hough transform-based track-before-detect technique for detecting extended targets in strong clutter backgrounds,” *Sensors (Switzerland)*, vol. 19, no. 4, 2019.
- [5] L. qiang Ni, S. sheng Gao, P. cheng Feng, and K. Zhao, “Rough Sets Probabilistic Data Association Algorithm and its Application in Multi-target Tracking,” *Defence Technology*, vol. 9, no. 4, pp. 208–216, 2013. [Online]. Available: <http://dx.doi.org/10.1016/j.dt.2013.11.004>
- [6] M. Galarnyk, “Explaining the 68-95-99.7 rule for a Normal Distribution,” 2018. [Online]. Available: <https://towardsdatascience.com/understanding-the-68-95-99-7-rule-for-a-normal-distribution-b7b7cbf760c2>
- [7] Q. Li and S. Y. Liang, “Degradation trend prediction for rotating machinery using long-range dependence and particle filter approach,” *Algorithms*, vol. 11, no. 7, 2018.
- [8] G. Margarit, J. A. B. Milanés, and A. Tabasco, “Operational ship monitoring system based on Synthetic Aperture Radar processing,” *Remote Sensing*, vol. 1, no. 3, pp. 375–392, 2009.
- [9] D. F. Crouse, “The tracker component library: free routines for rapid prototyping,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 32, no. 5, pp. 18–27, 2017.

- [10] D. L. Hall and J. Llinas, *Handbook of multisensor data fusion*, 2001, no. 1.
- [11] C. L. Bowman, "Data integration (fusion) tree paradigm," *Signal and Data Processing of Small Targets 1992*, vol. 1698, pp. 372–381, 1992.
- [12] R. C. Luo, Y. C. Chou, and O. Chen, "Multisensor fusion and integration: Algorithms, applications, and future research directions," *Proceedings of the 2007 IEEE International Conference on Mechatronics and Automation, ICMA 2007*, vol. 2, no. 2, pp. 1986–1991, 2007.
- [13] F. E. White, "JDL, Data Fusion Lexicon," *Technical Panel For C3*, vol. 15, no. 0704, p. 15, 1991.
- [14] H. F. Durrant-Whyte, "Sensor Models and Multisensor Integration," *The International Journal of Robotics Research*, vol. 7, no. 6, pp. 97–113, 1988. [Online]. Available: <http://dx.doi.org/10.1177/027836498800700608>
- [15] B. V. Dasarathy, "Sensor fusion potential exploitation-innovative architectures and illustrative applications," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 24–38, 1997.
- [16] J. Llinas, C. Bowman, G. Rogova, A. Steinberg, E. Waltz, and F. White, "Revisiting the JDL Data Fusion Model II," *Processing*, vol. 2, pp. 1218–1230, 2004. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.58.2996&rep=rep1&type=pdf>
- [17] A. Benaskeur, P. Mcguire, G. Liggins, R. W. Brennan, and P. Wojcik, "A distributed intelligent tactical sensor management system," *International Journal of Intelligent Control and Systems*, vol. 12, no. 2, pp. 97–106, 2007.
- [18] D. Smith and S. Singh, "Approaches to multisensor data fusion in target tracking: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 12, pp. 1696–1710, 2006.
- [19] Federico Castanedo, "A Review of Data Fusion Techniques," *The Scientific World Journal*, vol. 2013, p. 19, 2013.
- [20] P. C. Mahalanobis, "On the generalized distance in statistics," *Proceedings of the National Institute of Sciences (Calcutta)*, vol. 2, pp. 49–55, 1936.
- [21] H. Steinhaus, "Sur la division des corps materiels en parties," *Bulletin of the Polish Academy of Sciences*, vol. 4, no. 3, pp. 801–804, 1956. [Online]. Available: [http://www.laurent-duval.eu/Documents/Steinhaus{}\\_H{}\\_1956{}\\_j-bull-acad-polon-sci{}\\_division{}\\_cmp-k-means.pdf](http://www.laurent-duval.eu/Documents/Steinhaus{}_H{}_1956{}_j-bull-acad-polon-sci{}_division{}_cmp-k-means.pdf)

- [22] Y. Bar-Shalom and E. Tse, “Tracking in a cluttered environment with probabilistic data association,” *Automatica*, vol. 11, no. 5, pp. 451–460, 1975.
- [23] Y. Bar-Shalom, “Extension of the probabilistic data association filter to multi-target environment,” *Proceedings of the 5th Symposium on Non-linear Estimation*, pp. 16–21, 1974.
- [24] D. B. Raid, “An algorithm for tracking multiple targets,” *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979. [Online]. Available: <http://ieeexplore.ieee.org/document/1102177/>
- [25] H. W. Kuhn, “The Hungarian method for the assignment problem,” *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art*, 1955.
- [26] J. Munkres, “Algorithms for the Assignment and Transportation Problems,” *Journal of the Society for Industrial and Applied Mathematics*, 1957.
- [27] M. Leclerc, R. Tharmarasa, M. C. Florea, A. C. Boury-Brisset, T. Kirubarajan, and N. Duclos-Hindié, “Ship Classification Using Deep Learning Techniques for Maritime Target Tracking,” *2018 21st International Conference on Information Fusion, FUSION 2018*, no. October, pp. 737–744, 2018.
- [28] D. Amigo, D. Sánchez Pedroche, J. García, and J. M. Molina, “AIS trajectory classification based on IMM data,” *International Conference on Information Fusion (FUSION)*, 2019.
- [29] A. C. Boury-Brisset, “Ontology-based approach for information fusion,” *Proceedings of the 6th International Conference on Information Fusion, FUSION 2003*, vol. 1, no. 418, pp. 522–529, 2003.
- [30] J. Gomez-Romero, M. Patricio, J. García, and J. Molina, “Ontological representation of context knowledge for visual data fusion,” *Information Fusion (FUSION)*, 2009, 2009.
- [31] H. H. Afshari, S. A. Gadsden, and S. Habibi, “Gaussian filters for parameter and state estimation: A general review of theory and recent trends,” *Signal Processing*, vol. 135, no. January 2017, pp. 218–238, 2017. [Online]. Available: <http://dx.doi.org/10.1016/j.sigpro.2017.01.001>
- [32] É. Bossé, P. Valin, A. C. Boury-Brisset, and D. Grenier, “Exploitation of a priori knowledge for information fusion,” *Information Fusion*, vol. 7, no. 2, pp. 161–175, 2006.

- [33] P. Borkowski, “The ship movement trajectory prediction algorithm using navigational data fusion,” *Sensors (Switzerland)*, vol. 17, no. 6, 2017.
- [34] M. Mauthner, R. Altendorfer, W. Elmenreich, and A. Kirchner, “Optimization of Sensor, Bus, and Fusion Schedules of a Time-Triggered Sensor Fusion System,” *IEEE Intelligent Vehicles Symposium*, pp. 570–575, 2007.
- [35] G. Casella and R. L. Berger, *Statistical Inference*, second ed. ed. Duxbury, 2001.
- [36] S. Metropolis, Nicholas; Ulam, “The Monte Carlo Method,” *JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION*, 1949.
- [37] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” *Journal of Basic Engineering*, vol. 82, no. 1, p. 35, 2011. [Online]. Available: <http://www.unitedthc.com/DSP/Kalman1960.pdf>
- [38] S. G. Mohinder and P. A. Angus, “Applications of Kalman Filtering in Aerospace,” pp. 69–78, 2010. [Online]. Available: <http://ieeecss.org/CSM/library/2010/june10/11-HistoricalPerspectives.pdf><http://www.ieeecss.org/CSM/library/2010/june10/11-HistoricalPerspectives.pdf>
- [39] G. L. Smith, S. F. Schmidt, L. A. Mcgee, and M. Field, “Application of Statistical Filter Theory to the Optimal Estimation of Position and Velocity on Board a Circumlunar Vehicle,” *NASA Ames Research Center*.
- [40] B. A. Mcelhoe, “an Assessment of the Navigation and Course Corrections for a manned flyby of Mars or Venus,” *Supplement to IEEE Transactions on Aerospace and Electronic System*, vol. AES-2, no. 4, pp. 613–623, 1966.
- [41] L. A. McGee and S. F. Schmidt, “Discovery of the Kalman Filter as a Practical Tool for Aerospace and Industry,” *NASA Technical Memorandum*, no. November, p. 21, 1985. [Online]. Available: [http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19860003843\\_{\\_}1986003843.pdf](http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19860003843_{_}1986003843.pdf)
- [42] S. J. Julier and J. K. Uhlmann, “New extension of the Kalman filter to nonlinear systems,” p. 182, 1997. [Online]. Available: <http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.280797>
- [43] J. K. Uhlmann, “Simultaneous map building and localization for real time applications,” Ph.D. dissertation, University of Oxford, 1994.
- [44] G. Evensen, “Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error

- statistics,” *Journal of Geophysical Research*, vol. 99, no. C5, p. 10143, 1994. [Online]. Available: <http://doi.wiley.com/10.1029/94JC00572>
- [45] —, “The Ensemble Kalman Filter: Theoretical formulation and practical implementation,” *Ocean Dynamics*, vol. 53, no. 4, pp. 343–367, 2003.
- [46] M. Roth, G. Hendeby, C. Fritsche, and F. Gustafsson, “The Ensemble Kalman filter: a signal processing perspective,” *Eurasip Journal on Advances in Signal Processing*, vol. 2017, no. 1, pp. 1–35, 2017.
- [47] H. Blom, “An Efficient Decision-Making-Free Filter for Processes with Abrupt Changes,” *IFAC Proceedings Volumes*, vol. 18, no. 5, pp. 631–636, 1985. [Online]. Available: [http://dx.doi.org/10.1016/S1474-6670\(17\)60631-1](http://dx.doi.org/10.1016/S1474-6670(17)60631-1)
- [48] E. Mazor, A. Averbuch, and J. Dayan, “Interacting Multiple Model Methods in Target Tracking: A Survey,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 1, pp. 1–35, 2006.
- [49] N. Gordon, D. Salmond, and A. Smith, “Novel approach to nonlinear/non-Gaussian Bayesian state estimation,” *IEE Proceedings F Radar and Signal Processing*, vol. 140, no. 2, p. 107, 1993.
- [50] P. Del Moral, “Non-Linear Filtering: Interacting Particle Solution,” *Markov Processes Related Fields*, vol. 2, pp. 555–580, 1996.
- [51] D. BLOISI and L. IOCCHI, “Argos — a Video Surveillance System for Boat Traffic Monitoring in Venice,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 07, pp. 1477–1502, 2009.
- [52] Y. Zhang, Q. Z. Li, and F. N. Zang, “Ship detection for visual maritime surveillance from non-stationary platforms,” *Ocean Engineering*, vol. 141, no. April, pp. 53–63, 2017. [Online]. Available: <http://dx.doi.org/10.1016/j.oceaneng.2017.06.022>
- [53] D. K. Prasad, C. K. Prasath, D. Rajan, L. Rachmawati, E. Rajabaly, and C. Quek, “Challenges in video based object detection in maritime scenario using computer vision,” 2016. [Online]. Available: <http://arxiv.org/abs/1608.01079>
- [54] IMO, “Regulation 19 of SOLAS chapter V - Carriage requirements for shipborne navigational systems and equipment,” 2000. [Online]. Available: <http://solasv.mca.gov.uk/regulations/regulation19.htm>

- [55] E. Tu, G. Zhang, L. Rachmawati, E. Rajabally, and G. B. Huang, “Exploiting AIS Data for Intelligent Maritime Navigation: A Comprehensive Survey from Data to Methodology,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 5, pp. 1559–1582, 2018.
- [56] Danish Maritime Authority, “AIS data.” [Online]. Available: <https://www.dma.dk/SikkerhedTilSoes/Sejladsinformation/AIS/Sider/default.aspx{#}>
- [57] Ó. F. Eriksson, “Expert working group for mutual exchange and deliveries of AIS data,” Danish Maritime Authority, Tech. Rep. July, 2013. [Online]. Available: [https://portal.helcom.fi/Archive/SharedDocuments/AISEWG24-2013{\\_\\_}Presentation-1AISstatusinDenmark.pdf](https://portal.helcom.fi/Archive/SharedDocuments/AISEWG24-2013{__}Presentation-1AISstatusinDenmark.pdf)