



Universidad Carlos III de Madrid
Escuela Politécnica Superior

Desarrollo de sistemas de fusión de datos para seguimiento 3D

Trabajo de Fin de Máster
Máster Universitario en Ingeniería Informática
2018 - 2019

Por
Daniel Amigo Herrero

Supervisado por
Jesús García Herrero

26 de septiembre 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 el auge de vehículos autónomos, aún más.

Las técnicas de fusión de datos multisensor mejoran la precisión del seguimiento de los vehículos, mediante la combinación de información de varios sensores sobre un mismo objetivo. Ningún sensor es completamente preciso, cuentan con un ligero error, denominado ruido, que hace que su información no sea completamente fiable. El principal objetivo de la fusión de datos es realizar la estimación más precisa del entorno para utilizar dicha información en una toma de decisiones mejor razonada.

Los sistemas de fusión de datos son estructuras muy complejas resultado de la intercomunicación de varios componentes. Existen una gran cantidad de algoritmos propuestos para cada componente, cada uno con sus ventajas y desventajas en distintos casos. El mejor modo para conocer la mejor configuración de un sistema de fusión es someterlo, en el mismo entorno, a varios algoritmos y determinar cuál es el más adecuado.

En esta tesis, tras analizar todo el estado del arte de esta rama, se desarrolla un proyecto de ingeniería para la creación de dos herramientas interactivas con la que facilitar la comparación de sistemas de fusión de datos multisensor. La primera de ellas es la encargada de fijar el entorno en el que probar el sistema, y la segunda, la encargada de realizar las distintas pruebas sobre el entorno y comprobar cuál es la mejor configuración.

Las herramientas se desarrollan con éxito cumpliendo la planificación marcada inicialmente. El proyecto abre una línea futura para ampliar la herramienta con más algoritmos de más componentes de fusión de datos, para hacerla más completa.

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.

Multisensor data fusion techniques improve vehicle tracking accuracy, by combining information from several sensors on the same objective. Not a single sensor is completely accurate, all have a slight error, called noise, which makes their information not completely reliable. The main objective of data fusion is to provide the most accurate estimation of the environment to use this information in a better reasoned decision making.

Data fusion systems are very complex structures resulting from the intercommunication of several components. There are a large number of algorithms for proposals for each component, each with its advantages and disadvantages in different cases. The best way to know the best configuration of a fusion system is to submit it, in the same environment, to several algorithms and determine which one is the most appropriate.

In this thesis, after analyzing all the state of the art of this branch, an engineering project is developed for the creation of two interactive applications, developed to make easier the comparison of multisensor data fusion systems. The first application is responsible for the environment, generating the contacts of the sensors to test the system. The second application is responsible for carrying out the different tests in the environment and checking which is the best configuration.

Both tools are developed satisfactorily fulfilling the initial project planning. The project opens a future line to expand the tool with more algorithms of more data fusion components, to make it more complete.

Índice de contenidos

1	Introducción	1
1.1	Motivación	1
1.2	Objetivos	4
1.3	Estructura del documento	5
2	Estado del arte	6
2.1	Sistemas de fusión de datos multisensor	6
2.1.1	Clasificación de sistemas de fusión de datos	7
2.1.2	Principales componentes sistema de fusión	11
2.2	Sensores en el ámbito aéreo	22
2.2.1	Sistemas de navegación	23
2.2.2	Radar primario	24
2.2.3	Radar secundario	26
2.2.4	ADS	28
2.2.5	Multilaterización	30
2.2.6	IRST	31
2.2.7	Comparativa	32
2.3	Soluciones alternativas existentes	33
2.3.1	Software para generar simulaciones	33
2.3.2	Software para probar sistemas de fusión	34
3	Gestión del proyecto	37
3.1	Metodología de trabajo	37
3.2	Organización del proyecto	38
3.2.1	Recursos humanos	38
3.2.2	Recursos hardware	39
3.2.3	Recursos software	40
3.2.4	Visual Studio Code + Latex	40
3.2.5	Microsoft Visio	40
3.2.6	WireframePro	40
3.2.7	MATLAB + App Designer	41
3.2.8	TimeCamp + Trello	41
3.2.9	Microsoft Project	41
3.3	Planificación del proyecto	41

3.3.1	Bloques de la planificación	42
3.4	Presupuesto del proyecto	43
4	Alcance y usabilidad del software	48
4.1	Alcance del software	48
4.2	Artefactos persona	49
4.2.1	Personas	49
4.3	Escenarios	51
4.4	Wireframes	52
4.4.1	Wireframe generador de contactos	52
4.4.2	Wireframe sistema de fusión	54
4.5	Validación mediante entrevistas	54
4.6	Identificación de requisitos	56
4.7	Plan de riesgos	57
4.7.1	Riesgos encontrados	57
4.8	Marco regulador	61
5	Diseño del sistema	63
5.1	Introducción	63
5.1.1	Introducción MATLAB App Designer	63
5.1.2	Componentes implementados en ambas aplicaciones	64
5.2	Generador de contactos	67
5.2.1	Sensores implementados	67
5.2.2	Funcionamiento de la aplicación	73
5.2.3	Algoritmia de la aplicación	76
5.3	Sistema de fusión	78
5.3.1	Algoritmos implementados	78
5.3.2	Métricas de evaluación implementadas	78
5.3.3	Funcionamiento de la aplicación	79
5.3.4	Algoritmia de la aplicación	83
6	Pruebas del sistema	86
6.1	Introducción	86
6.2	Comprobación escenarios	86
6.2.1	Comprobación del escenario 1	86
6.2.2	Comprobación del escenario 2	87
6.2.3	Comprobación del escenario 3	87
6.2.4	Comprobación del escenario 4	87
7	Conclusiones y líneas futuras	88
7.1	Conclusiones	88
7.1.1	Cumplimiento de los objetivos	88
7.1.2	Conclusiones personales	90
7.2	Comparación tiempos de trabajo	91

7.3	Líneas futuras de investigación	92
Anexos		94
A	Entrevistas	95
A.1	Estructura de las entrevistas	95
A.2	Preguntas de las entrevistas	96
A.2.1	Introducción	96
A.2.2	Calentamiento	96
A.2.3	Fase principal	97
A.2.4	Fase de descanso	98
A.2.5	Fase final	98
A.3	Transcripción de las entrevistas	98
A.3.1	Transcripción de la entrevista 1	98
A.3.2	Transcripción de la entrevista 2	103
B	Requisitos	108
B.1	Requisitos funcionales	108
B.1.1	Requisitos del simulador	108
B.1.2	Requisitos de la app de fusión de datos	113
B.2	Requisitos de datos	118
B.3	Requisitos de ambiente	119
B.4	Requisitos de usuario	120
B.5	Requisitos de usabilidad	121
Bibliografía		122

Índice de figuras

Fig.1.1	Escenario aéreo sin niebla	3
Fig.1.2	Escenario aéreo con niebla	4
Fig.2.1	Modelo DFD [1]	8
Fig.2.2	Clasificación arquitectura sistemas fusión	10
Fig.2.3	Modelo del proceso de fusión JDL-DFIG [1]	11
Fig.2.4	Ejemplo filtro de estimación	13
Fig.2.5	Clasificación filtros de estimación	14
Fig.2.6	Funcionamiento Filtro de Kalman (KF)	15
Fig.2.7	Comparativa filtros de estimación	17
Fig.2.8	Ejemplo contactos reales sin identificador (b) filtrados y asociados en 8 trazas (a) [2]	18
Fig.2.9	Rotación avión	24
Fig.2.10	Radar de superficie	26
Fig.2.11	Radar primario (abajo) y secundario (encima)	27
Fig.2.12	Ejemplo MLAT [3]	31
Fig.2.13	Implementación de un IRST	32
Fig.3.1	Planificación del proyecto	42
Fig.4.1	Wireframe 1 generador de contactos	53
Fig.4.2	Wireframe mapa auxiliar	53
Fig.4.3	Wireframe 2 generador de contactos	54
Fig.4.4	Wireframe 1 sistema de fusión	55
Fig.4.5	Wireframe 2 sistema de fusión	55
Fig.5.1	Desarrollando interfaz a mano	64
Fig.5.2	Mensaje de progreso en la interfaz	65
Fig.5.3	Mensaje de error en la interfaz	66
Fig.5.4	Ejemplo extracto de fichero exportado	67
Fig.5.5	Ejemplo de tooltips	68
Fig.5.6	Ejemplo del componente del mapa	68
Fig.5.7	Ejemplo de información adicional en el mapa	68
Fig.5.8	Selección del tipo de escenario	73
Fig.5.9	Nuevo punto de paso	74

Fig.5.10	Uso de distintos sensores	75
Fig.5.11	Nuevo punto de paso	75
Fig.5.12	Algoritmo generador de contactos	76
Fig.5.13	Vista de prueba. Estimación	79
Fig.5.14	Listado de resultados para evaluar	81
Fig.5.15	Ejemplo de asociación en la herramienta. En rojo, verde y azul, las trazas locales de varios sensores del blanco número 3, y en negro su traza global	81
Fig.5.16	Ejemplo de métrica en la herramienta. Comparativa de tiempos de cómputo de la estimación del mismo blanco, con filtros de estimación distintos	82
Fig.5.17	Vista de prueba. Estimación	83
Fig.5.18	Algoritmo fusión de datos multisensor	84
Fig.5.19	Ejemplo de tabla de asociación	84
Fig.7.1	Ejemplo entrada TimeCamp	91

Índice de tablas

Tab.2.1	Técnicas fusión de datos según nivel JDL	12
Tab.2.2	Técnicas fusión de datos según nivel JDL	33
Tab.3.1	Currículo empleado	39
Tab.3.2	Componentes ordenador sobremesa 1	39
Tab.3.3	Componentes ordenador sobremesa 2	39
Tab.3.4	Componentes portátil	40
Tab.3.5	Tareas de cada rol	45
Tab.3.6	Sueldo por rol	45
Tab.3.7	Amortización por recurso hardware	46
Tab.3.8	Coste por recurso software	46
Tab.3.9	Costes indirectos	47
Tab.3.10	Presupuesto final	47
Tab.4.1	Persona 1	50
Tab.4.2	Persona 2	50
Tab.4.3	Persona 3	51
Tab.4.4	Plantilla requisitos	56
Tab.4.5	Plantilla definición de riesgos	57
Tab.4.6	Riesgo 1	58
Tab.4.7	Riesgo 2	58
Tab.4.8	Riesgo 3	59
Tab.4.9	Riesgo 4	59
Tab.4.10	Riesgo 5	60
Tab.4.11	Riesgo 6	60
Tab.4.12	Riesgo 7	61
Tab.4.13	Riesgo 8	61
Tab.5.1	Parámetros del sensor RADAR	69
Tab.5.2	Parámetros del sensor RADAR3D	70
Tab.5.3	Parámetros del sensor SSR	70
Tab.5.4	Parámetros del sensor IFF	71
Tab.5.5	Parámetros del sensor IRST	72
Tab.5.6	Parámetros del sensor MLAT	72
Tab.5.7	Parámetros del sensor ADS-B	72

Tab.2.1	Requisito funcional 1	108
Tab.2.2	Requisito funcional 2	108
Tab.2.3	Requisito funcional 3	109
Tab.2.4	Requisito funcional 4	109
Tab.2.5	Requisito funcional 5	109
Tab.2.6	Requisito funcional 6	110
Tab.2.7	Requisito funcional 7	110
Tab.2.8	Requisito funcional 8	110
Tab.2.9	Requisito funcional 9	111
Tab.2.10	Requisito funcional 10	111
Tab.2.11	Requisito funcional 11	111
Tab.2.12	Requisito funcional 12	111
Tab.2.13	Requisito funcional 13	112
Tab.2.14	Requisito funcional 14	112
Tab.2.15	Requisito funcional 15	112
Tab.2.16	Requisito funcional 16	113
Tab.2.17	Requisito funcional 17	113
Tab.2.18	Requisito funcional 18	113
Tab.2.19	Requisito funcional 19	114
Tab.2.20	Requisito funcional 20	114
Tab.2.21	Requisito funcional 21	114
Tab.2.22	Requisito funcional 22	114
Tab.2.23	Requisito funcional 23	115
Tab.2.24	Requisito funcional 24	115
Tab.2.25	Requisito funcional 25	115
Tab.2.26	Requisito funcional 26	116
Tab.2.27	Requisito funcional 27	116
Tab.2.28	Requisito funcional 28	116
Tab.2.29	Requisito funcional 29	117
Tab.2.30	Requisito funcional 30	117
Tab.2.31	Requisito funcional 31	117
Tab.2.32	Requisito funcional 32	117
Tab.2.33	Requisito funcional 33	118
Tab.2.34	Requisito funcional 34	118
Tab.2.35	Requisito de datos 1	118
Tab.2.36	Requisito de datos 2	119
Tab.2.37	Requisito de datos 3	119
Tab.2.38	Requisito de datos 4	119
Tab.2.39	Requisito de ambiente 1	119
Tab.2.40	Requisito de ambiente 2	120
Tab.2.41	Requisito de ambiente 3	120
Tab.2.42	Requisito de usuario 1	120
Tab.2.43	Requisito de usuario 2	121
Tab.2.44	Requisito de usabilidad 1	121

Tab.2.45 Requisito de usabilidad 2 121
Tab.2.46 Requisito de usabilidad 3 121

Acrónimos

A continuación, se exponen los acrónimos utilizados a lo largo del documento, explicando su significado.

ADS (Automatic Dependant Surveillance) Sistema automático dependiente de vigilancia Primer uso en pág. 28

ADS-B (Automatic Dependant Surveillance Broadcast) Sistema automático dependiente de vigilancia - difusión Primer uso en pág. ix, 29, 72

ADS-C (Automatic Dependant Surveillance Contract) Sistema automático dependiente de vigilancia - contrato Primer uso en pág. 28, 29

ANN (Artificial Neural Network) Redes de neuronas artificiales Primer uso en pág. 19

CCOO (Comisiones Obreras) Comisiones Obreras Primer uso en pág. 45

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. 8

DEI-DEO (Decision in - Decision out) Entrada de decisión - Salida de decisión Primer uso en pág. 8

DFD (Data Feature Decision) Datos Característica Decisión Primer uso en pág. 7

DGPS (Differential Global Positioning System) Sistema de Posicionamiento Global Diferencial Primer uso en pág. 24

DT (Decision Tree) Árbol de decisión Primer uso en pág. 19

EKF (Extended Kalman Filter) Filtro de Kalman Extendido Primer uso en pág. 15, 16, 78, 116

EnKF (Ensemble Kalman Filter) Filtro de Kalman Ensemble Primer uso en pág. 16, 78, 116

FEI-DEO (Feature in - Decision out) Entrada de característica - Salida de decisión Primer uso en pág. 8

- FEI-FEO (Feature in - Feature out)** Entrada de característica - Salida de característica Primer uso en pág. 8
- GDPR (General Data Protection Regulation)** Reglamento General Europeo de Protección de Datos Primer uso en pág. 61
- GNSS (Global Navigation Satellite System)** Sistema global de navegación por satélite Primer uso en pág. 23, 28
- GPS (Global Positioning System)** Sistema de Posicionamiento Global Primer uso en pág. 23, 30, 77
- HCI (Human Computer-Interaction)** Interacción humano-computador Primer uso en pág. 21
- IFF (Identification Friend or Foe)** Identificador amigo-enemigo Primer uso en pág. ix, 26, 71
- IMM (Interacting Multiple Modes)** Filtro de Múltiples modos inter-operantes Primer uso en pág. 16, 78, 116
- IMU (Inertial Measurement Unit)** Unidad de medición Inercial Primer uso en pág. 24
- INS (Inertial Navigation System)** Sistema de navegación inercial Primer uso en pág. 24
- IRST (InfraRed Search and Track)** Búsqueda y seguimiento por infrarrojos Primer uso en pág. ix, 71, 72
- ISIF (International Society of Information Fusion)** Sociedad internacional de fusión de información Primer uso en pág. 35
- ISO (International Organization for Standardization)** Organización Internacional de Normalización Primer uso en pág. 57
- JDL (Joint Directors of Laboratories)** Unión de directores de laboratorios Primer uso en pág. 6, 10, 11, 19, 82
- KF (Kalman Filter)** Filtro de Kalman Primer uso en pág. 14, 15
- MHT (Multi-Hypothesis Tracking)** Seguimiento con Hipótesis Múltiples Primer uso en pág. 19, 78, 116
- MLAT (Multilateration)** Multilateración Primer uso en pág. ix, 30, 72
- MoE (Measures of Effectiveness)** Medidas de Efectividad Primer uso en pág. 20

- MoP (Measures of Performance)** Medida del rendimiento Primer uso en pág. 20
- MSSR (Monopulse Secondary Surveillance Radar)** Radar secundario monopulso Primer uso en pág. 28
- NEES (Normalized Estimation Error Squared)** Error cuadrático de estimación normalizado Primer uso en pág. 22, 79, 117
- NIS (Normalized Innovation Squared)** Innovación cuadrática normalizada Primer uso en pág. 22, 79, 117
- NN (Nearest Neighbour)** Vecino más próximo Primer uso en pág. 17, 78, 116
- PDA (Probabilistic Data Association)** Asociación Probabilística de Datos Primer uso en pág. 18, 78, 116
- PDF (Probability Density Function)** Función Densidad de Probabilidad Primer uso en pág. 22
- PF (Particle Filter)** Filtro de partículas Primer uso en pág. 16, 78, 116, 117
- PSR (Secondary Surveillance Radar)** Radar primario Primer uso en pág. 24–26, 28
- PVP (Precio de Venta al Público)** Precio de Venta al Público Primer uso en pág. 45
- RMSE (Root Mean Square Error)** Raíz del error cuadrático medio Primer uso en pág. 21, 79, 117
- SBAS (Satellite-Based Augmentation System)** Sistema de aumentación basado en satélites Primer uso en pág. 24
- SLAMEM (Simulation of the Locations and Attack of Mobile Enemy Missiles)** Simulador de posición y ataque de misiles enemigos Primer uso en pág. 33
- SMR (Surface Movement Radar)** Radar de superficie Primer uso en pág. 25
- SSR (Secondary Surveillance Radar)** Radar secundario Primer uso en pág. ix, 26, 28, 30, 70
- TDOA (Time Difference Of Arrival)** Diferencia de tiempo de llegada Primer uso en pág. 30

TOA (Time Of Arrival) Tiempo de llegada Primer uso en pág. 23, 30

TOT (Time Of Transmission) Tiempo de transmisión Primer uso en pág. 23

UKF (Unscented Kalman Filter) Filtro de Kalman “Unscented” Primer uso en pág. 15, 16, 78, 116

WAM (Wide-Area Multilateration) Multilateración en áreas amplias Primer uso en pág. 30

Capítulo 1

Introducción

En este primer capítulo de la tesis se introduce, en la sección 1.1, la motivación que ha conllevado el desarrollo de este proyecto de ingeniería. A continuación, la sección 1.2 enumera los objetivos principales a cumplir en este trabajo. Finalmente, en la sección 1.3, se introduce una breve estructura sobre los siguientes capítulos.

1.1 Motivación

Todos ser humano busca validar cada acción que realiza en cada momento. Dicha validación, de manera consciente o no, se realiza utilizando información. Dicho concepto, denominado toma de decisiones de manera informada, es una característica inherente al ser humano. Este busca utilizar toda la información disponible para evaluar todas las opciones existentes en dicho escenario y ejecutar la más adecuada según los intereses del humano. Desde la prehistoria se puede observar este comportamiento, cuando por ejemplo, al saber que tenemos que comer, desarrollamos lanzas para facilitar la caza para facilitar el combate.

Las acciones que tomar se han ido complicando con el paso de los años, hasta llegar a la actualidad, donde la información disponible a tener en cuenta para tomar la mejor acción posible es prácticamente infinita, y la acción puede afectar en muchos aspectos durante mucho tiempo. Para entender la velocidad a la que avanza la cantidad de información que se genera, IBM pronunció que “el 90% de la información existente a día de hoy, se había generado en los dos últimos años” [4].

Es por ello que actualmente nos apoyamos en máquinas y tecnologías como Big Data para facilitar la toma de decisiones de manera informada. Gracias a ellas, los seres humanos pueden obtener información útil para esta toma de decisiones sin tener que comprenderla, puesto que el procesamiento de cantidades de datos inmensas lo realiza la máquina. Además, incluso son capaces de concluir cuál es la mejor acción ellos mismos.

Los sistemas informáticos para facilitar la toma de decisiones tienen infinitas aplicaciones, en todo tipo de ámbitos. Uno de ellos son los vehículos. Desde hace unos años la aplicación de sistemas informáticos en vehículos ha tomado un nuevo impulso gracias a los vehículos autónomos y la miniaturización de la tecnología. La aplicación más común de los sistemas informáticos en vehículos es la monitorización, tanto del entorno como de los parámetros internos del mismo. Para esto, el vehículo utiliza una serie de sensores distribuidos a lo largo del mismo para recopilar el estado de cada rincón de este. Toda esta información llega a la máquina, donde un sistema, denominado sistema de fusión de datos multisensor recopila toda esta información y realiza la tarea que deba cumplir. Por ejemplo, para ser un vehículo autónomo, la tarea a realizar es con esta información, generar un mapa del entorno actual, y, junto a más información, como las leyes de tráfico o el destino deseado, decidir la acción más correcta. En otras palabras, se programa la máquina para realizar la tarea que hasta ahora hacía el conductor, observar el entorno, controlar el vehículo y junto a la normativa, llegar a un destino.

Así como la conducción autónoma, existen otras tareas donde utilizar esta información, como la optimización de los parámetros del vehículo para optimizar su funcionamiento o en el ámbito militar, la detección de amenazas cercanas u oportunidades de ataque.

Tampoco son iguales todos los tipos de vehículos. Por ejemplo, un vehículo marítimo, debe tener en cuenta, tanto con sensores que lo parametrizan como en sus ecuaciones internas, factores como el oleaje, el viento o la niebla, que en vehículos terrestres no tienen tanta relevancia. Lo mismo sucede con los vehículos aéreos. Estos cuentan con una normativa más reducida (al haber menos vehículos) pero incorporan una dimensión más, la presión por la altitud, la falta de visión por las nubes, etc.

Todos estos sensores proporcionan unos valores adecuados respecto a la medida original, pero en situaciones tan críticas como el transporte de personas o mercancías, es imprescindible garantizar la correcta medición del valor y su disponibilidad en todo momento. Para ello, puesto que todo sensor tiene un cierto margen de error, al estar expuestos a roturas, interferencias, ruido, fallos de comunicación, e incluso ser atacados directamente, los sistemas de fusión de datos multisensor tienen como principal objetivo solventar esta imprecisión mediante distintas vías, sobre todo en la detección de blancos cercanos.

Estos sistemas utilizan múltiples sensores redundantes y complementarios para obtener la mayor cantidad de información posible, de manera que el sistema de fusión de datos, con todas las medidas que apuntan a un mismo vehículo, sea capaz de realizar una fusión de todas ellas, para obtener una detección del mismo más precisa. Por ejemplo, en una situación aérea de

ámbito militar como la mostrada en la figura Fig.1.1, el caza puede contar por ejemplo con tres sensores: un radar tradicional, detección mediante visión por computador y sensores infrarrojos. En una situación como la mostrada, los tres sensores si no tienen ningún problema son capaces de detectar al avión que se ve a lo lejos, así como el propio piloto.

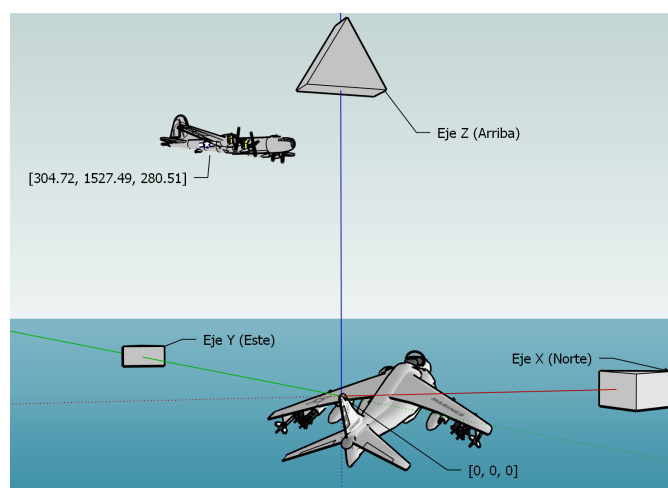


Fig.1.1: Escenario aéreo sin niebla

En cambio, si la situación empeora, como en la figura Fig.1.2, ni el piloto, ni la detección mediante visión por computador y sensores infrarrojos deberían detectar al blanco. Pero el radar sí funciona, por lo que al menos se cuenta con una entrada para el sistema de fusión. Sus algoritmos no podrán aumentar tanto la precisión de la posición del vehículo como si se tuvieran los tres sensores, pero igualmente mejorarán los valores teniendo en cuenta la trayectoria del mismo. Esto es debido a que el propio sistema de fusión es capaz de generar el modelo de la cinemática del vehículo, de manera que se tiene el conocimiento de la velocidad y el rumbo, siendo capaz de estimar donde estará en unos segundos en el futuro y utilizarlo para comparar la posición medida.

Además de esto, si el sistema de fusión tiene grandes capacidades, es capaz de predecir cuál es la situación global actual para decidir cuál es la mejor decisión a tomar, tanto de movimiento como de ataque. También puede llegar a ofrecer distintas alternativas teniendo en cuenta las posibles reacciones del otro barco, para obtener la decisión porcentualmente más adecuada en esta situación.

Como queda claro, un sistema de fusión de datos multisensor es un componente software de vital importancia para la seguridad e integridad de un vehículo. Por ello mismo es un software tan complejo, con varios subsistemas que trabajan al unísono para facilitar la toma de decisiones en situaciones tan críticas.

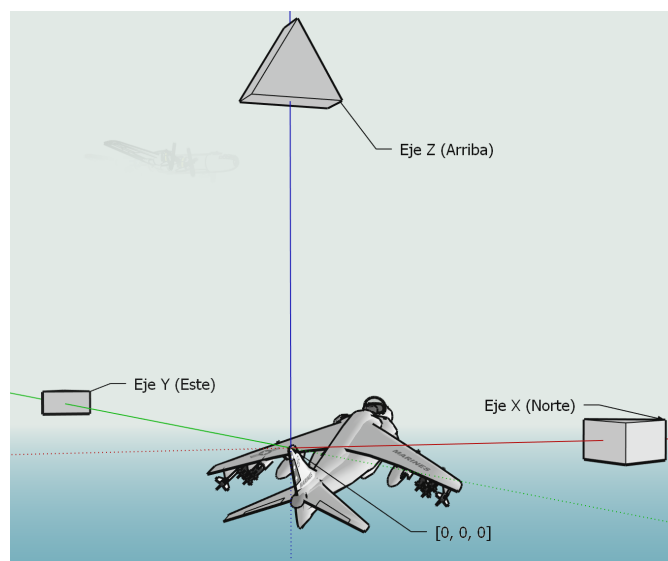


Fig.1.2: Escenario aéreo con niebla

Este área de investigación en auge sigue creciendo y cada vez realizar pruebas sobre él es más y más complejo. Esto es debido tanto a la complejidad del software y parámetros que optimizar, como a la dificultad de encontrar datos de sensores con los que realizar pruebas.

1.2 Objetivos

Dada esta necesidad de desarrollar sistemas de fusión multisensor más y más complejos, facilitar el proceso de prueba de estos sistemas parece una necesidad importante a corto plazo. Este proyecto se marca este objetivo, proporcionar a los usuarios una herramienta común sobre la que desarrollar sus pruebas de sistemas de fusión de datos multisensor. Para llegar a dar una solución, se han definido los siguientes objetivos:

- De manera teórica, es necesario comprender el funcionamiento teórico de los sistemas de fusión multisensor, especialmente los destinados a entornos 3D.
- Junto a este, se necesita estudiar los algoritmos que dan soporte a estos sistemas de fusión para poder comprender su funcionamiento, sus variables y ser capaz de implementarlo y analizarlo.
- A su vez, es importante conocer el funcionamiento detrás de cada tipo de sensor que sirve de entrada para los sistemas de fusión.
- Por otro lado, en el aspecto de la usabilidad de la herramienta a desarrollar, hay que analizar cuales son los problemas que tienen los usuarios

para así poder comprender sus necesidades y desarrollar un proyecto de ingeniería que los resuelva.

- Una vez conocidas las necesidades, se debe realizar este proceso de ingeniería, estableciendo una metodología de trabajo, desarrollar una planificación adecuada y un presupuesto válido.
- Este proyecto de ingeniería debe estar validado antes de comenzar con el desarrollo. Para ello se debe utilizar una técnica de validación para contrastar las necesidades de los usuarios y la utilidad de este proyecto.
- Una vez validado el prototipo, se procederá a desarrollar la herramienta. Para ello, se debe generar tanto la interfaz de usuario final, como la estructura de datos de cada aplicación, como la algoritmia que tienen por detrás.
- Todo este desarrollo debe ir acompañado de una explicación adecuada y solvente, tanto en esta memoria como en el código para ser reutilizable y ampliable en el futuro.

1.3 Estructura del documento

Una vez introducido el ámbito de la tesis, se procede a exponer las distintas secciones que tiene la misma. En el capítulo 2 se introducen de manera teórica todos los conceptos necesarios para la comprensión del resto del documento. Se continúa con la organización del proyecto, en el capítulo 3, introduciendo todos los aspectos para definir el trabajo como un proyecto de ingeniería. El siguiente capítulo nos lleva a la Especificación de requisitos 4, donde se analizan todas las necesidades a satisfacer con este proyecto, y se define la manera de acometer el proyecto con éxito. El capítulo 5 explica la herramienta software desarrollada, tanto desde el punto de vista de la usabilidad como de la algoritmia introducida en ella. Se continúa el capítulo con el destinado a probar el software desarrollado 6. En el, a partir de las necesidades marcadas se comprueba si se satisfacen las mismas con éxito.

Finalmente, en el capítulo 7, se concluye el proyecto, analizando de qué manera se puede prolongar y mejorar el trabajo aquí realizado.

Capítulo 2

Estado del arte

El estado del arte de este documento tiene la siguiente estructura. Se comienza exponiendo el término principal del trabajo en la sección 2.1, los sistemas de fusión multisensor, comprendiendo todos sus distintos componentes y posibles variantes. A continuación, en la sección 2.2, se introducen los principales sensores que sirven como fuente de entrada para un sistema de fusión multisensor, orientado a la vigilancia aérea. Por último, se analizan en la sección 2.3 distintas soluciones, tanto comerciales como de acceso libre, para facilitar el proceso de desarrollo y pruebas de un sistema de fusión.

2.1 Sistemas de fusión de datos multisensor

La fusión de datos es un término introducido por primera vez en 1985 por un grupo de expertos estadounidense, el Unión de directores de laboratorios (JDL). Este grupo lo definió como [5]:

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.

Esta definición era correcta en dicha época, donde la única aplicación de este término era en el ámbito militar. Actualmente, la tecnología se ha abaratado, permitiendo aplicar la tecnología en otros ámbitos. Por ello, Hall y Llinas [6], dos investigadores, emitieron esta definición más aceptada:

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.

Queda reflejado en ambas definiciones como el aspecto de combinación de la información de varias fuentes es el objetivo principal de un sistema de fusión, de manera que se obtenga una medida más precisa. Esta medida

después es procesada para varios fines distintos, siendo la inferencia, la capacidad de extraer información de dichas medidas, un aspecto común. En el ámbito de la vigilancia, esta inferencia se refiere a la capacidad de detectar, por ejemplo, si un vehículo está realizando un movimiento sospechoso, si está accediendo a una zona prohibida, superando un límite de velocidad, etc.

2.1.1 Clasificación de sistemas de fusión de datos

Los sistemas de fusión de datos son componentes software tan complejos, intersectando varias disciplinas informáticas, que cuentan con multitud de variantes. Por ello, es necesario clasificar cada uno en función de sus características.

2.1.1.1 Según los datos de entrada

Durrant-Whyte emitió la siguiente clasificación en 1988 [7], la cual discrimina al sistema de fusión en función de las características de los sensores utilizados:

- Complementarios: la información no coincide, sino que se aumenta entre sí el conocimiento sobre el escenario global. Por ejemplo: dos cámaras que apuntan en dos direcciones distintas.
- Cooperativas: la información recopilada es del mismo objetivo pero de distinta forma, haciendo que al final su fusión produzca información nueva. Por ejemplo: una cámara de vídeo, que fusiona el audio del micrófono y el vídeo.
- Redundantes: los sensores obtienen información repetida, de manera que aumenta la confianza en el resultado ante posible ruido. Por ejemplo: varias cámaras que apuntan al mismo objetivo.

2.1.1.2 Según la entrada-salida

En función del flujo de información generado por cada componente, Dasarathy [8] propone cinco categorías dentro de su modelo Datos Característica Decisión (DFD). La propagación de información en este modelo es bidireccional (ver figura Fig.2.1).

Los componentes propuestos son los siguientes:

- Entrada de datos - Salida de datos (DAI-DAO): se reciben los datos directamente del sensor y su salida son datos 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.

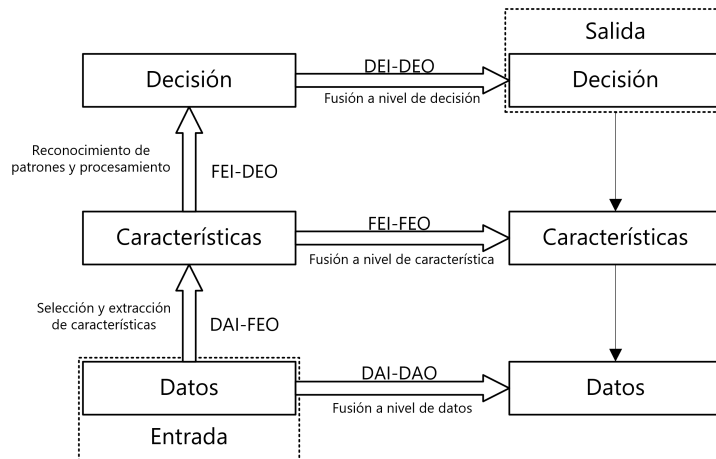


Fig.2.1: Modelo DFD [1]

- Entrada de datos - Salida de característica (DAI-FEO): a partir de datos en bruto, 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 Según la arquitectura

Los sistemas de fusión de datos también se pueden clasificar según el lugar donde se desarrolla el proceso de fusión. Se cuentan con cuatro tipos de arquitecturas distintas:

- **Arquitectura centralizada:** únicamente se procesa en el centro de fusión. Por ello, los sensores deben tener comunicación directa con este para transmitir la medición, y este almacenarla 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:** una red de nodos de procesamiento distribuidos son los encargados de realizar su propia fusión de datos. Estos nodos suelen ser distintos vehículos de una misma formación. Este nodo cuenta con sus propios sensores, y una vez fusionada la información en el nodo, es capaz de compartirla con el resto de nodos, realizando una segunda fusión. 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:** se realiza un método 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. [9], al proponer una arquitectura distribuida, muestra la clasificación de la figura Fig.2.2 según las necesidades:

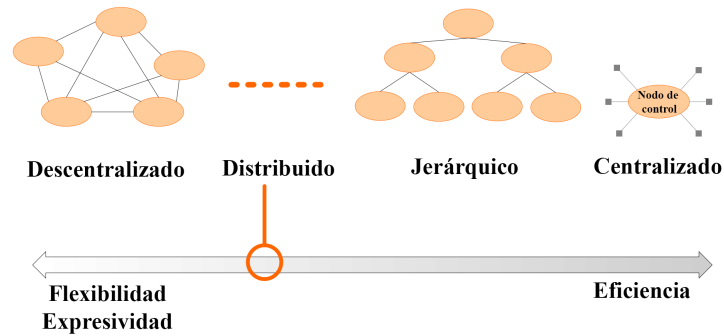


Fig.2.2: Clasificación arquitectura sistemas fusión

2.1.1.4 Según el modelo JDL

Por último, tenemos la clasificación más popular. La propuso el ya mencionado grupo estadounidense JDL, y fue ampliada posteriormente por una expansión del grupo, propuesta por Blasch y Plano [1]. El modelo cuenta con cinco (más un) niveles los procesos de los sistemas de fusión, en función de su complejidad. Los dos primeros se centran en el bajo nivel del procesamiento, mientras que los demás utilizan esos resultados para trabajar sobre una visión más general.

Las categorías a bajo nivel son las siguientes:

- Nivel 0 – preprocesado de fuentes de entrada: trabaja con las medidas de los sensores. Realiza la gestión de los datos en bruto, permitiendo su proceso al resto de niveles. EJ: colocar las mismas unidades a todas las entradas para ser comparables.
- 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 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.
- Nivel 5 - refinamiento humano: un sistema cuyo fin es proporcionar información a humanos debe permitir y facilitar la interacción entre el humano y el software. De esta manera, el humano puede modificar ciertos parámetros del resto del sistema, para ajustarlo a sus necesidades.

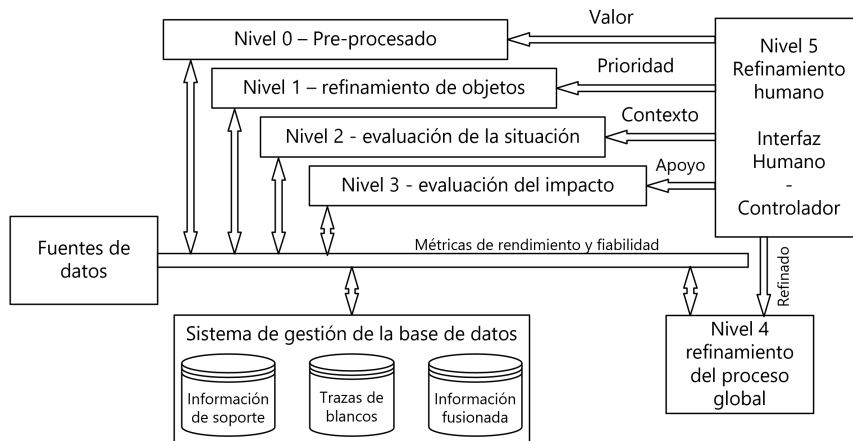


Fig.2.3: Modelo del proceso de fusión JDL-DFIG [1]

2.1.2 Principales componentes sistema de fusión

Como se observa en la clasificación realizada, los sistemas de fusión de datos multisensor cuentan con una gran cantidad de variantes y por ende de componentes que lo conforman. Por ello, es necesario entender y comprender las principales técnicas que se utilizan en estos sistemas, para poder implementarlas.

A modo de resumen, Hall y Llinas realizaron una tabla [6] en 1997 que clasifica las distintas técnicas según el nivel del modelo JDL. Ampliando esta con estudios de otros investigadores como Smith y Singh [10] o Castanedo [11], se muestra la tabla de técnicas de fusión de datos.

Nivel de proceso JDL	Tipo de técnica	Técnica	
Nivel 1 - refinamiento de objetos	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
Nivel 2 - evaluación de la situación	Agregación de objetos	Sistemas basados en el conocimiento • Sistemas expertos • Ontologías • Inferencia Dempster-Shafer	
	Interpretación de eventos/actividades		
	Interpretación del contexto		
Nivel 3 - evaluación del impacto	Estimación agregada	Redes de neuronas	
	Predicción del futuro		
	Evaluación múltiples perspectivas		
Nivel 4 - refinamiento del proceso de fusión	Evaluación del rendimiento	Técnicas de optimización multiobjetivo Técnicas de planificación	
	Control del proceso de fusión		
	Gestión de la misión		

Tab.2.1: Técnicas fusión de datos según nivel JDL

A continuación, se van a explicar las técnicas más importantes, haciendo hincapié en algunos algoritmos habitualmente implementados en sistemas de fusión.

2.1.2.1 Técnicas nivel 1

Este primer nivel es el más estudiado, al servir de base y entrada para el resto de componentes. Como se muestra en la tabla Tab.2.2, existen cuatro tipos de componentes principales en este nivel:

2.1.2.1.1 Alineación de datos Los contactos de cada sensor tiene características particulares. Unidades, precisión, frecuencia de detección, etc. Es necesario colocar todas las variables en el formato estándar definido en el sistema, de manera que sean comparables entre sí.

Este paso puede considerarse parte del nivel 0, dependiendo de si se almacenan los contactos originales o se transforman directamente a las unidades y coordenadas utilizadas en el sistema.

Cabe indicar que algunos sensores utilizan un software denominado tracker para identificar al vehículo detectado, puesto que algunos sensores, sobre todo los no cooperativos, un elemento denominado "tracker", no tienen un identificador de vehículo que relacione todos los contactos de un mismo vehículo.

2.1.2.1.2 Estimación de posición y atributos Las mediciones de cada sensor de un sistema de fusión de datos multisensor, orientado a la vigilancia, detectan objetos reales y devuelven su posición detectada. Estas posiciones detectadas están sujetas a ruidos, interferencias y otros posibles errores que pueden hacer que la medición registrada sea errónea. Este ruido variante a lo largo de una trayectoria introducido en el sistema hace que el resto de componentes actúen de manera errónea, y finalmente las decisiones tomadas no tengan los fundamentos adecuados.

Para ello, estos filtros de estimación tratan de extraer, sobre todos los contactos de cada sensor, la trayectoria más probable realizada por el vehículo, eliminando así parcialmente dicho ruido. Obteniendo una trayectoria más fiable que la original.

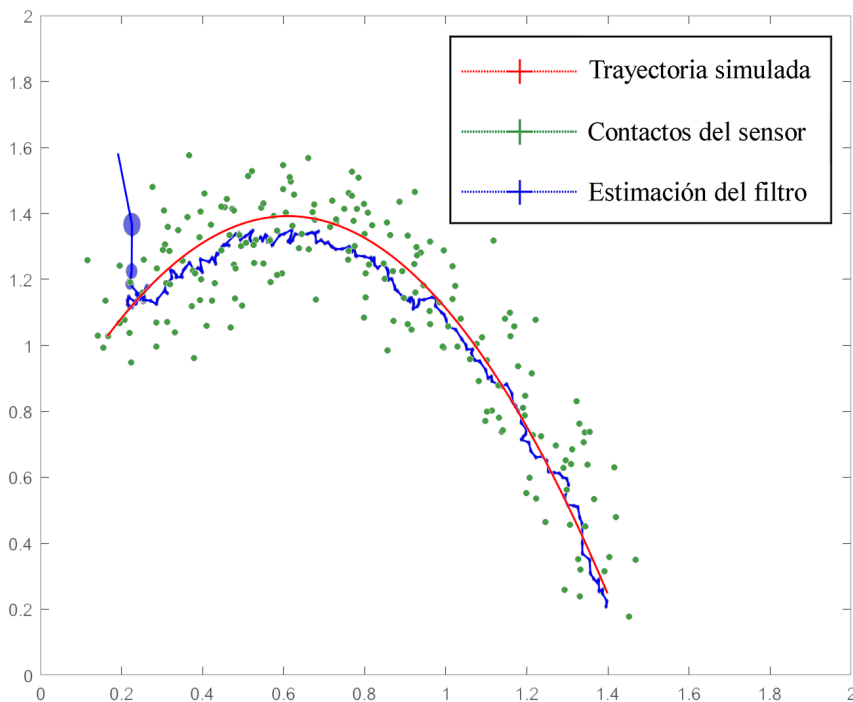


Fig.2.4: Ejemplo filtro de estimación

En la figura Fig.2.4 se muestra un ejemplo que explica la necesidad de utilizar un filtro de estimación. La posición XY real (rojo) realiza un giro constante y uniforme. En cambio, si se trazase la línea utilizando los con-

tactos del sensor, se contaría con una línea realizando un zig-zag continuo y muy abrupto, lo cual haría que la velocidad y aceleración fuesen extremadamente variables, haciéndolas inútiles para cualquier decisión. Tras aplicar un filtro de estimación se obtiene la línea azul, la cual modela de mejor manera la trayectoria real, eliminando en gran manera el ruido introducido por el sensor. Este es una entrada mucho más óptima para un sistema de fusión.

Los filtros de estimación se aplican sobre modelos matemáticos que representan los componentes cinemáticos, es decir, de movimiento, del vehículo monitorizado. Este se modela como un vector que recoge las variables cinemáticas utilizadas en el filtro. Las más utilizadas son tanto la posición, en sus distintas coordenadas (X, Y, Z) , como la velocidad e incluso la aceleración. También se suele utilizar la orientación del vehículo.

Existen múltiples técnicas para completar este componente. En la figura Fig.2.5 se muestra una clasificación de las técnicas de estimación desde sus raíces más genéricas, clasificadas según sus características.

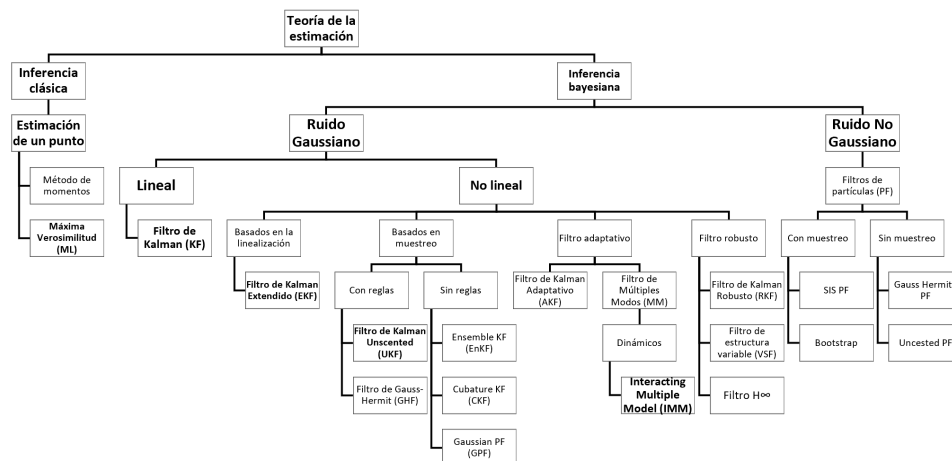


Fig.2.5: Clasificación filtros de estimación

Las utilizadas en este ámbito son técnicas bayesianas, al modelarse el ruido según la teoría de Bayes. Dentro de estos, se utilizan los filtros no lineales y filtros de partículas, puesto que la variación de sus componentes no se modela mediante una función lineal (ver por ejemplo la figura Fig.2.4).

A continuación se exponen, de manera superficial, las técnicas mostradas en **negrita** en la figura Fig.2.5.

Filtro de Kalman El Filtro de Kalman (KF) sienta las bases de los filtros de estimación actuales. Este, desarrollado en 1960 [12], fue primordial para hitos como el alunizaje del Apollo 11 [13]. Su funcionamiento erradica en un bucle compuesto por dos fases, ver imagen Fig.2.6. A partir de todos los contactos de un mismo sensor, se genera en cada instante un contacto

corregido, generando la trayectoria estimada.

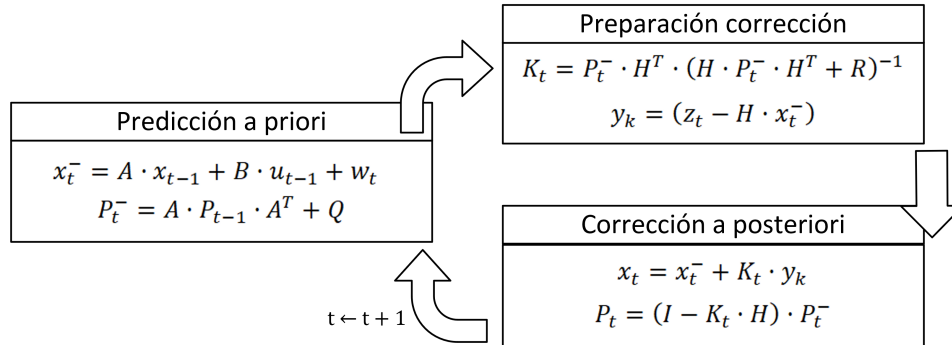


Fig.2.6: Funcionamiento Filtro de Kalman (KF)

El primer paso es la fase de predicción. A partir del modelo matemático (la posición, dirección y velocidad del vehículo), estimar cómo se encontrará el modelo dentro de X tiempo. Esta predicción se genera sin utilizar ningún contacto, únicamente en el modelo que ha aprendido hasta ahora. En la primera iteración, el modelo es muy ruidoso, pero a medida que avanza, va mejorando su comportamiento.

Por otro lado, la corrección utiliza la observación, el contacto de los sensores, para matizar el modelo predicho en la fase anterior, y de esta manera obtener una corrección que utilizar en el siguiente instante.

Este filtro sienta las bases de casi todos los filtros posteriores, desarrollados debido a que este funciona (de manera óptima) únicamente en variables lineales. Al estimar las componentes cinemáticas, que no siguen una función lineal, no es el filtro adecuado para este problema.

Filtro de Kalman Extendido El Filtro de Kalman Extendido (EKF) es la primera abstracción del KF para variables no lineales. Es uno de los filtros más conocidos, al estar muy estudiado en la literatura de la vigilancia. Su funcionamiento utiliza las ecuaciones del KF, adaptando la entrada no lineal mediante derivadas parciales (conocidos como matrices jacobianas) para linealizar la entrada.

Este filtro tiene un funcionamiento adecuado en sistemas con una no linealidad reducida. A medida que aumenta la no linealidad, es recomendable utilizar otros filtros más complejos. Además, tiene un alto coste computacional, al ejecutar las derivadas parciales.

En la figura Fig.2.7 se ilustra el funcionamiento de este filtro.

Filtro de Kalman Unscented El Filtro de Kalman "Unscented" (UKF) es uno de los filtros adaptados para una mayor no linealidad. Su funcionamiento utiliza una serie de valores concretos, denominados "sigma points",

para estimar la no linealidad del problema, y así calcular de mejor manera la predicción. En la figura Fig.2.7 se ilustra el funcionamiento de este filtro.

Filtro de Kalman Ensemble Este filtro Filtro de Kalman Ensemble (EnKF) surge también de la necesidad de mejorar el rendimiento del EKF. Al igual que el UKF, utiliza una serie de puntos para estimar la no linealidad, pero a diferencia de este, que utiliza unos puntos concretos, el EnKF escoge una serie de puntos aleatorios, mediante el método de Monte-Carlo.

Interacting Multiple Model Este filtro Filtro de Múltiples modos interoperantes (IMM) cuenta con un enfoque distinto. En vez de utilizar un filtro para realizar la estimación con unas características concretas, se encarga de gestionar la estimación de varios filtros en paralelo, de manera que cada filtro abarque un tipo de movimiento del vehículo. De esta manera, se genera un filtro que se puede adaptar a todo tipo de condiciones y movimientos.

Filtro de partículas Por último, se cuenta con los Filtro de partículas (PF). Estos filtros siguen también el enfoque de utilizar una gran cantidad de puntos, escogidos de manera aleatoria mediante el método de Monte-Carlo. Estos puntos, denominados partículas, en función de cuán bien modelen la no linealidad, se les otorga un peso distinto, el cual favorece utilizarlos en el futuro y tenerlos en cuenta para generar la estimación.

Conclusión Como se puede observar, se cuenta con una gran cantidad de filtros de estimación, cada uno con unas características diferenciadas del resto, lo que hace que su resultado varíe en unas u otras condiciones. Por ello, la mejor manera de obtener el mejor filtro en cada escenario es probar y comparar el funcionamiento de cada uno.

2.1.2.1.3 Asociación de datos / objetos Otro componente principal es el encargado de realizar la asociación de los datos de cada fuente en un mismo ente común. Ya se ha comentado que en sistemas orientados a vigilancia, el objetivo es aumentar la precisión del movimiento que está realizando el vehículo. Esto se resumen en el modelo matemático explicado anteriormente.

Este componente se encarga de evaluar cada contacto de los distintos sensores y asignarlo a un vehículo global. Los sensores tienen su propio identificador de vehículo (ya sea propio o generado por un tracker), pero estos identificadores no son comparables entre sensores. Por ello, es necesario incorporar un componente capaz de relacionar los componentes cinemáticos del mismo para asociarlos a un vehículo, dentro del escenario global.

Por ejemplo, en la figura Fig.2.8 se muestra un caso complejo de asociación. En la gráfica b se observan todos los contactos, con su posición

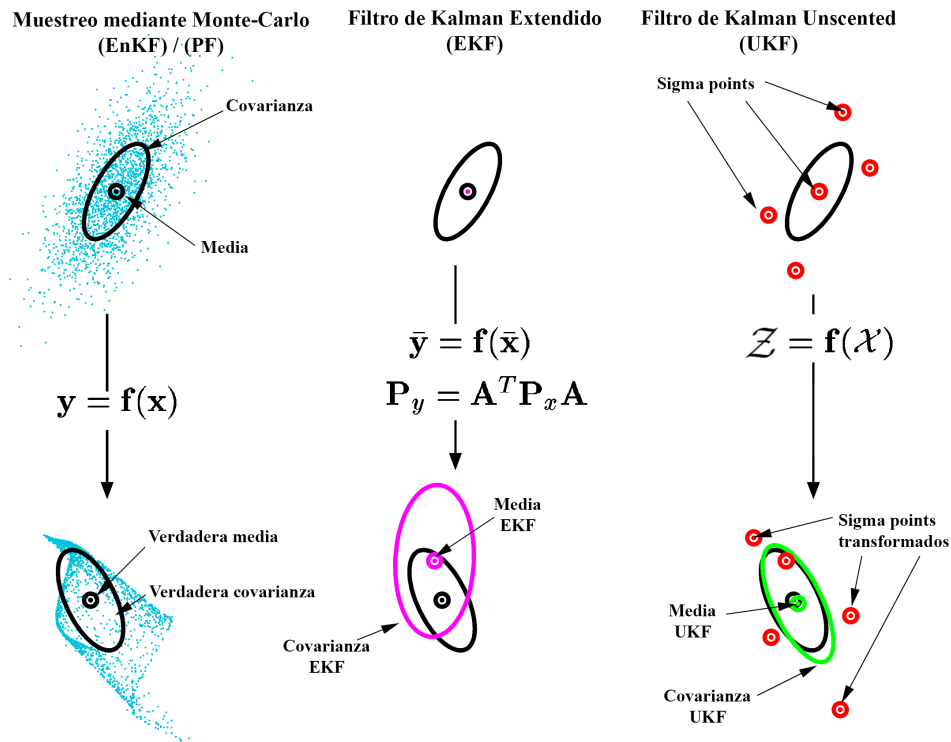


Fig.2.7: Comparativa filtros de estimación

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.

Para este paso es habitual utilizar la distancia Mahalanobis [14]. 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

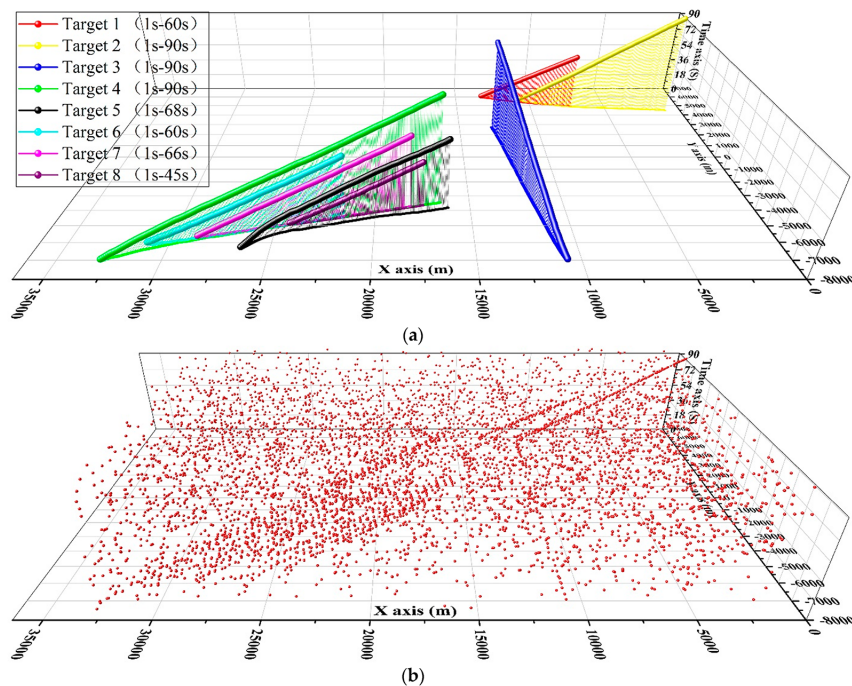


Fig.2.8: Ejemplo contactos reales sin identificador (b) filtrados y asociados en 8 trazas (a) [2]

más simples que se puede desarrollar para asociar las trazas, destacando el algoritmo k-medias [15].

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 [16] (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.

Poco tiempo después, los mismos investigadores adaptaron este algoritmo para funcionar para múltiples trazas locales a la vez [17], 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 Raid [18] 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 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 [19] y después Munkres en 1957 [20]).

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, el nivel 1 del modelo JDL tiene un componente destinado a extraer información que identifique el objeto que se está vigilando. Esta información, denominada información del contexto del escenario, se utiliza en posteriores niveles para tomar decisiones y ajustar parámetros internos del sistema.

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, Árbol de decisión (DT), etc. Por ejemplo, en la literatura, en [21], se utilizan imágenes como entrada de una red neuronal para clasificarla y reconocer el tipo de barco. En [22] a su vez 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 escenario, a lo largo de la literatura se han hecho algunas propuestas. A pesar de ello, este ámbito de la literatura no está tan explorado 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 [23], 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 [24] se implementa una ontología sobre un sistema de fusión real basado en cámaras.

Otras, como redes bayesianas [25] 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 [26] 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 [27] 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 [28] se optimiza la precisión de los resultados de fusión utilizando un sistema planificado respecto a un sistema estacionario.

2.1.2.4 Métricas de rendimiento

Como se puede observar, todas estas técnicas son muy abstractas y su funcionamiento no es trivial. Por ello, para conocer si su resultado es aceptable bajo una serie de condiciones, se utilizan una serie de métricas. Estas métricas habitualmente se utilizan para el nivel 4.

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, como por ejemplo en una simulación), 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. $\tilde{z}_t = z_t - Hx_t^-$

2.1.2.4.1 Precisión del estado Para medir la precisión de la estimación proporcionada (tanto en los estimadores de estado, como posteriormente al generar la traza asociada), 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.

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.

2.1.2.4.2 Consistencia Por otro lado, 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.

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 utilizando las mediciones de los sensores, al utilizar únicamente la innovación.

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.

2.1.2.4.3 Tiempo de ejecución Por último, para comprobar si la carga computacional del componente es adecuada, se mide el tiempo de cómputo del mismo para poder comprobar problemas en una instancia concreta o compararlo con otros algoritmos.

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

2.2 Sensores en el ámbito aéreo

Los sistemas de fusión de datos destinados a la vigilancia aérea, como su propio nombre indica, trabajan con los datos que representan los distintos vehículos, recopilados a través de distintos sensores que los detectan y registran distintas mediciones con una cierta frecuencia. Como ya se ha comentado, es importante que los sensores implementados sean complementarios entre sí para funcionar en todo tipo de condiciones. Por ello, existen distintas tecnologías utilizadas para recopilar esta información.

Antes de exponerlas, es necesario comprender dos conceptos clave que discriminan a cada tipo de sensor.

- Cooperativos / no cooperativos: un sensor es cooperativo si el vehículo detectado debe realizar algún procesamiento o tarea para generar la información.
- Dependiente / independiente: el sensor es dependiente de los sistemas del vehículo detectado, si la información registrada es la generada por el mismo vehículo mediante sus sistemas de navegación. No es dependiente, si el propio sensor tiene una tecnología externa capaz de pro-

porcionar una posición completamente independiente de sus sistemas de navegación.

Los sensores dependientes utilizan los sistemas de navegación de los vehículos aéreos. Para comprender qué información y cómo la recogen, a continuación se explican de manera breve los sistemas de navegación de vehículos aéreos.

2.2.1 Sistemas de navegación

2.2.1.1 GNSS

Los Sistema global de navegación por satélite (GNSS) utilizan un conjunto de satélites (constelación de satélites) que orbitan alrededor de la Tierra para localizar en qué punto se encuentra cualquier receptor sobre la superficie terrestre. Existen varios sistemas GNSS distintos, propuestos por distintas regiones de la Tierra. Todos ellos cuentan con características y precisiones similares, pero les interesa crear el suyo propio para no depender de la tecnología de otra región. A pesar ello, el concepto de manera habitual es conocido como GPS, al ser el primero en funcionar, en 1978. Actualmente funcionando a nivel global existen los siguientes sistemas:

- GPS, de Estados Unidos.
- GALILEO, de la Unión Europea.
- GLONASS, de la Federación Rusa.
- BeiDou, de China.

2.2.1.1.1 Funcionamiento Este sistema de geolocalización permite a cualquier receptor de estas señales inferir su posición sobre la Tierra. Para ello, a partir varias señales emitidas por una serie de satélites que orbitan alrededor de la Tierra (requiere mínimo 4 de ellas), las cuales proporcionan la posición del satélite y Tiempo de transmisión (TOT), el receptor mediante ecuaciones es capaz de aproximar su posición en latitud, longitud y altura. Estas ecuaciones utilizan tanto el TOT como el Tiempo de llegada (TOA) del receptor.

El problema del GNSS es su precisión, al depender de la calidad del receptor para detectar las transmisiones de más satélites, y la altura calculada es calculada a partir de la elipsoide WGS84, la cual difiere de la altura real habitual, medida sobre el nivel del mar. Para mejorar esta superficie y hacerla más real, se calcula la altura sobre el geoide, una aproximación irregular de la forma del mar de la tierra, calculado mediante el potencial gravitatoria de la tierra. Sus irregularidades aún así son de como mucho 100 metros, lo cual dista de los 10 kilómetros de la tierra real (Everest y la Fosa de las Marianas).

2.2.1.1.2 Sistemas de aumentación Todos estos estándares funcionan de manera adecuada, pero con el objetivo de garantizar un mejor funcionamiento, tanto en precisión como en cobertura y fiabilidad, se han desarrollado sistemas de aumentación Sistema de aumentación basado en satélites (SBAS), los cuales apoyan y corrigen las desviaciones que producen los propios satélites. Estos sistemas de aumentación (existen varios, desarrollados o por países o regiones) utilizan estaciones en la Tierra, cuya posición es conocida con exactitud, para corregir la desviación proporcionada por las señales de los satélites, y así notificarlo a los receptores, de manera que puedan corregir dicha desviación y mejorar su localización.

Otra técnica de aumentación es la conocida como Sistema de Posicionamiento Global Diferencial (DGPS), cuya única diferencia es que la información de la corrección no se envía desde satélites, sino que se hace desde la propia estación terrestre.

2.2.1.2 INS

Otro de los sistemas más importantes en un vehículo es su Sistema de navegación inercial (INS). Este cuenta con un componente denominado Unidad de medición Inercial (IMU), el cual integra varios sensores de movimiento y rotación para estimar tanto la posición, como el rumbo y la velocidad del propio vehículo en cada instante. De esta manera, el vehículo, ya sea sobre el plano o un vehículo volador, puede estimar con precisión sus componentes kinemáticas sin depender de información externa. Es importante puesto que es el método más sencillo para estimar los ángulos de rotación en la dinámica del avión. Es decir, su alabeo (roll), cabeceo (pitch) y guiñada (yaw). En la figura Fig.2.9 se muestra el eje de rotación de cada uno de estos términos.

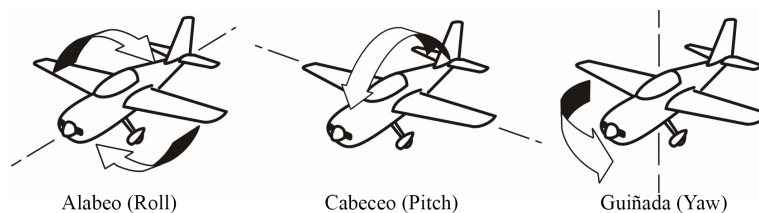


Fig.2.9: Rotación avión

Esta información es menos precisa que cualquier sistema de vigilancia externo, pero es útil cuando no se puede disponer de alguno de ellos, un hecho común al desplazarse por ejemplo en los océanos.

2.2.2 Radar primario

El Radar primario (PSR) es el sistema de vigilancia que utiliza la definición original de esta tecnología. RADAR es un acrónimo que significa por sus

siglas en inglés: detección y medición de distancias por radio (RAdio Detec-tion And Ranging). Como se indica, se utilizan impulsos electromagnéticos para detectar objetos y medir la distancia a los mismos. Los objetos pueden ser tanto estáticos como móviles. Si el impulso lanzado en una dirección im-pacta contra un objeto, esta onda será reflejada en todas direcciones (efecto conocido como eco). Este eco termina volviendo a la antena de emisión, conociendo así la dirección donde está el vehículo detectado (azimut), y la diferencia de tiempo entre la emisión y recepción del reflejo, su distancia aproximada. Además, gracias al efecto Doppler, se puede obtener la veloci-dad angular del objeto a partir del eco recibido y su diferencia respecto a la onda emitida.

Este tipo de ondas se emiten desde antenas de grandes dimensiones para contar con una mayor cobertura. Existen varios tipos de antenas en función de la forma de la onda a enviar, pero lo más habitual es colocar la antena sobre una superficie rotatoria. Gracias a esta superficie, el radar es capaz de realizar barridos sobre los 360 grados, detectando todo objetivo cercano a él.

Las antenas se pueden colocar también en vehículos, de manera que sean portables, o existen combinaciones de antenas (array antena) para aumentar las capacidades a recibir.

2.2.2.1 Versiones 3D

Una de las variantes que se están implementando a día de hoy son los radares en tres direcciones. Esto quiere decir que, aparte del azimut (direc-ción) y distancia, se proporciona la elevación del vehículo detectado. Esto se obtiene implementando una antena que emita haces filiformes (pencil beams) sobre una superficie que además de rotar sobre su propio eje vertical, rota su eje X, variando el ángulo de elevación de las ondas y gracias a dicho ángulo y la distancia, se permite conocer la altura de los vehículos.

2.2.2.2 Radar de superficie

El Radar de superficie (SMR) es una variación del radar primario es-pecializado en vigilancia aérea, por lo que se incorpora habitualmente en aeropuertos. La ventaja de este radar es que está pensado para distancias más cortas que un PSR habitual, al dedicarse exclusivamente a monitorizar vehículos en fase de despegue / aterrizaje. Gracias a esta distancia más corta puede disminuir la longitud de onda, aumentar la velocidad de giro de la base y por ende obtener una tasa de refresco más elevada.

2.2.2.3 Ventajas y desventajas

La principal ventaja del radar primario es ser un sistema de vigilancia que no necesita de la cooperación del vehículo, de manera que si hay un vehículo cercano, este lo detectará, coopere o no. La detección no está garantizada



Fig.2.10: Radar de superficie

siempre, sino que depende de de varios factores, como la superficie del vehículo para que se refleje la emisión y las características del propio radar (potencia, superficie de antena, longitud de onda emitida, etc.). El concepto Radar Cross-Section mide la 'detectabilidad' de un objetivo por un radar. Aun así, el PSR tiene desventajas. Es capaz de detectar el vehículo, pero no de identificarlo. Tampoco es capaz de determinar la altura a la que se encuentra respecto al radar, factor muy importante al hablar de sistemas para vigilancia aérea.

2.2.3 Radar secundario

A la par que se desarrolló el PSR, se creó el IFF. Esta tecnología es la base militar de lo que hoy se conoce como Radar secundario (SSR). Con el objetivo de diferenciar a los vehículos amigos de los enemigos, ya que el PSR no los identifica, se incorpora junto al radar primario otra antena capaz de enviar mensajes codificados al transpondedor instalado en el vehículo amigo, con forma de un tren de impulsos modulados. En la figura Fig.2.11 se muestra un radar primario y un radar secundario implementados en la misma torre. Tras realizar la interrogación, el transpondedor receptor del vehículo procesa el tren de pulsos, comprende el mensaje y manda una respuesta a la estación emisora. De esta manera, los vehículos se identifican utilizando IFF.

Con esta base, se desarrolla un sistema SSR, el cual utiliza el tren de pulsos para aparte de identificarse, permite enviar información de los componentes kinemáticos internos del avión y que la estación los conozca. La información enviada depende del modo de interrogación en el que se coloque el transpondedor. Los modos se diferencian por la separación entre cada pulso. Los modos utilizados a día de hoy son:



Fig.2.11: Radar primario (abajo) y secundario (encima)

- Modo 1: Modo militar para notificar la misión o rol actual.
- Modo 2: Modo militar para identificar al vehículo.
- Modo 3/A: El modo más habitual. Identificador colocado por el piloto para el control del tráfico aéreo. También se utiliza en el modo S.
- Modo C: Se transmite la altitud barométrica del vehículo para el control del tráfico aéreo. También se utiliza en el modo S.
- Modo S: realiza una interrogación selectiva para transmitir distinta información. Es uno de los modos más habituales a día de hoy, al permitir comunicarse con un vehículo en concreto. Para ello cada vehículo cuenta con un identificador asignado, el cual puede variar en función del país, para responder únicamente a sus llamadas. Se cuenta con tres tipos de comunicaciones principales.
 - La primera recopila el identificador de cada transpondedor del vehículo.
 - La segunda envía información desde la estación a todos los receptores, sin esperar respuesta.
 - La tercera es la ya explicada comunicación selectiva para un único receptor.

Finalmente, para el ámbito civil, se cuenta con un intermodo, el cual envía a los transpondedores peticiones para varios modos a la vez. En concreto se tienen dos.

- La llamada de A/C/S: se hace la comunicación habitual de A y C, y se pide el identificador del modo S.
- Lo mismo pero solo con A y C.
- Modo 5: Modo militar equivalente al modo S. Envía la información codificada de manera especial.

2.2.3.1 Radar secundario monopulso

El Radar secundario monopulso (MSSR) surgió posteriormente para solventar los problemas del SSR. Los radares SSR han mejorado sus prestaciones con el paso de los años, y ya no es necesario realizar un amplio barrido cónico continuo para obtener el azimut de un blanco. Actualmente, los radares utilizan el concepto de monopulso. Este se basa en realizar una comparación entre la emisión y el eco recibido de un único haz, pero dividido en dos señales. Estas señales cuentan con una dirección diferente que al rebotar y volver a la antena, se comparan y se determina de mejor manera en dónde se encuentra el blanco. El error en azimut se reduce aproximadamente de 0.1 grados a 0.006 grados.

2.2.3.2 Ventajas y desventajas

Los radar SSR son sistemas bastante más complejos que los PSR, pero son muy complementarios entre sí. Uno no proporciona altitud ni identificación, pero encuentra todo blanco cercano. El otro conociendo que hay un blanco, le realiza una interrogación para obtener esta información. Por ello, es habitual ver el SSR implementado en la misma superficie rotatoria, encima del PSR.

2.2.4 ADS

Los sistemas más nuevos son los Sistema automático dependiente de vigilancia (ADS). Estos, como su nombre indica, buscan automatizar todo el proceso de vigilancia, mediante un sistema cooperativo y dependiente entre avión y receptor, el cual puede ser la estación base, u otro avión. Este concepto hace que el propio vehículo proporcione de manera automática su información cinemática captada por sus sistemas de navegación y GNSS, así como información extra que le identifique. La comunicación se realiza por los transpondedores SSR que hemos observado anteriormente, utilizando el modo S. ADS actualmente se implementa de dos manera distintas.

2.2.4.1 ADS-C

La primera es Sistema automático dependiente de vigilancia - contrato (ADS-C), donde la información se envía de manera periódica a una estación

terrestre, previa solicitud de la estación terrestre de la frecuencia e información deseada, el "contrato" con el vehículo. De esta manera y de manera automática, el vehículo envía la información a la estación terrestre, generando un canal de información punto a punto.

2.2.4.1.1 Ventajas y desventajas Ventajas:

- Vigilancia en áreas más complejas y amplias, donde otros sensores no funcionan adecuadamente.
- Permite comunicación de datos tierra-aire.

Desventajas:

- Requiere un gran desembolso en comparación con otras soluciones.
- Se basa en los sistemas internos de la aeronave.
- No sirve para vigilancia, al no comunicarse con el resto de aeronaves.
- Tiene una frecuencia de refresco peor que el resto de soluciones.

2.2.4.2 ADS-B

Por otro lado tenemos el Sistema automático dependiente de vigilancia - difusión (ADS-B). Este modo, en vez de enviar la información solicitada por una estación terrestre, se difunde de manera automática a todo receptor con un transpondedor en modo "ADS-B in". De manera periódica envía su identificador y componentes cinemáticos, de manera que el resto de vehículos conozcan su posición. Esta tecnología es relativamente nueva y va a ser obligatoria en todo avión civil a partir de 2020, tanto en USA como en Europa.

2.2.4.2.1 Ventajas y desventajas Las principales ventajas de ADS-B son:

- Rango de cobertura elevada, al igual que ADS-C.
- Funciona adecuadamente en entornos de alta densidad de vehículos, como por ejemplo un aeropuerto, al permitir procesar más de 500 vehículos por segundo.
- Alta tasa de refresco, variable según necesidad.
- Barato de implementar y mantener.
- No requiere comunicación entre emisor receptor, únicamente recibir la señal.

Desventajas:

- Su principal desventaja es que no se puede conocer quién ha escuchado dicha información, al no haber ninguna verificación y enviarse por difusión.
- Están perfeccionándose a día de hoy sus estándares, por lo que no todos los aviones lo tienen incorporado.
- Depende de la información recibida por el GPS, que puede tener ruido.

2.2.5 Multilaterización

La multilaterización es un sistema de vigilancia y posicionamiento cooperativo independiente, el cual utiliza tres o más estaciones base para posicionar al vehículo. Su funcionamiento es sencillo. Una vez el vehículo está dentro de rango, los sistemas detectan su señal SSR y le realizan un interrogatorio para identificarlo, en el intermodo ACS. Es con la respuesta emitida por el vehículo con lo que se posiciona al mismo. Las estaciones base están sincronizadas con el mismo tiempo y ellas miden el Tiempo de llegada (TOA) de la respuesta. A partir de la Diferencia de tiempo de llegada (TDOA) de cada par de estaciones, se genera una hiperboloide que define la zona donde estaría el vehículo. Una vez se añaden tres de ellas, se cuenta con un posicionamiento en tres dimensiones. Su error de posición varía en función de la distancia del blanco, siendo entre 3 y 6 metros en los alrededores de un aeropuerto, y entre 10 y 13 a unos 16 kilómetros de distancia.

En la figura Fig.2.12 se expone un ejemplo para comprender el funcionamiento de esta tecnología.

2.2.5.1 WAM

Multilateración en áreas amplias (WAM) utiliza el mismo concepto del MLAT, pero aplicado a superficies mucho más amplias, de hasta 100 kilómetros de distancia entre antenas. Aparte de mejorar la precisión de un SSR por separado, su coste es bastante inferior a una alternativa que utilice radares secundarios.

2.2.5.2 Ventajas y desventajas

Ventajas:

- Funciona en lugares donde los radares generan una dead zone, o su implementación es muy compleja.
- Permite la detección a alturas más bajas que un SSR habitual.

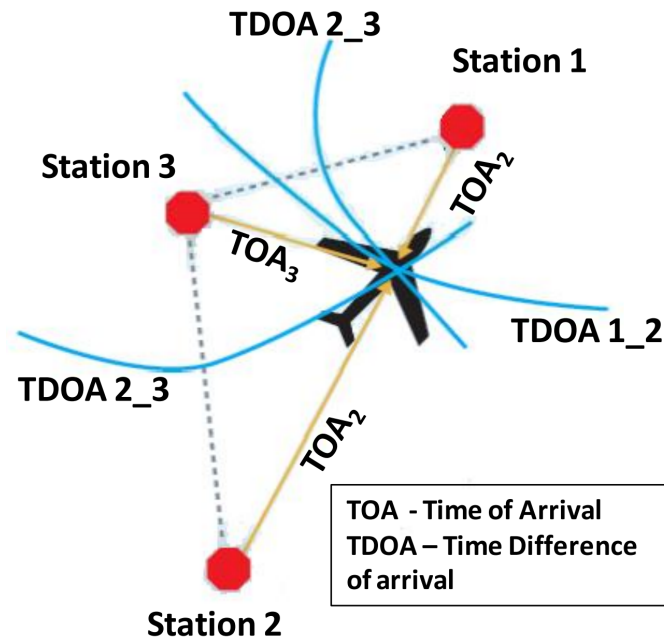


Fig.2.12: Ejemplo MLAT [3]

- Se puede utilizar únicamente tres antenas si se utiliza también un SSR para obtener la altitud del blanco.

Desventajas:

- Este sistema requiere sí o sí de varias antenas, mientras que un radar funciona con una única antena.
- Su principal desventaja es que las antenas deben ser capaces de comunicarse entre sí, por lo que en caso de contar con montañas u otros elementos que limiten la visión, es necesario incluir más antenas auxiliares.

2.2.6 IRST

Esta tecnología, implementada por primera vez en un vehículo en 1950, detectan las emisiones infrarrojas que dejan los objetos, como aviones y helicópteros. Esta tecnología es útil, puesto que permite en el entorno militar a un piloto detectar cualquier estela de un enemigo de manera pasiva, puesto que a diferencia del radar, este sistema no emite ninguna onda que le exponga.

En la figura Fig.2.13 se muestra una implementación en un caza de un sistema IRST. Cabe destacar que estos sistemas también se utilizan en estaciones base.



Fig.2.13: Implementación de unIRST

2.2.6.1 Ventajas y desventajas

Sus ventajas son:

- Es independiente, por lo que detecta todo tipo de vehículos, cooperen o no.
- Difícil detección de uso de las emisiones infrarrojas, a diferencia de las ondas Radar.
- Proporciona una alta precisión en azimut.

Sus desventajas son:

- Alcance reducido en comparación con otras soluciones.
- No funciona en malas condiciones atmosféricas.
- No es capaz de medir la distancia con precisión.

2.2.7 Comparativa

Como se ha expuesto, se cuentan con una gran cantidad de posibles entradas para un sistema de fusión de datos de vigilancia ideal. Existen muchos de ellos muy complementarios entre sí, pero a pesar de ello, a más fuentes utilizadas, mayor fiabilidad general tendrá una traza fusionada.

Para concluir, se expone una tabla que resume las capacidades de cada tipo de fuente de información y facilita su comprensión. En la tabla se utiliza una escala, de 1 a 10, para valorar las características, siendo 1 el mejor valor y 10 el peor.

	PSR	SSR	ADS-B	ADS-C	MLAT	IRST
Dependiente	No	No	Sí	Sí	No	No

	PSR	SSR	ADS-B	ADS-C	MLAT	IRST
Cooperativo	No	Sí	Sí	Sí	Sí	No
Identifica blancos	No	Sí	Sí	Sí	No	No
En malas condiciones atmosféricas	Sí	No	No	No	No	No
En malas condiciones de terreno	No	No	No	Sí	Sí	No
Tasa de refresco	4	6	8	6	6	9
Precio	2	5	8	3	6	7
Alcance	5	8	8	9	7	6

Tab.2.2: Técnicas fusión de datos según nivel JDL

2.3 Soluciones alternativas existentes

En lo referente a las soluciones existentes que cumplan los objetivos marcados en este trabajo, no existe ninguna aplicación ni software accesible con todas estas características concentradas. Existen, eso sí, software para generar simulaciones de sensores y por ende, generar contactos de vehículos, y por otro lado, software para desarrollar sistemas de fusión.

2.3.1 Software para generar simulaciones

Tras buscar tanto en publicaciones científicas como a lo largo de Internet, se han encontrado las siguientes herramientas que permiten realizar una simulación de un escenario de vehículos, y obtener su posición mediante varios sensores.

2.3.1.1 SLAMEM

La compañía de investigación Toyon, destinada a la investigación de sistemas de defensa, se encuentra desarrollando su sistema de simulación Simulador de posición y ataque de misiles enemigos (SLAMEM) [29]. Este tiene como principal objetivo validar sistemas de fusión de datos y algoritmos concretos sin tener que ponerlos en un sistema y escenario real, minimizando así costes. En palabras textuales de la empresa:

SLAMEM ofrece un puente entre la fusión de datos más académica y su implementación en plataformas finales con testeo en entornos controlados.

2.3.1.2 TRACKER

Este software, desarrollado por Okello en 1999 [30] permite simular los contactos de los vehículos y aplicarle filtros de estimación. Todo ello utilizando MATLAB, y gestionado mediante una interfaz gráfica desarrollada en JAVA.

2.3.1.3 Flames

El software de Ternion es una herramienta con interfaz gráfica que permite el análisis de escenarios [31]. Está enfocado a todo el proceso de generación de escenarios y análisis sobre el mismo. La herramienta ofrece una gran cantidad de customización. No está orientada directamente a sistemas de fusión de datos, pero es posible que se pueda incorporar este tipo de sistemas en la herramienta.

2.3.1.4 Multi-source Report-level Simulator

Por último se cuenta con el software Multi-source Report-level Simulator (MRS) [32]. Esta herramienta creada en el ámbito académico tiene como objetivo generar contactos de una serie de sensores incorporados y customizables sobre superficies con distintas alturas generadas aleatoriamente. Se permite escoger los sensores y los vehículos (terrestres) a utilizar.

2.3.1.5 Comparativa

De todas las herramientas, la más cercana a la idea original del trabajo, es la herramienta TRACKER, seguida por MRS y finalmente, a priori, (no está terminada) por SLAMEM. Aún así, TRACKER tiene más de 20 años y no se ha encontrado un enlace de descarga, MRS tampoco, al tener 15 años y SLAMEM parece ser un proyecto que abarca mucho más de lo necesario a priori para los sistemas de fusión de datos.

2.3.2 Software para probar sistemas de fusión

Por otro lado, sistemas de fusión de acceso sencillo existen pocos. A continuación se muestran los encontrados.

2.3.2.1 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.

2.3.2.2 Tracker Component Library

Por otro lado, se ha encontrado otra librería implementada en el lenguaje de programación MATLAB. Esta, desarrollada por Crouse [33] 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.

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

2.3.2.3 Stone Soup

Por último, se expone el proyecto más similar y ambicioso de todos. Propuesto por la Sociedad internacional de fusión de información (ISIF), este proyecto [34] aún en fase de desarrollo pretende desarrollar un framework que sirva de base para el desarrollo de nuevos algoritmos y la comparación entre distintos de ellos. Este framework en Python pretende incorporar tanto

la capacidad de simular contactos de distintos, la interoperabilidad entre distintos algoritmos y posteriormente la capacidad de evaluar los resultados mediante un conjunto de métricas disponibles.

2.3.2.4 Comparativa

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.

En cuanto a Stone Soup, su alcance parece similar, pero ambos proyectos tienen objetivos muy distintos. Stone Soup pretende generar una herramienta muy completa y detallada, con más variables y complejidad, sin ninguna interacción (únicamente vía código), lo cual le posiciona en una solución más cercana a las otras dos herramientas, pero de manera open-source.

El proyecto de este trabajo quizá no tendrá la misma exactitud al comienzo, pero pretende, además de sentar las bases para probar todos los componentes de un sistema de fusión multisensor, proporcionar una usabilidad y facilidad con la que realizar pruebas de manera rápida y sencilla que lo posiciona como otro tipo de producto distinto, enfocado a unos usuarios distintos.

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

Gestión del proyecto

Cualquier proyecto de ingeniería de una cierta seriedad debe definir el funcionamiento para el desarrollo a realizar. Este proyecto tiene unas características especiales como se explica en el resto del capítulo. Se comienza introduciendo la metodología de trabajo seguida durante todo el proyecto en la sección 3.1. Se continúa en la sección 3.2 con la organización de los recursos utilizados en todo el proyecto. En la sección 3.3 se introduce la planificación inicial del proyecto, así como el presupuesto asociado en la sección 3.4

3.1 Metodología de trabajo

Una metodología ajustada al ámbito del proyecto es muy importante para que el trabajo se realice de manera adecuada. Por ello, para este trabajo, al tener unas características muy específicas (desarrollo sin cliente, equipo de desarrollo de una persona, etc.) se va a utilizar una metodología muy concreta y distinta a lo habitual.

La metodología está basada en la metodología ágil Kanban. Esta se basa en dos conceptos que conforman su nombre Japonés. "Kan", significa visual y "ban" significa tarjeta. La metodología funciona gracias al orden de estas tarjetas en un tablón. Se ordenan las tarjetas que representan las tareas a realizar, en distintas columnas, que representan el estado de las mismas. De esta manera, en un rápido vistazo se puede obtener el estado del proyecto de manera global.

Este tipo de metodología favorece el desarrollo software en ciclos de desarrollo cortos y concisos, obteniendo entregas de manera continua. En este proyecto en concreto, el apartado de reuniones, muy importante en las metodologías ágiles se obvia, puesto que como se explica en la organización del proyecto (3.2), el proyecto lo realiza únicamente una persona.

A diferencia de un proyecto habitual, este proyecto tiene, como se verá posteriormente, dos bloques de desarrollo muy concretos, por lo que las tarjetas del segundo bloque quedan estancadas hasta el cumplimiento parcial

o total del primero. Esto lo asemeja ligeramente a una metodología Scrum con dos sprints.

3.2 Organización del proyecto

Este proyecto al ser parte de un trabajo de fin de máster, está realizado por una única persona. Este hecho limita la demostración de las capacidades de gestión de un proyecto software. A pesar de ello, el desarrollo del mismo se va a emular separando de manera clara los distintos roles que tendría un proyecto real, de manera que se pueda observar qué tareas realizaría cada uno y el tiempo dedicado a cada una.

Los roles necesarios para el desarrollo de este proyecto son:

- **Jefe de proyecto:** el principal responsable del proyecto. Se encarga de dirigir y gestionar el equipo de trabajo, priorizando y asignando tareas a los mismos. En este proyecto, además, apoyará ligeramente a la parte de diseño al tener conocimientos útiles para el proyecto.
- **Analista funcional:** es el encargado de obtener y comprender las necesidades del usuario final del software y con ellas desarrollar la ingeniería de requisitos asociada al producto.
- **Analista UX:** se centra en la interacción entre el software y el usuario. Para ello, analiza el perfil del usuario potencial y las tareas en las que se va a utilizar el software, desarrolla un prototipo que muestre la viabilidad del software. Evalúa las capacidades de usabilidad del software con un usuario real, diseña la interfaz de usuario y las pruebas a pasar sobre el software.
- **Programador:** utilizando los requisitos y pautas definidas por el analista, el programador desarrolla tanto los algoritmos definidos como la interfaz de usuario. Además, es el encargado de diseñar las pruebas y comprobar su funcionamiento.

3.2.1 Recursos humanos

El trabajo, como se ha comentado, lo realiza una única persona. A continuación, se adjunta una versión reducida de su currículum con fines documentales.

Nombre	Daniel Amigo Herrero
Edad	24 años
Nacionalidad	Española

Estudios	<ul style="list-style-type: none"> • Doctorando en Ciencia y Tecnología Informática. Universidad Carlos III de Madrid (2019 – actualidad) • Doble Máster de Ingeniería Informática y Ciencia y Tecnología Informática. Universidad Carlos III de Madrid (2017 – 2019) • Grado de Ingeniería Informática. Universidad Carlos III de Madrid (2013 – 2017) • Bachillerato en Ciencias y Tecnología. IES Valmayor (2011 – 2013) • Educación Secundaria Obligatoria. IES Valmayor (2007 – 2011) <p>Ayuda al máster universitario M02-1819 Universidad Carlos III de Madrid</p>
Experiencia laboral	<ul style="list-style-type: none"> • Apoyo a la docencia en las asignaturas Programación, TALF e IA. • Artículos admitidos en las conferencias FUSION 2019 y MSAW 2019. • Desarrollo de sistemas de fusión de datos para las fragatas F-110.

Tab.3.1: Currículo empleado

3.2.2 Recursos hardware

Para la ejecución de todas las tareas y roles se van a utilizar indistintamente tres equipos informáticos. A continuación se hace un despliegue de las características de cada uno:

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.3.2: Componentes ordenador sobremesa 1

Componente	Marca	Modelo	Capacidad
Procesador	Intel	i7 8700	6 núcleos
RAM	Corsair	DDR4 3000	32GB
Almacenamiento	Samsung	960 EVO	250GB
Tarjeta gráfica	Gigabyte	Vega 64	8GB

Tab.3.3: Componentes ordenador sobremesa 2

Componente	Marca	Modelo	Capacidad
Procesador	Intel	Celeron N3450	4 núcleos
RAM	Integrada	DDR3 1600	6GB
Almacenamiento	Samsung	960 EVO	128GB

Tab.3.4: Componentes portátil

3.2.3 Recursos software

Por último, se detallan los productos software necesarios para la elaboración de todo el proyecto, desde el desarrollo de este documento al control de tiempo trabajado.

3.2.4 Visual Studio Code + Latex

Visual Studio Code, el IDE gratuito de Microsoft, es la herramienta de desarrollo más popular en 2019. Esto es debido a su portabilidad, al estar disponible en todos los sistemas operativos y su alta capacidad de expansión, al permitir la inclusión de extensiones desarrolladas por la comunidad que amplían su funcionamiento.

Una de estas extensiones es LaTeX Workshop. Utilizada por más de 3.5 millones de usuarios, permite escribir y compilar documentos utilizando el lenguaje LaTeX. Esta alternativa a Microsoft Word es utilizada mayoritariamente en el ámbito académico e investigador, al ser multiplataforma, facilitar el uso de ecuaciones, ser altamente personalizable y gratuito. Su principal desventaja es ser bastante complejo su uso comparado con herramientas más sencillas.

3.2.5 Microsoft Visio

Para el apoyo a toda la documentación, diseño de algoritmos y flujo de trabajo, se utiliza la herramienta Microsoft Visio, muy orientada al diseño de diagramas software.

3.2.6 WireframePro

Para la generación de wireframes, prototipos rápidos para representar la idea del proyecto, se ha utilizado la herramienta de Mockflow, WireframePro. Este software online permite la creación de wireframes de una manera sencilla y rápida. No genera los prototipos más sofisticados, pero este desarrollo no los requiere.

3.2.7 MATLAB + App Designer

Para el desarrollo de todo el sistema de fusión de datos, tanto el back-end, los algoritmos y la gestión de la información, como el front-end, con la interfaz gráfica y ajustes de los algoritmos, se realizan en el lenguaje MATLAB. En concreto, se utiliza su entorno denominado App Designer, el cual permite la creación de aplicaciones finales utilizando MATLAB. Estas aplicaciones pueden ser compiladas en ejecutables o incluso funcionar como aplicaciones web.

3.2.8 TimeCamp + Trello

Para controlar toda la gestión del proyecto y cumplir con la metodología indicada, se van a utilizar dos herramientas. Para la gestión del tiempo de trabajo se utiliza TimeCamp. Esta herramienta multiplataforma permite controlar el tiempo de trabajo productivo, al detectar el programa que se está utilizando y el propio uso del ordenador. Para la gestión de tareas se utiliza Trello. Esta herramienta permite colocar distintas tareas en bloques, asignar miembros a tareas, colocar fechas límite, etc.

3.2.9 Microsoft Project

Finalmente, para una gestión más global del proyecto se utiliza Microsoft Project. Con esta herramienta se permite realizar la planificación inicial del proyecto, así como ir ajustando los plazos una vez se avanza el proyecto.

3.3 Planificación del proyecto

Como se ha comentado en la subsección , para hacer la planificación del proyecto se va a utilizar la herramienta Microsoft Project. Este proyecto se va a dividir en cinco bloques principales, como se puede observar en el diagrama Gantt de la figura Fig.3.1.

Cabe destacar que la estimación de tiempos es coherente a la cantidad de requisitos y la experiencia de proyectos anteriores del grupo de trabajo. Indicar también que todas las tareas se hacen tras la anterior, sin solaparse ninguna, puesto que aunque existan distintos roles, todos los ejerce una misma persona. Cada tarea del proyecto conlleva tanto el desarrollo de la misma como el registro de ese trabajo en la presente memoria.

Finalmente, indicar que en el capítulo de conclusiones 7 se muestra el cumplimiento de esta planificación inicial y los problemas que hayan podido surgir.

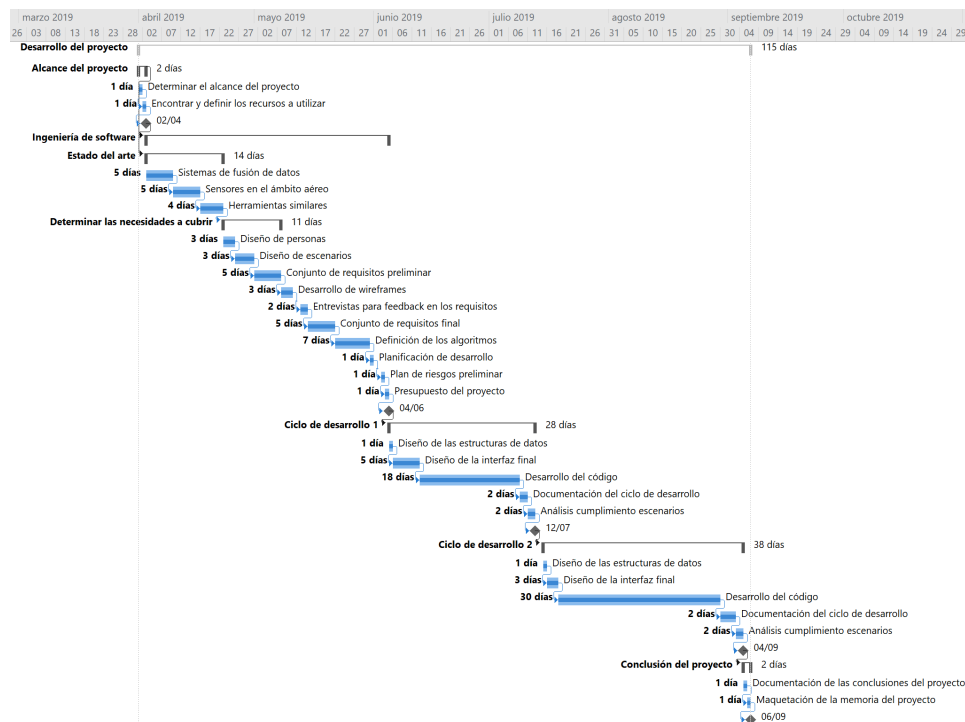


Fig.3.1: Planificación del proyecto

3.3.1 Bloques de la planificación

3.3.1.1 Alcance

El primer paso de todo proyecto es definir qué tarea se quiere desarrollar, por qué se realiza, qué necesidades busca saciar, etc. Sobre esta idea inicial, hay que acotar la misma con el objetivo de realizar un trabajo entregable en el plazo estimado. Esta labor recae exclusivamente en el Project Manager, así como identificar y encontrar los recursos a utilizar, tanto humanos como de trabajo.

3.3.1.2 Ingeniería de software

Una vez definido el alcance del proyecto, se procede a analizar todo el ámbito del proyecto, para identificar las necesidades de manera más específica y poder satisfacerlas de mejor manera. En este bloque intervienen los dos tipos de analistas y el Project Manager. El analista funcional define de manera concreta las necesidades a satisfacer con el proyecto, mientras que el analista UX genera artefactos persona que modelen potenciales usuarios de la aplicación. También diseña escenarios esperables de uso, y con ello y las funcionalidades, diseña la interfaz gráfica a grandes rasgos.

Posteriormente, se realizan unas entrevistas para validar el modelo de-

sarrollado y se concluye la idea de aplicación. Con esta idea fija y definida, se pasa a establecer el orden de desarrollo priorizando requisitos, con ellos se realiza esta planificación y el presupuesto necesario para todo el proyecto. También se realiza el plan de riesgos potenciales de este proyecto y cómo mitigarlos.

3.3.1.3 Ciclo de desarrollo 1 y 2

A continuación, se comienza el desarrollo del software. Como se ha indicado anteriormente, el mismo se descompone en dos proyectos separados, pero que dependen el uno del otro. Por ello, primero se realiza el que sirve de entrada al segundo.

El analista funcional define la estructura de los datos en la aplicación a desarrollar, así como la estructura de los ficheros CSV a exportar / importar. El analista UX define la interfaz final que debe tener la aplicación, mientras que el programador se ocupa de hacer funcionar todos los algoritmos con las restricciones que le han indicado.

Una vez desarrollado, se realiza la documentación de este componente y se realiza el testeo de la herramienta. Este utiliza los escenarios definidos anteriormente para observar si sacia las necesidades.

3.3.1.4 Conclusión del proyecto

Finalmente, con el software totalmente funcional y el trabajo terminado, se realizan unas conclusiones sobre el trabajo para próximos trabajos, así como identificar los fallos de planificación.

3.4 Presupuesto del proyecto

A partir de la planificación realizada, los roles del equipo de trabajo y los recursos materiales definidos, ya se cuenta con toda la información necesaria para establecer el presupuesto del proyecto.

Observando en la tabla Tab.3.5 la cantidad de días establecidos para cada día y el encargado de cada una, podemos obtener el número de horas trabajadas de cada rol.

Tarea	Duración	Desarrollada por
TOTAL	115 días	
Determinar el alcance del proyecto	1 día	Project Manager
Encontrar y definir los recursos a utilizar	1 día	Project Manager
Sistemas de fusión de datos	5 días	Analista UX

Tarea	Duración	Desarrollada por
Sensores en el ámbito aéreo	5 días	Analista Funcional
Herramientas similares	4 días	Project Manager
Diseño de personas	3 días	Analista UX; Project Manager
Diseño de escenarios	3 días	Analista UX; Project Manager
Conjunto de requisitos preliminar	5 días	Analista Funcional; Analista UX
Desarrollo de wireframes	3 días	Analista UX
Entrevistas para feedback en los requisitos	2 días	Analista UX; Project Manager
Conjunto de requisitos final	5 días	Analista Funcional; Analista UX
Definición de los algoritmos	7 días	Analista Funcional; Analista UX
Planificación de desarrollo	1 día	Todos
Plan de riesgos preliminar	1 día	Project Manager
Presupuesto del proyecto	1 día	Project Manager
Diseño de las estructuras de datos	1 día	Analista Funcional
Diseño de la interfaz final	5 días	Analista UX
Desarrollo del código	18 días	Desarrollador
Documentación del ciclo de desarrollo	2 días	Analista Funcional; Analista UX
Análisis cumplimiento escenarios	2 días	Analista Funcional; Analista UX
Diseño de las estructuras de datos	1 día	Analista Funcional
Diseño de la interfaz final	3 días	Analista UX
Desarrollo del código	30 días	Desarrollador
Documentación del ciclo de desarrollo	2 días	Analista Funcional; Analista UX
Análisis cumplimiento escenarios	2 días	Analista Funcional; Analista UX

Tarea	Duración	Desarrollada por
Documentación de las conclusiones del proyecto	1 día	Project Manager
Maquetación de la memoria del proyecto	1 día	Project Manager

Tab.3.5: Tareas de cada rol

Los sueldos de cada trabajador se han establecido a partir del sueldo medio en España de cada rol, disponibles en Internet. Todos estos sueldos han sido obtenidos mediante encuestas con una muestra bastante amplia.

Rol	Días trabajados	Sueldo anual medio	Sueldo por día trabajado ¹	Total
Project Manager	19 días	25.972€ ²	113.41€	2154.79€
Analista UX	49 días	30.000€ ³	131.00€	6419.00€
Analista funcional	32 días	32.858€ ⁴	143.48€	4591.36€
Desarrollador	48 días	20.247€ ⁵	88.41€	4243.68€
TOTAL				17408.83€

Tab.3.6: Sueldo por rol

En cuanto a los recursos hardware, se calcula la amortización de los dispositivos mostrados anteriormente (3.2.2), sobre su Precio de Venta al Público (PVP):

Hardware	PVP	Tiempo de uso	Periodo de depreciación	Amortización
Ordenador portátil	250€	3.5 meses	60 meses	14.58€
Ordenador sobremesa 1	1000€	3.5 meses	60 meses	58.33€

¹Considerando la jornada de trabajo máxima anual de 1826 horas y 27 minutos que marcan las CCOO, son aproximadamente 229 días de trabajo

²Sueldo extraído de: <https://www.indeed.es/salaries/Gestor/a-de-proyectos-Salaries/>

³Sueldo extraído de: <https://www.uifrommars.com/salarios-diseno-espana-2018/>

⁴Sueldo extraído de: <https://www.indeed.es/salaries/Analista-de-sistema-Salaries/>

⁵Sueldo extraído de: <https://www.indeed.es/salaries/Programador/a-junior-Salaries/>

Hardware	PVP	Tiempo de uso	Periodo de depreciación	Amortización
Ordenador sobremesa 2	2000€	3.5 meses	60 meses	116.66€
TOTAL				189.57€

Tab.3.7: Amortización por recurso hardware

Por otro lado, se cuenta con el coste de las licencias software para hacer posible el proyecto. Estas licencias son o mensuales o anuales, por lo que se calcula el importe necesario para cada una.

Software	Coste	Tiempo de uso	Coste total
Windows 10	0 €	3.5 meses	0€
Microsoft Visio	5€ / mes	3.5 meses	17.5€
WireframePro	gratis	3.5 meses	0€
Microsoft Project	7€ / mes	3.5 meses	24.5€
Google Drive	1.99€ / mes	3.5 meses	5.97€
TimeCamp	7.50€ / mes	3.5 meses	26.25€
Trello	gratis	3.5 meses	0€
Visual Studio Code + Latex	gratis	3.5 meses	0€
MATLAB + App Designer	800€ / año	3.5 meses	800€
TOTAL			931.22€

Tab.3.8: Coste por recurso software

Por último, se calculan los costes indirectos sobre el tiempo de desarrollo en el lugar destinado al mismo.

Concepto	Coste	Tiempo de uso	Coste total
Luz	40€ / mes	3.5 meses	140€
Agua	15€ / mes	3.5 meses	52.5€
Internet	40€ / mes	3.5 meses	140€

Concepto	Coste	Tiempo de uso	Coste total
Transporte	20€ / mes	3.5 meses	70€
TOTAL			402.5€

Tab.3.9: Costes indirectos

Con todos los costes del proyecto listados, se calcula el importe total del mismo, estableciendo el presupuesto final, con un 15% de margen de beneficios.

Concepto	Coste
Salarios	17408.83€
Recursos hardware	189.57€
Recursos software	931.22€
Costes indirectos	402.5€
TOTAL SIN IVA	18932.12€
TOTAL CON IVA	22907.87€
Con margen de beneficios	26344.05€

Tab.3.10: Presupuesto final

El coste total del proyecto en **ventiseis mil trescientos cuarenta y cuatro euros con cinco céntimos**.

Capítulo 4

Alcance y usabilidad del software

Este capítulo recoge todas las características que debe comprender el producto software a desarrollar, así como el proceso para obtenerlas. El analista funcional define una lista inicial de requerimientos funcionales, a partir del alcance del software 4.1. Con esta lista, el analista UX genera los artefactos persona 4.2 y escenarios 4.3, para representar los usuarios potenciales de la herramienta. El analista UX desarrolla unos simples prototipos y diseña unas entrevistas, para obtener y contrastar el proyecto con usuarios que pueden aportar información al proyecto. A continuación, con los resultados de la validación, los analistas refinan los requerimientos mostrados en el apartado 4.6. Por último, se definen el plan de riesgos de este proyecto de ingeniería 4.7 y el marco regulador que afecta al mismo 4.8

4.1 Alcance del software

El alcance del software a desarrollar es una de las primeras aproximaciones que se deben realizar en un proyecto software. En este se definen las necesidades, de manera a priori, a saciar con esta herramienta.

El software a desarrollar en este proyecto tiene como objetivo principal pasar a ser la herramienta de referencia de todo investigador en sistemas de fusión, tanto novatos como más expertos. Para ello, la herramienta pretende abarcar todos los aspectos relacionados con un sistema de fusión, tanto la generación de contactos de los sensores, la optimización de todos los componentes de un sistema de fusión y la observación de los resultados y comparación entre distintas aproximaciones.

Con este objetivo, para generar los contactos se ha de implementar un sistema para generar trayectorias de vehículos, modelar los distintos sensores que producen los contactos y desarrollar un algoritmo que genere estos contactos.

Por otro lado, para simular los sistemas de fusión, se han de implementar tantos componentes de un sistema de fusión como sean posibles, y de cada componente sus distintos algoritmos. Además, generar un algoritmo

que permita el funcionamiento del sistema según los componentes deseados. Destacar que se cuentan con una serie de algoritmos ya desarrollados y validados de los componentes estimación y asociación, de manera que la carga de desarrollo se reduce. Por último, se tiene que desarrollar un sistema de comparación entre los resultados de cada prueba de un sistema de fusión con una interfaz común.


Todo lo explicado anteriormente se ha de implementar en herramientas con interfaces gráficas, las cuales sean usables y faciliten el funcionamiento a los investigadores. El software debe contar con una documentación y limpieza de código adecuada para facilitar tanto su uso como su ampliación en el futuro.

4.2 Artefactos persona

Uno de los aspectos más importantes a la hora de diseñar una aplicación que va a utilizar un cierto tipo de usuario concreto es analizar qué tipo de interacción va a tener con la misma. Para ello, siguiendo las técnicas aprendidas en el máster, se van a diseñar distintos artefactos persona, que representen a cada segmento potencial de usuarios de la aplicación. Conociendo las necesidades del software a desarrollar definidas en el alcance, se definen los siguientes persona:

4.2.1 Personas


4.2.1.1 Persona 1

Nombre	Javier Pérez	
Edad	22 años	
Nacionalidad	Española	
Sexo	Hombre	
Estudios	Estudiante de Ingeniería Informática	
Ocupación	Estudiante de Ingeniería Informática	
Acerca de mí	Me gusta la tecnología, soy muy activo y me gusta aprender, conocer y compartir todo tipo de retos de programación.	
Relación con los sistemas de fusión	No entiendo muy bien cómo funciona un sistema de fusión ahora mismo. Según mi tutor, es un sistema muy complejo y no he visto ninguno funcionando. Ahora mismo tengo que generar contactos y no conozco ninguno de los sensores.	

Necesidades e inquietudes	Estoy haciendo un TFG con un profesor y me ha dicho que genere los contactos recibidos de varios aviones por una estación aérea. No encuentra ninguna herramienta ni ayuda con la que comenzar
----------------------------------	--


Tab.4.1: Persona 1

4.2.1.2 Persona 2

Nombre	Luisa Gómez	
Edad	39 años	
Nacionalidad	Española	
Sexo	Mujer	
Estudios	Ingeniería Aeronáutica	
Ocupación	Profesora de universidad	
Acerca de mí	Me encanta la docencia. Estudio los sistemas de detección de aviones y cómo afectan a la trayectoria del resto de tráfico aéreo	
Relación con los sistemas de fusión	Bastante profunda. Junto a mis alumnos estudio principalmente el nivel 2, las características del escenario en el que se encuentra el avión en un instante	
Necesidades e inquietudes	Me gustaría introducir a mis alumnos a mis proyectos aéreos, pero es complejo explicarles y que tengan los conocimientos a tiempo	

Tab.4.2: Persona 2

4.2.1.3 Persona 3

Nombre	Luke Adams	
Edad	57 años	
Nacionalidad	Inglesa	
Sexo	Hombre	
Estudios	Ingeniería Aeronáutica	
Ocupación	Jefe de proyectos aeronáuticos para una multinacional	
Acerca de mí	Llevo 30 años dedicado a desarrollar los sistemas internos de varios modelos de aviones de mi empresa.	
Relación con los sistemas de fusión	Los desarrolla mi equipo. Desde procesar los contactos de los sensores hasta predecir qué maniobra es la más adecuada.	

Necesidades e inquietudes	Lo más complejo es gestionar tantas facetas a la vez, son muchos empleados y cada uno trabaja de manera distinta. También me cuesta entenderme con los superiores, que no están puestos en la faceta más técnica. Ellos solo quieren resultados y beneficios.
----------------------------------	---

Tab.4.3: Persona 3

4.3 Escenarios

Una vez tenemos los persona generados, se diseñan los escenarios que muestran la necesidad de dicho usuario de una herramienta como propuesta en este trabajo.

4.3.0.1 Escenario 1

Javier se encuentra leyendo sobre sistemas de fusión de datos, los radares y sensores que funcionan en el ámbito aéreo para su trabajo de fin de grado y se encuentra perdido. Ve muy difícil entender a tiempo el funcionamiento para después pensar de qué manera simular todos los objetos que debe conocer para conseguir los contactos que son su objetivo. Desea que existiera una herramienta o cualquier guía para comenzar con el trabajo, pero no la encuentra. Acaba desistiendo y haciendo otro trabajo distinto.

4.3.0.2 Escenario 2

Luisa propone trabajos de fin de grado en el tablón de su universidad, pero los alumnos no lo escogen. Debido a ello, debe hacer la mayoría de sus pruebas por su cuenta, pero no se le da demasiado bien programar (no es lo que estudió) ni tiene el tiempo para ello. Le gustaría tener una herramienta para generar las pruebas de una manera más sencilla, porque le es muy complicado.

4.3.0.3 Escenario 3

Luke y su equipo hacen un buen trabajo con los sistemas de fusión, la empresa está contenta con ello. El problema que tiene Luke es que no le es sencillo gestionar a todo su equipo y que compartan la información de las pruebas que realizan. Como cada uno se centra en un componente de los sistemas de fusión, no comparten el mismo código, lo cual genera problemas a la hora de generar los sistemas finales. Lamenta no haber introducido un sistema común y revisiones del mismo para mantenerlo todo actualizado. Ahora mismo cada empleado es imprescindible, solo el entiende su código.

4.3.0.4 Escenario 4

Luke tiene otro problema. Los trabajos los realiza su equipo, pero es él quien los presenta a los superiores. Los superiores tratan de entender si la propuesta es interesante, pero su falta de los conocimientos técnicos le impiden comprender si el trabajo aporta una utilidad real a sus aviones. El equipo de Luke trata de hacer explicaciones lo más simplificadas posible, pero habitualmente las terminan rechazando por no considerarlas demasiado interesantes, cuando realmente sí lo son. Luke desearía tener un método para enseñar de manera más real a los superiores el funcionamiento de las mejoras propuestas y su comparación con los sistemas actuales.

4.4 Wireframes

A partir del alcance definido para el proyecto, el analista UX ha generado los siguientes wireframes para comprender de manera sencilla si la propuesta es válida o no. Los wireframes, como se comentó anteriormente, se definen con la herramienta WireframePro.

Se cuenta con dos vistas distintas para cada aplicación del proyecto. La primera, para generar los contactos a utilizar en la segunda, que genera tantas pruebas como se desee sobre un sistema de fusión de datos.

4.4.1 Wireframe generador de contactos

Como se puede observar en la imagen Fig.4.1, se cuenta con una gestión de blancos en la barra superior, para generar tantos como se deseen. Se cuenta con un botón de ajuste de la aplicación, para parámetros que afecten únicamente a la interfaz (idioma, resolución, etc.).

La gestión del algoritmo de generación de trayectorias se incorpora en los parámetros de la izquierda, mientras que a su derecha se tiene una función para seleccionar qué se quiere observar en el mapa de apoyo (ver la figura Fig.4.2). Por último se cuenta con los botones de importar y exportar la simulación, que permite volver al instante de la generación actual en cualquier momento. También se tienen los botones de comenzar la simulación, y una vez terminada, poder guardar sus resultados.

Cabe destacar que en el mapa anteriormente mencionado se pintarán todas las trayectorias generadas, por cada sensor y blanco, para tener el control total de la simulación.

Si pasamos a la segunda vista, la vista de un blanco concreto Fig.4.3, vemos a la izquierda como se permite generar puntos de paso del blanco (waypoint) para generar su trayectoria. Estos puntos se seleccionarán directamente sobre el mapa, pero se podrán ajustar mediante los parámetros que tienen debajo. Como es lógico, se cuenta con una gestión de los puntos de paso, generando nuevos, editando y borrando si se desea.



Fig.4.1: Wireframe 1 generador de contactos

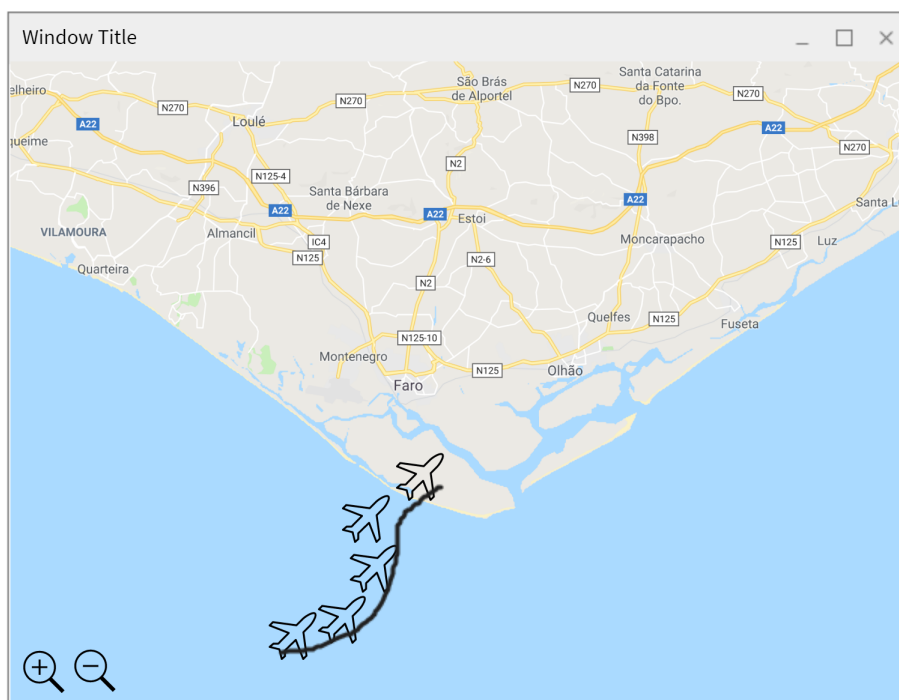


Fig.4.2: Wireframe mapa auxiliar

En la parte derecha se muestran los sensores. Se permite seleccionar, para cada sensor, si se utiliza el sensor y sus parámetros para ajustarlo a las

necesidades de cada blanco.

También se permite importar y exportar únicamente el blanco, con sus puntos de paso y sensores, de manera que reutilizarlos en otros escenarios.

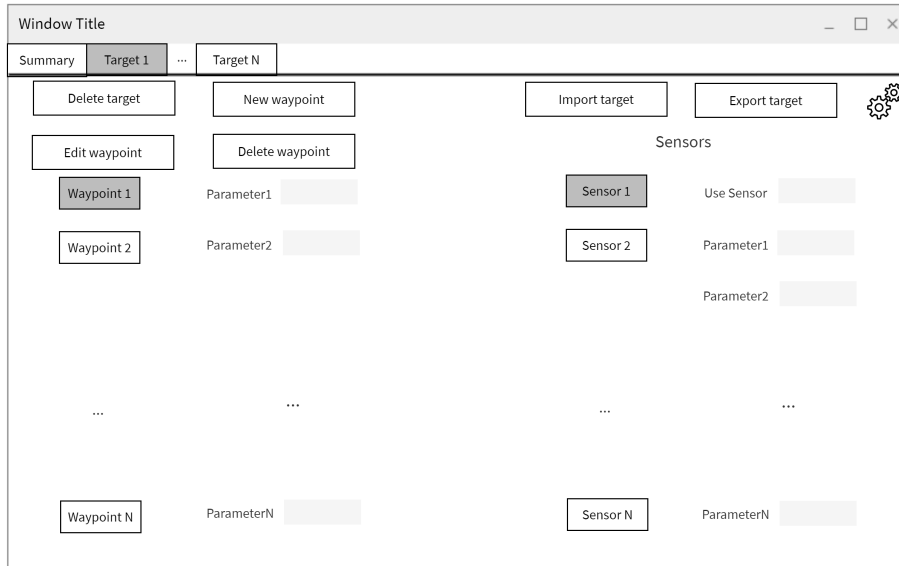


Fig.4.3: Wireframe 2 generador de contactos

4.4.2 Wireframe sistema de fusión

En cuanto a la otra herramienta, la destinada a generar pruebas sobre un sistema de fusión, tiene una interfaz similar en su vista inicial. La variante se produce con los tests, las pruebas sobre un sistema de fusión. Se pueden generar tantas como se deseen, y se customizan en su propia pestaña en la figura Fig.4.5. También se permite importar y exportar el estado total de la herramienta en cualquier instante, como en el generador de contactos.

Una vez encima de una prueba (Fig.4.5), se permite seleccionar sobre cada componente de un sistema de fusión (estimador, asociador, etc.) si se desea utilizar en esta prueba. A la derecha, en caso de tener varios algoritmos disponibles para un mismo componente, se puede seleccionar el deseado. Cada algoritmo y componente tiene sus propios parámetros a ajustar.

También se cuenta arriba con botones para eliminar esta prueba actual e importar o exportar la prueba concreta.

4.5 Validación mediante entrevistas

La ingeniería de software es uno de los pasos más cruciales de un proyecto software, puesto que es imprescindible plasmar correctamente las necesidades

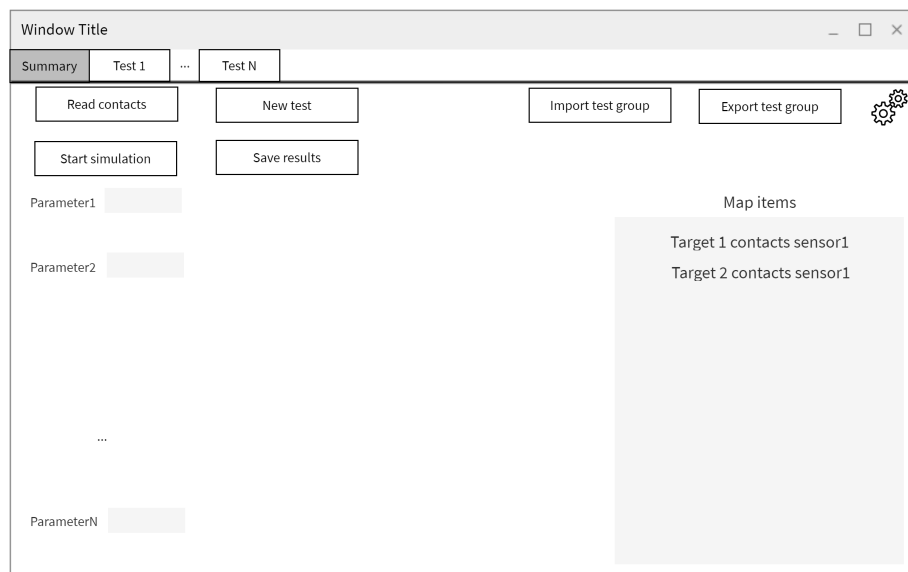


Fig.4.4: Wireframe 1 sistema de fusión

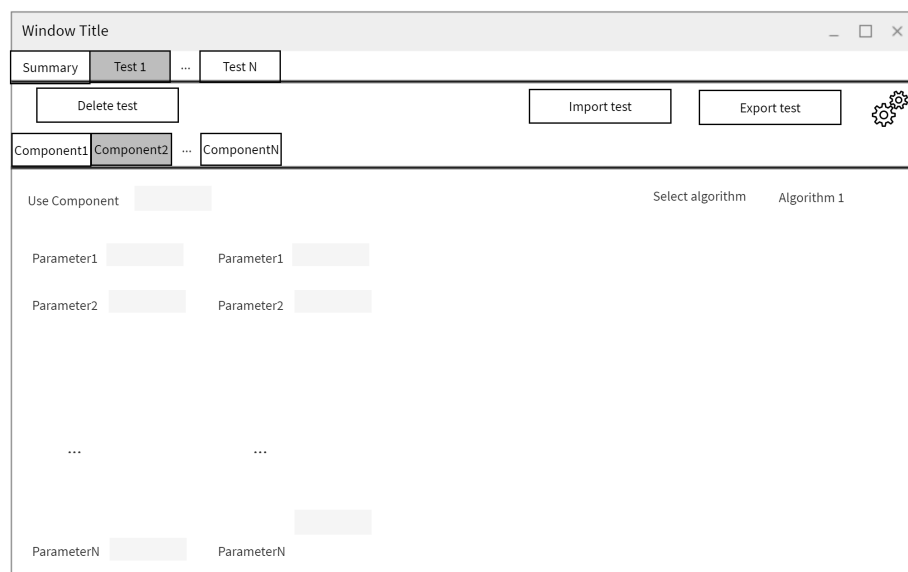


Fig.4.5: Wireframe 2 sistema de fusión

del cliente o usuario para satisfacerle con el producto a desarrollar. Al no contar con un cliente concreto en este proyecto, pero ser un software orientado a investigadores en el ámbito de fusión de datos para vigilancia, se han realizado entrevistas a usuarios potenciales de la aplicación para obtener sus necesidades y poder paliarlas con esta herramienta.

En el anexo A se muestran tanto las preguntas realizadas como las tran-

scripciones de sus perfiles y respuestas. Todas estas entrevistas han servido para matizar los requisitos definidos en la sección siguiente.

4.6 Identificación de requisitos

Una vez se tienen definidas las necesidades del cliente, el siguiente paso es definir los requerimientos necesarios para cumplir con todas las necesidades impuestas. Además, estos requerimientos deben ser claros y explícitos, al ser uno de las bases a utilizar por el programador.

Existen varios tipos de requerimientos, en función del enfoque utilizado:

- Requisitos funcionales: qué debe hacer el software. Sus funcionalidades.
- Requisitos de datos: cuál es el formato de los datos que utiliza el software. EJ: precisión, unidades, etc.
- Requisitos de entorno: de qué manera se utiliza el software. Tanto físicamente, como de interacción.
- Requisitos de usuario: características que tiene cada usuario potencial del software y su incidencia para el mismo.
- Requisitos de usabilidad: aspectos relacionados con las metas definidas para este software en términos de usabilidad.

Identificador	Identifica de manera unívoca a cada requisito, mediante un acrónimo y un número X. Los acrónimos son: <ul style="list-style-type: none"> • RF.X: Requisito funcional. • RD.X: Requisito de datos. • RA.X: Requisito de ambiente. • RU.X: Requisito de usuario. • RX.X: Requisito de usabilidad.
Nombre	Indica con dos o tres palabras el contenido del requisito.
Descripción	Contiene una descripción detallada sobre el requisito.
Razón	Justifica la necesidad de la creación de este requisito.

Tab.4.4: Plantilla requisitos

Para facilitar la lectura del documento, se añaden los requisitos en el anexo B.

4.7 Plan de riesgos

Un trabajo de ingeniería como este, de una duración elevada y en el que de su éxito depende el futuro de varias personas requiere tener controlados todos los parámetros y variables que puedan poner en peligro su correcta culminación. Para ello, se define en un documento auxiliar, denominado plan de riesgos, todos los posibles ataques o desastres que puedan mermar la continuidad del proyecto, provocando una pérdida o daño tanto hardware, software o personal.

La Organización Internacional de Normalización (ISO) define, mediante su norma ISO 31000, principios y directrices para la elaboración de un plan de riesgos adecuado. Siguiendo las mismas, se ha definido un plan de riesgos adaptado a este proyecto, el cual se compone varias directrices. A continuación, se van a exponer los riesgos identificados, obviando las directrices que no afectan al mismo.

4.7.1 Riesgos encontrados

Para la definición de todos los riesgos se utiliza la siguiente plantilla, incluyendo tanto el riesgo en sí como la manera en la que se pueden mitigar sus efectos, es decir, un plan de contingencia reducido:

Identificador	Identifica de manera unívoca a cada requisito, mediante el siguiente acrónimo: Ri.X, siendo X el número de este riesgo.
Nombre	Indica con dos o tres palabras el contenido del riesgo.
Descripción	Contiene una descripción detallada sobre el riesgo.
Probabilidad de ocurrencia	Se establece la probabilidad de que este riesgo ocurra y afecte al proyecto. Escala: Improbable, poco probable, probable, muy probable
Impacto sobre el proyecto	En caso de que suceda este riesgo, de qué manera afecta a la ejecución planificada del proyecto.
Mitigación del riesgo	Se expone la solución con la que remediar este riesgo, de manera activa o pasiva

Tab.4.5: Plantilla definición de riesgos

4.7.1.1 Riesgos relativos al alcance

Identificador	Ri.1
Nombre	Requisitos no suficientemente definidos

Descripción	Los requisitos deben ser concretos y específicos, para que el desarrollador realice el trabajo deseado y cumpla las necesidades buscadas
Probabilidad de ocurrencia	Probable
Impacto sobre el proyecto	Provocaría que el resultado del software no cumpla las expectativas, necesitando de ampliar el periodo de ingeniería de software para mejorar el producto
Mitigación del riesgo	Los requisitos serán validados por el desarrollador una vez terminados, para asegurar que entiende las necesidades exigidas por los analistas

Tab.4.6: Riesgo 1

4.7.1.2 Riesgos relativos a las adquisiciones

Identificador	Ri.2
Nombre	Corte eléctrico
Descripción	Un corte del suministro eléctrico en el lugar de trabajo afecta a la continuidad del trabajo de cualquier parte del proyecto
Probabilidad de ocurrencia	Poco probable
Impacto sobre el proyecto	Si se corta la electricidad se puede romper cualquier componente eléctrico de los ordenadores. Igualmente, durante el tiempo de apagón no se podría realizar ningún trabajo, afectando directamente a la planificación
Mitigación del riesgo	Se tendrán todos los ordenadores preparados con la última copia de seguridad realizada, y el portátil con batería en caso de que el corte sea prolongado

Tab.4.7: Riesgo 2

Identificador	Ri.3
Nombre	Rotura del ordenador
Descripción	Los ordenadores pueden romperse y son la principal herramienta de trabajo de este proyecto, ya sea por cortocircuito o por caída del mismo

Probabilidad de ocurrencia	Poco probable
Impacto sobre el proyecto	Al tener varios de ellos, cualquier rotura no limitaría en exceso la continuación del trabajo, pero puede tener un ligero impacto en el presupuesto
Mitigación del riesgo	Se tendrán todos los ordenadores preparados con la última copia de seguridad realizada, y el portátil con batería en caso de que el corte sea prolongado

Tab.4.8: Riesgo 3

Identificador	Ri.4
Nombre	Corte de Internet
Descripción	En este caso es el suministrador de Internet quien tiene una avería que evite la posibilidad de crear copias de seguridad en Google Drive, o realizar consultas
Probabilidad de ocurrencia	Poco probable
Impacto sobre el proyecto	No generar copias de seguridad aumenta la probabilidad de pérdida del trabajo realizado y no poder realizar consultas limita la velocidad de desarrollo
Mitigación del riesgo	Se tendrán todos los ordenadores preparados con la última copia de seguridad realizada, y el portátil con batería en caso de que el corte sea prolongado

Tab.4.9: Riesgo 4

4.7.1.3 Riesgos relativos al alcance

Identificador	Ri.5
Nombre	Falta aptitudes personal
Descripción	Si algún rol del equipo de trabajo no tiene las capacidades adecuadas para su trabajo asignado
Probabilidad de ocurrencia	Probable

Impacto sobre el proyecto	Cualquier aspecto desconocido para un rol afecta al tiempo esperado de trabajo, afectando directamente a la planificación. Al tratarse de conceptos muy complejos de algoritmia, pueden tener un alto impacto
Mitigación del riesgo	Todos los roles tendrán un tiempo adecuado para adaptarse al proyecto, antes de comenzar con el trabajo "real"

Tab.4.10: Riesgo 5

4.7.1.4 Riesgos relativos a la comunicación

Identificador	Ri.6
Nombre	Falta de actualización cada rol
Descripción	Este riesgo representa la falta de comunicación, de compartir información o de actualización del tablón por parte de cualquier rol
Probabilidad de ocurrencia	Poco probable
Impacto sobre el proyecto	Una baja profesionalidad en un proyecto software en grupo como este y con una planificación muy específica afectaría directamente a la planificación general
Mitigación del riesgo	El jefe de proyecto estará pendiente de facilitar la comunicación entre todos los roles, solventando cualquier problema que surja

Tab.4.11: Riesgo 6

4.7.1.5 Riesgos relativos al coste del proyecto

Identificador	Ri.7
Nombre	Presupuesto mal estimado
Descripción	Un presupuesto demasiado ajustado puede comprometer a todo el proyecto
Probabilidad de ocurrencia	Poco probable
Impacto sobre el proyecto	Si el proyecto se ha planificado y presupuestado de manera inadecuada, puede afectar a la consumación final del trabajo

Mitigación del riesgo	El presupuesto cuenta con un porcentaje para solventar posibles variaciones en la planificación a medida que avance el tiempo
------------------------------	---

Tab.4.12: Riesgo 7

4.7.1.6 Riesgos relativos a la integración del proyecto

Identificador	Ri.8
Nombre	Mala calidad en la documentación
Descripción	La documentación del software es muy importante para comprender el software desarrollado
Probabilidad de ocurrencia	Poco probable
Impacto sobre el proyecto	Una mala documentación afectaría a la necesidad de rehacerla, o perjudicaría la capacidad de ampliación del software en el futuro
Mitigación del riesgo	Cada apartado de la documentación, realizado por un rol, deberá ser revisado y comprendido por otro miembro del equipo, garantizando su comprensión desde cualquier punto de vista

Tab.4.13: Riesgo 8

4.8 Marco regulador

Finalmente, es necesario indicar la legislación que aplica sobre una aplicación como esta.

El software a desarrollar es una herramienta que no cuenta con ningún tipo de seguimiento ni datos recolectados. Por tanto, no requiere de la aplicación del Reglamento General Europeo de Protección de Datos (GDPR), al no afectar de ninguna manera a la privacidad o los derechos humanos de las personas de la Unión Europea. En una posible expansión de la herramienta sí podría aplicarse algún tipo de normativa de protección de datos, pero actualmente no es el caso.

El otro tipo de legislación que puede afectar a un proyecto como este es la protección del software desarrollado. Al desarrollar una herramienta software desde cero, esta se legisla mediante el Real Decreto Legislativo 1/1996, de 12 de abril, en el cual se aprueba la Ley de Propiedad Intelectual [35]. En esta ley, en el Libro I Título VII "Programas de Ordenador", se indica que el autor u autores de un software son sus propietarios, sin tener necesidad de registrarlos o patentarlos.

Además, al haberse desarrollado bajo el marco del Trabajo de Fin de Máster de la Universidad Carlos III de Madrid, es necesario observar cuál es el reparto que realiza la universidad sobre la propiedad intelectual. Como se puede observar en esta entrada de la universidad¹, la titularidad de los derechos son íntegramente del creador de la misma, a no ser que sea 'personal de la Universidad'.

¹<https://www.uc3m.es/biblioteca/propiedad-intelectual-industrial/propiedad-intelectual/consultas-resueltas/derechos-explotacion>

Capítulo 5

Diseño del sistema

En este capítulo se expone el sistema desarrollado en profundidad, una vez definidas todas las características del mismo. En la sección 5.1, se explica la herramienta utilizada para generar la interfaz, puesto que tiene características peculiares que es interesante resaltar para comprender el trabajo. Después, se explican de manera separada ambas aplicaciones, primero el generador de contactos 5.2 y después la aplicación para generar pruebas sobre el sistema de fusión 5.3.

5.1 Introducción

5.1.1 Introducción MATLAB App Designer

Como se ha comentado, la aplicación, tanto los algoritmos como la interfaz, como la estructura de datos, se implementa en el mismo código, en MATLAB App Designer. Esta implementación tiene sus ventajas y sus inconvenientes. Como ventajas principalmente existe la capacidad de aprovechar todas las funcionalidades de MATLAB, así como la estructura de datos que ya está funcionando. No utilizar la interfaz de MATLAB implicaría desarrollar otra estructura auxiliar en la interfaz deseada. De esta manera, se añade mucho más trabajo y complicación a una interfaz bastante compleja de por sí. Además, habría que relacionar ambos componentes, la "librería de código de MATLAB" con la interfaz.

Por otro lado, utilizar otro lenguaje para la interfaz proporcionaría más versatilidad de componentes para la interacción, puesto que MATLAB App Designer cuenta con menos de 30 objetos distintos y predeterminados para toda la interfaz ¹ Esto limita en exceso la funcionalidad y usabilidad que se le puede otorgar a una herramienta. Igualmente, es una adición en constante evolución por parte de MATLAB.

¹es.mathworks.com/help/matlab/creating_gui/choose-components-for-your-app-designer-app.html

De todos los componentes disponibles, la interfaz se ha resuelto únicamente con 6 de ellos. El más importante es el *uitab* y *uitabgroup*. Ellos dos son los encargados de generar pestañas, en donde se han implementado de manera general los blancos y las pruebas, y también dentro de cada blanco la serie de sensores disponibles. MATLAB App Designer permite generar la interfaz de manera manual, arrastrando y colocando los componentes "a ojo" (como se muestra en la figura Fig.5.1), pero si quieres hacer que se añada automáticamente un conjunto de objetos, como en estos casos con los botones de "Nuevo Blanco" o "Nueva prueba", debes escribir todo el código para ello dentro de un botón.

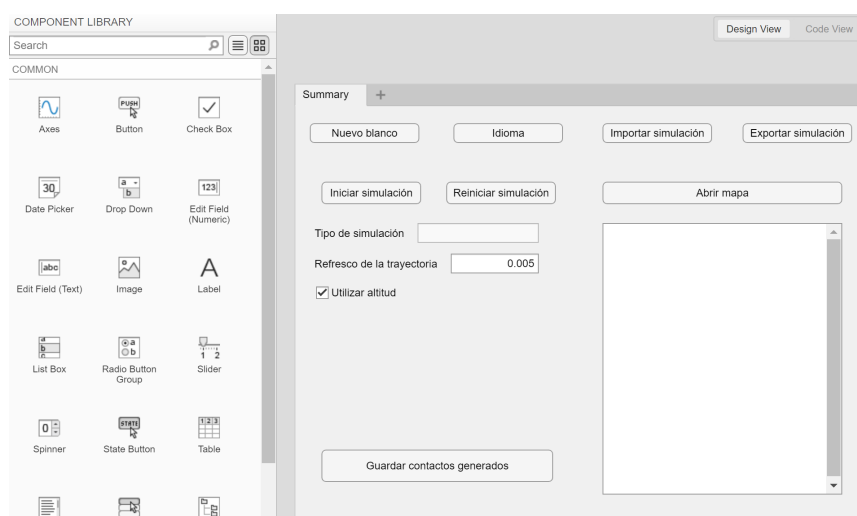


Fig.5.1: Desarrollando interfaz a mano

5.1.2 Componentes implementados en ambas aplicaciones

A continuación, se van a exponer algunos elementos desarrollados cuyo trabajo ha sido reseñable. Todos estos componentes están implementados en ambas aplicaciones.

5.1.2.1 Mensaje de progreso

En la figura Fig.5.1 se puede observar que no está ningún blanco incluido, por lo que en cada momento se tiene que generar toda la interfaz y toda la estructura de datos, de cero. Este proceso lleva su tiempo (así como otros), por lo que se ha implementado tanto para este caso, como para otros algoritmos de alto coste computacional, una función para mostrar mensajes de progreso, como se puede observar en la figura Fig.5.2.

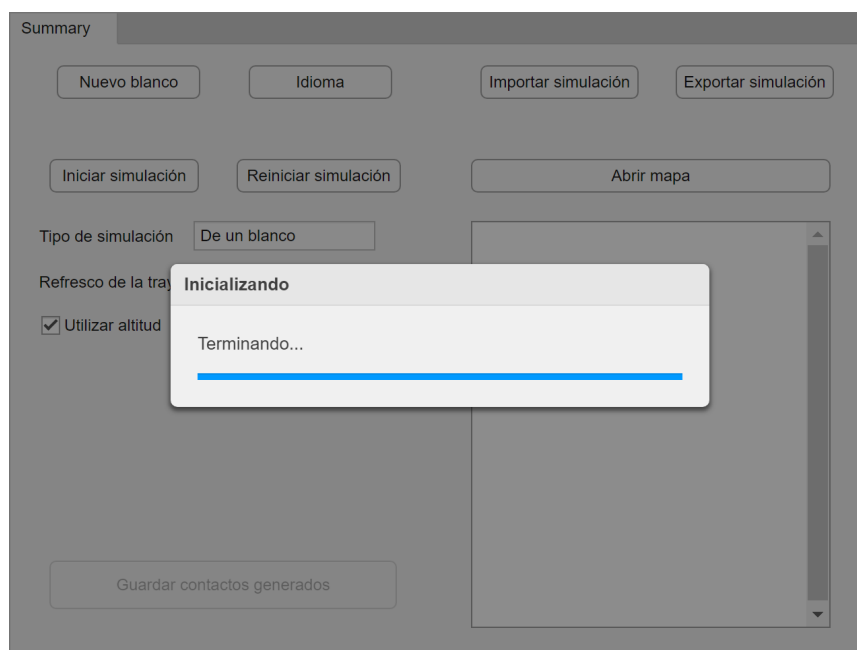


Fig.5.2: Mensaje de progreso en la interfaz

5.1.2.2 Mensaje de error

Se ha desarrollado también un mensaje de error visual, el cual aporta información al usuario del error producido y cancela la ejecución (ver figura Fig.5.3). Estos se introducen tanto al comienzo de los algoritmos principales de cada aplicación, como al tratar de ejecutar una acción no permitida, como eliminar la estación fija, o añadir varios puntos de paso en una estación fija.

5.1.2.3 Interfaz lectura/escritura ficheros

Como se ha comentado previamente, es importante la escritura de información en ficheros con unas características muy concretas. Por ello, se ha desarrollado una interfaz de lectura y escritura de ficheros, basada en ficheros CSV, tanto para exportar / importar el estado actual, como salvar los resultados. Debido a que el número de objetos (y sus dimensiones) a tratar varía en función de las necesidades del usuario, antes de escribir cada objeto se introduce el número de entradas a leer de dicho tipo. En la figura Fig.5.4 se muestra un extracto de un fichero exportado directamente de la aplicación.

Toda esta información, como se ve, se separa mediante punto y coma, siendo la unidad de decimales la coma. Esta implementación es la adecuada para abrir los ficheros con un editor de hojas de cálculo (como Microsoft Excel)

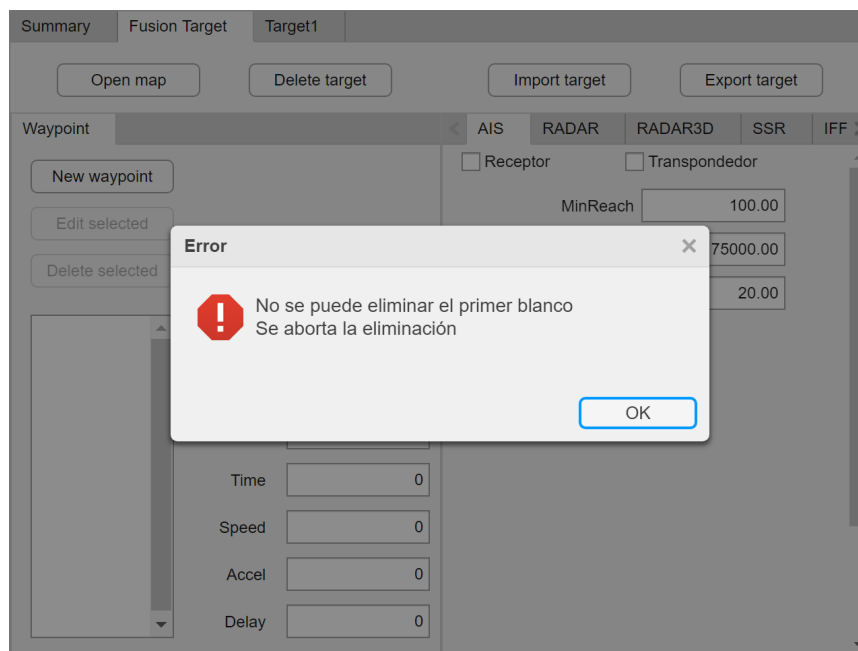


Fig.5.3: Mensaje de error en la interfaz

en España, pero en otros idiomas se utilizan otros delimitadores. Estos son customizables en el botón de ajustes.

5.1.2.4 Mensajes de explicación parámetros

Otra de las funciones comunes es la capacidad de autoexplicación de los distintos parámetros a utilizar. Gracias a los tooltip, como el que se muestra en la figura Fig.5.5, se puede incorporar de manera poco intrusiva la información justa para que cualquier usuario entienda la utilidad del parámetro dentro de su algoritmo. Además, si se trata de un valor, se añade un tooltip sobre el parámetro indicando las unidades de dicho valor.

5.1.2.5 Funcionamiento del mapa

Por último, se va a explicar el funcionamiento del mapa implementado. MATLAB es una herramienta muy potente en el aspecto de generar gráficos, siendo uno de ellos los mapas geográficos. Aprovechando los mismos, y la alta gestión de las estructuras de datos implementadas, se generó el panel de la derecha en la pestaña de resumen. En esta, se pueden seleccionar todos los objetos deseados a mostrar en el mapa, como se puede observar en la figura Fig.5.6.

Además de la utilidad de facilidad implementada gracias a distintos colores según el blanco, y distintas figuras según el tipo de punto (círculo los

```

typeScenario; 2;
FusionCenter_ID; 1;
FusionCenter_numLines; 130;
FusionCenter_Time; 0,0050; 0,5050; 1,0050; 1,5050; 2,0050; 2,5050; 3,0050; 3,5050;
FusionCenter_Latitude; 36,795984; 36,796064; 36,796168; 36,796296; 36,796448; 36,7
FusionCenter_Longitude; -9,519035; -9,518846; -9,518599; -9,518296; -9,517937; -9,
FusionCenter_Altitude; -0,00000; -0,00154; -0,00604; -0,01343; -0,02363; -0,03652;
FusionCenter_SpeedX; 28,61556; 38,95647; 49,13034; 59,13732; 68,97747; 78,65064; 8
FusionCenter_SpeedY; 15,02701; 20,45298; 25,78759; 31,03051; 36,18168; 41,24132; 4
FusionCenter_SpeedZ; -0,00006; -0,00620; -0,01231; -0,01840; -0,02447; -0,03053; -
numSensors; 7;
sensor_1_Name; AIS;
sensor_1_ID; 3;
sensor_1_numTargets; 2; |
sensor_1_target_1_numLines; 31;
sensor_1_target_1_Sensor; AIS;
sensor_1_target_1_ID; 2
sensor_1_target_1_Time; 0,0050000000; 2,0050000000; 4,0100000000; 6,0150000000; 8,
sensor_1_target_1_Latitude; 36,7787260252; 36,7746804203; 36,7702407603; 36,765724
sensor_1_target_1_Longitude; -9,4952229495; -9,4915571145; -9,4881370002; -9,48445
sensor_1_target_1_Speed; 295,5979107787; 292,8911127048; 290,1348223509; 287,33575
sensor_1_target_1_Heading; 146,8485196778; 147,2489638283; 148,6993310325; 151,169
sensor_1_target_2_numLines; 24;
sensor_1_target_2_Sensor; AIS;
sensor_1_target_2_ID; 3
sensor_1_target_2_Time; 0,0050000000; 2,0050000000; 4,0100000000; 6,0150000000; 8,
sensor_1_target_2_Latitude; 36,4752045944; 36,4769655626; 36,4787122928; 36,481348
sensor_1_target_2_Longitude; -9,6942953383; -9,7018092840; -9,7090050968; -9,71580
sensor_1_target_2_Speed; 350,7684194939; 345,8708195435; 340,8440334317; 335,70021
sensor_1_target_2_Heading; -72,3731435607; -72,0674693275; -69,8234879682; -65,725
sensor_2_Name; RADAR;
sensor_2_ID; 2;

```

Fig.5.4: Ejemplo extracto de fichero exportado

puntos de paso, otras figuras los contactos de cada sensor, línea la trayectoria generada, etc.), se ha incorporado información personalizada si se pincha sobre el punto en el mapa, como se ve en la figura. Además, si se selecciona el punto de paso de un blanco en concreto (como en la figura el Waypoint4), este es resaltado en el mapa con un cuadrado azul más claro.

5.2 Generador de contactos

Una vez explicadas las facetas comunes e introducida la herramienta utilizada, se pasa a explicar en detalle la primera aplicación, el generador de contactos.

5.2.1 Sensores implementados

Los sensores implementados en la herramienta son todos los expuestos en el estado del arte (ver 2.2). Para comprender la profundidad del modelado de cada sensor, a continuación, se muestran los parámetros utilizados en cada uno de ellos para permitir su ajuste por parte del usuario:

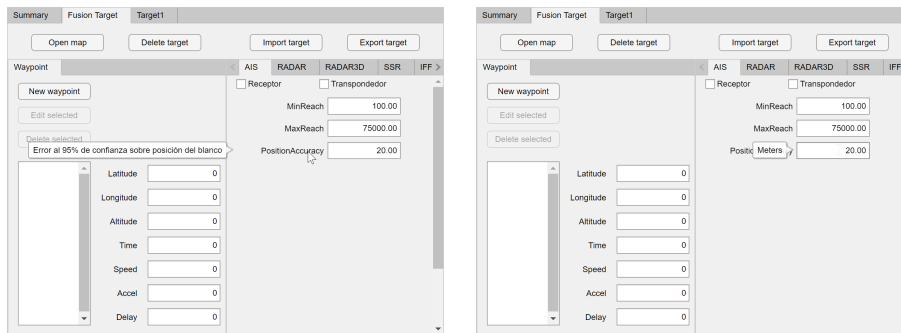


Fig.5.5: Ejemplo de tooltips

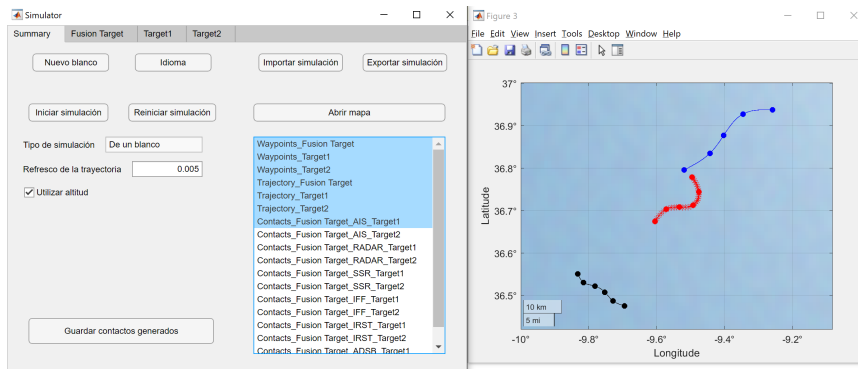


Fig.5.6: Ejemplo del componente del mapa

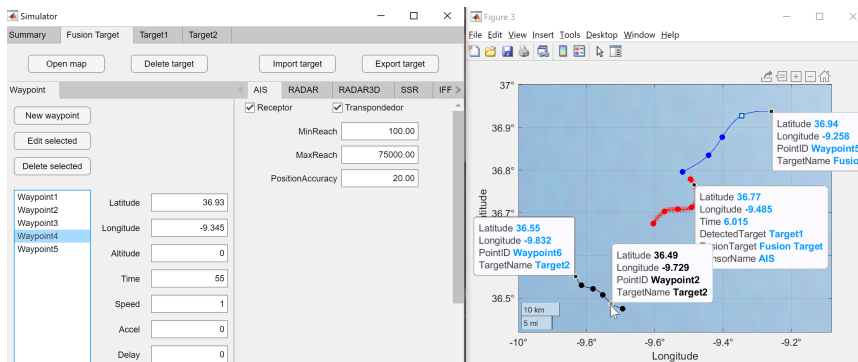


Fig.5.7: Ejemplo de información adicional en el mapa

Sensor	RADAR
MinReach	Mínima distancia para detectar un blanco
MaxReach	Máxima distancia para detectar un blanco
MinElevation	Límite inferior del ángulo de elevación para detectar un blanco

Sensor	RADAR
MaxElevation	Límite superior del ángulo de elevación para detectar un blanco
AzimuthBeamwidth	Ángulo que determina la anchura del rayo emitido por el sensor
PulsesPerSecond	Número de haces lanzados por el sensor cada segundo
AzimuthResolution	Ángulo mínimo de distancia entre dos blancos para que no se superpongan sus señales
RangeResolution	Distancia mínima entre dos blancos para que no se superpongan sus señales
AzimuthAccuracy	Error al 95% de confianza sobre el ángulo de azimut medido
RangeAccuracy	Error al 95% de confianza sobre la distancia del sensor al blanco
RotationPeriod	Tiempo para completar una vuelta sobre sí mismo el sensor ¹
Lim1Azimuth	Se colocan los límites entre donde puede apuntar el sensor ²
Lim2Azimuth	Se colocan los límites entre donde puede apuntar el sensor ²
SectorPeriod	Tiempo para ir de límite a límite sobre el sector del sensor ²
DirectionAzimuth	Ángulo hacia el que apunta el sensor ³

Tab.5.1: Parámetros del sensor RADAR

Sensor	RADAR3D
MinReach	Mínima distancia para detectar un blanco
MaxReach	Máxima distancia para detectar un blanco
MinElevation	Límite inferior del ángulo de elevación para detectar un blanco
MaxElevation	Límite superior del ángulo de elevación para detectar un blanco
AzimuthBeamwidth	Ángulo que determina la anchura del rayo emitido por el sensor
PulsesPerSecond	Número de haces lanzados por el sensor cada segundo
AzimuthResolution	Ángulo mínimo de distancia entre dos blancos para que no se superpongan sus señales
RangeResolution	Distancia mínima entre dos blancos para que no se superpongan sus señales

¹Únicamente si el tipo es "Rotatorio"²Únicamente si el tipo es "Delimitado"³Únicamente si el tipo es "Fijo"

Sensor	RADAR3D
AzimuthAccuracy	Error al 95% de confianza sobre el ángulo de azimut medido
RangeAccuracy	Error al 95% de confianza sobre la distancia del sensor al blanco
RotationPeriod	Tiempo para completar una vuelta sobre sí mismo el sensor ¹
Lim1Azimuth	Se colocan los límites entre donde puede apuntar el sensor ²
Lim2Azimuth	Se colocan los límites entre donde puede apuntar el sensor ²
SectorPeriod	Tiempo para ir de límite a límite sobre el sector del sensor ²
DirectionAzimuth	Ángulo hacia el que apunta el sensor ³
ElevationAccuracy	Error al 95% de confianza sobre el ángulo de elevación medido

Tab.5.2: Parámetros del sensor RADAR3D

Sensor	SSR
MinReach	Mínima distancia para detectar un blanco
MaxReach	Máxima distancia para detectar un blanco
MinElevation	Límite inferior del ángulo de elevación para detectar un blanco
MaxElevation	Límite superior del ángulo de elevación para detectar un blanco
RotationPeriod	Tiempo para completar una vuelta sobre sí mismo el sensor
AzimuthBeamwidth	Ángulo que determina la anchura del rayo emitido por el sensor
PulsesPerSecond	Número de haces lanzados por el sensor cada segundo
AzimuthResolution	Ángulo mínimo de distancia entre dos blancos para que no se superpongan sus señales
RangeResolution	Distancia mínima entre dos blancos para que no se superpongan sus señales
PositionAccuracy	Error al 95% de confianza sobre posición del blanco
RangeAccuracy	Error al 95% de confianza sobre la distancia del sensor al blanco

Tab.5.3: Parámetros del sensor SSR

¹Únicamente si el tipo es "Rotatorio"²Únicamente si el tipo es "Delimitado"³Únicamente si el tipo es "Fijo"

Sensor	IFF
MinReach	Mínima distancia para detectar un blanco
MaxReach	Máxima distancia para detectar un blanco
MinElevation	Límite inferior del ángulo de elevación para detectar un blanco
MaxElevation	Límite superior del ángulo de elevación para detectar un blanco
RotationPeriod	Tiempo para completar una vuelta sobre sí mismo el sensor
AzimuthBeamwidth	Ángulo que determina la anchura del rayo emitido por el sensor
PulsesPerSecond	Número de haces lanzados por el sensor cada segundo
AzimuthResolution	Ángulo mínimo de distancia entre dos blancos para que no se superpongan sus señales
RangeResolution	Distancia mínima entre dos blancos para que no se superpongan sus señales
PositionAccuracy	Error al 95% de confianza sobre posición del blanco
RangeAccuracy	Error al 95% de confianza sobre la distancia del sensor al blanco
EncryptKey	El código utilizado para comunicarse una serie de blancos entre sí

Tab.5.4: Parámetros del sensor IFF

Sensor	IRST
MinReach	Mínima distancia para detectar un blanco
MaxReach	Máxima distancia para detectar un blanco
MinElevation	Límite inferior del ángulo de elevación para detectar un blanco
MaxElevation	Límite superior del ángulo de elevación para detectar un blanco
AzimuthBeamwidth	Ángulo que determina la anchura del rayo emitido por el sensor
PulsesPerSecond	Número de haces lanzados por el sensor cada segundo
AzimuthResolution	Ángulo mínimo de distancia entre dos blancos para que no se superpongan sus señales
RangeResolution	Distancia mínima entre dos blancos para que no se superpongan sus señales
AzimuthAccuracy	Error al 95% de confianza sobre el ángulo de azimut medido
RangeAccuracy	Error al 95% de confianza sobre la distancia del sensor al blanco

Sensor	IRST
RotationPeriod	Tiempo para completar una vuelta sobre sí mismo el sensor ¹
Lim1Azimuth	Se colocan los límites entre donde puede apuntar el sensor ²
Lim2Azimuth	Se colocan los límites entre donde puede apuntar el sensor ²
SectorPeriod	Tiempo para ir de límite a límite sobre el sector del sensor ²
DirectionAzimuth	Ángulo hacia el que apunta el sensor ³
ElevationAccuracy	Error al 95% de confianza sobre el ángulo de elevación medido

Tab.5.5: Parámetros del sensor IRST

Sensor	MLAT
MinReach	Mínima distancia para detectar un blanco
MaxReach	Máxima distancia para detectar un blanco
MinAltitude	Mínima altitud necesaria para detectar un blanco
PositionAccuracy	Error al 95% de confianza sobre posición del blanco
RangeAccuracy	Error al 95% de confianza sobre la distancia del sensor al blanco

Tab.5.6: Parámetros del sensor MLAT

Sensor	ADS-B
MinReach	Mínima distancia para detectar un blanco
MaxReach	Máxima distancia para detectar un blanco
MinAltitude	Mínima altitud necesaria para detectar un blanco
PositionAccuracy	Error al 95% de confianza sobre posición del blanco
RangeAccuracy	Error al 95% de confianza sobre la distancia del sensor al blanco

Tab.5.7: Parámetros del sensor ADS-B

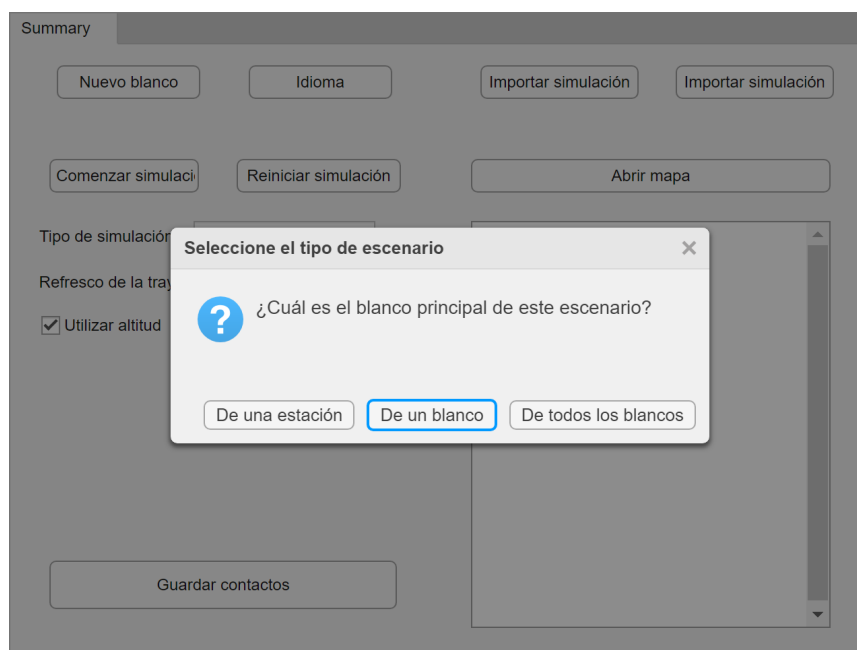


Fig.5.8: Selección del tipo de escenario

5.2.2 Funcionamiento de la aplicación

5.2.2.1 Inicio

La aplicación comienza preguntando al usuario el tipo de escenario que desea generar (ver figura Fig.5.8). Esto es referido a qué blanco es el encargado de generar los contactos. Tiene tres opciones: Una estación / vehículo quieto, un vehículo en movimiento (lo que modifica el algoritmo), o si se desea generar los contactos de todos los blancos entre sí. Una vez seleccionados, se generan dos blancos, el número mínimo para poder obtener contactos desde uno sobre el otro. Todo este proceso es idéntico a si se pulsara el botón de "nuevo blanco".

Una vez termina, se ve que el tipo de simulación ha cambiado a la selección que se ha realizado. Si se quisiera cambiar este modo, es necesario reiniciar la simulación para comenzar de nuevo, al afectar a todo el escenario el valor original.

5.2.2.2 Nuevo blanco

Si se pulsa el botón de nuevo blanco, se genera un blanco adicional a los que existieran actualmente. Se pueden generar tantos como se deseen. Se

¹Únicamente si el tipo es "Rotatorio"

²Únicamente si el tipo es "Delimitado"

³Únicamente si el tipo es "Fijo"

muestra una barra de progreso y se genera la estructura para almacenar los puntos de paso que introduzca el usuario, y los sensores seleccionables, todos ellos desactivados, con los parámetros por defecto. Una vez termina, se quita la barra de progreso y se cuenta con una nueva pestaña con el nuevo blanco.

5.2.2.3 Nuevo punto de paso

Una vez en un blanco, si el usuario presiona el botón de nuevo blanco, se abre el mapa para registrar su latitud y longitud, como se observa en la figura. Una vez terminado, se añade a la lista de puntos de paso. Si se selecciona dicho punto de paso, el usuario puede editar dicho punto en el mapa, pero también puede editarlo modificando los valores directamente en las entradas de su derecha. También se puede eliminar el punto de paso, colocando así los nombres tanto en la lista como en el mapa.

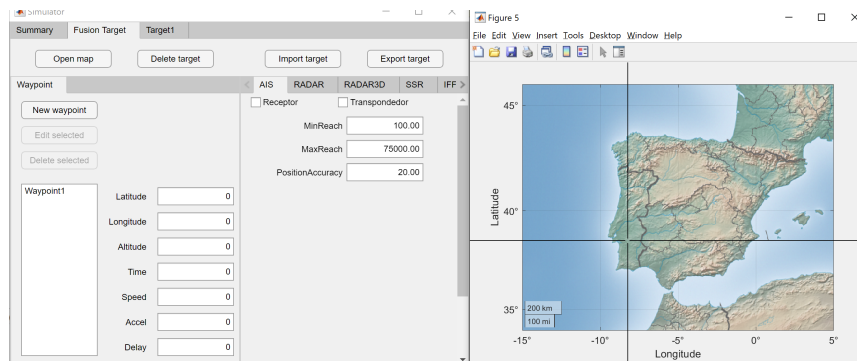


Fig.5.9: Nuevo punto de paso

5.2.2.4 Selección de sensores

En el lado derecho, contamos con otro conjunto de pestañas en el cual se encuentran todos los sensores por defecto desactivados, para que el usuario, si lo desea, los utilice en dicho blanco o no. Además, se ha dividido su uso en función del tipo de sensores en varios aspectos (ver figura Fig.5.10). Un sensor independiente, como puede ser el RADAR, cuenta con tres tipos de funcionamiento, aparte de si se utiliza o no. Estos modos definen el rango de cobertura del mismo. El sensor puede rotar continuamente abarcando los 360 grados, abarcando una cobertura fija o haciendo pequeños patrones de giro, delimitados dentro de un rango. Cada uno de los modos hace variar algunos de los parámetros del sensor, modificando la ejecución del algoritmo y por ello los contactos que detecta.

Por otro lado, tenemos los sensores dependientes, en los que se incluye el SSR, donde su activación cuenta con dos parámetros, si desea recibir contactos, y si desea ser detectado por otros blancos. De esta manera, en

estos sensores es imprescindible que los blancos tengan esta opción activa para ser detectados. Además, se permite detectar pero no ser detectado, una situación habitual en el ámbito militar.

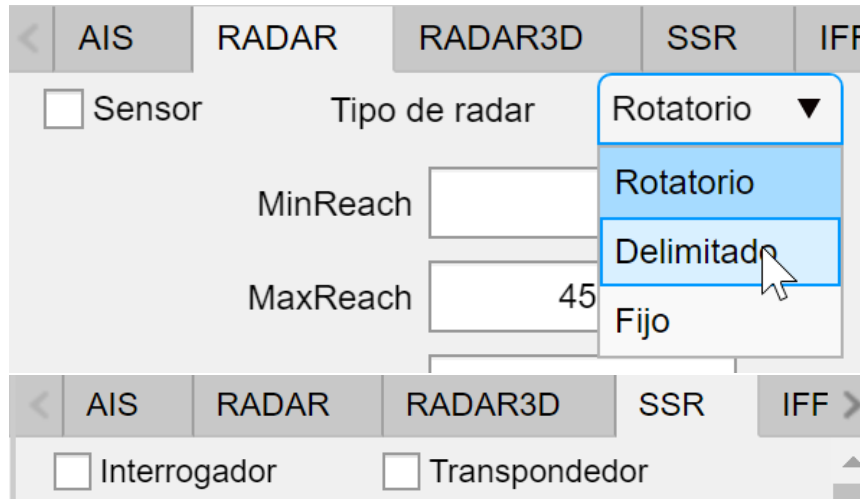


Fig.5.10: Uso de distintos sensores

5.2.2.5 Importar / exportar

Las funciones de importar / exportar tanto blancos como toda la generación tienen un comportamiento similar. Una vez presionado el botón, se abre una ventana del sistema operativo para seleccionar el fichero que leer / guardar. Por supuesto, si existe un error en los parámetros salta un error como se explicó anteriormente. Se ha implementado una ayuda a la hora de exportar, colocando el nombre del fichero unívoco de manera automática (ver figura Fig.5.11).

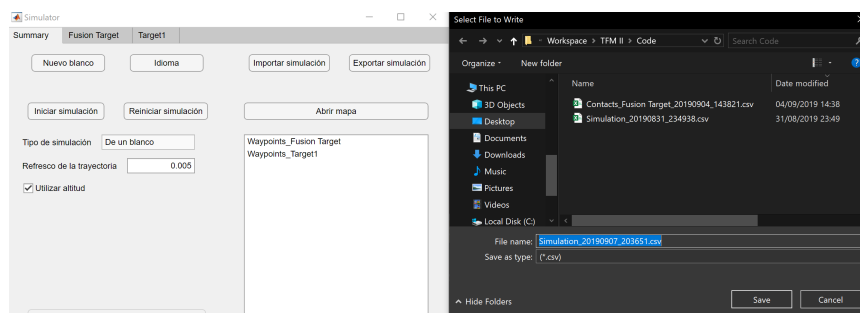


Fig.5.11: Nuevo punto de paso

Una vez comienza la importación se genera un mensaje de progreso, puesto que debe eliminar toda la información existente en dicha estructura

(también del mapa) y cargar la nueva información. En cambio, si se presiona exportar toda la información deseada se escribe en el fichero de manera inmediata.

5.2.3 Algoritmia de la aplicación

Una vez expuesta toda la información de la interfaz de la aplicación, se pasa a explicar la algoritmia utilizada para generar los contactos. Como se muestra en la figura Fig.5.12, se puede dividir en 7 principales pasos:

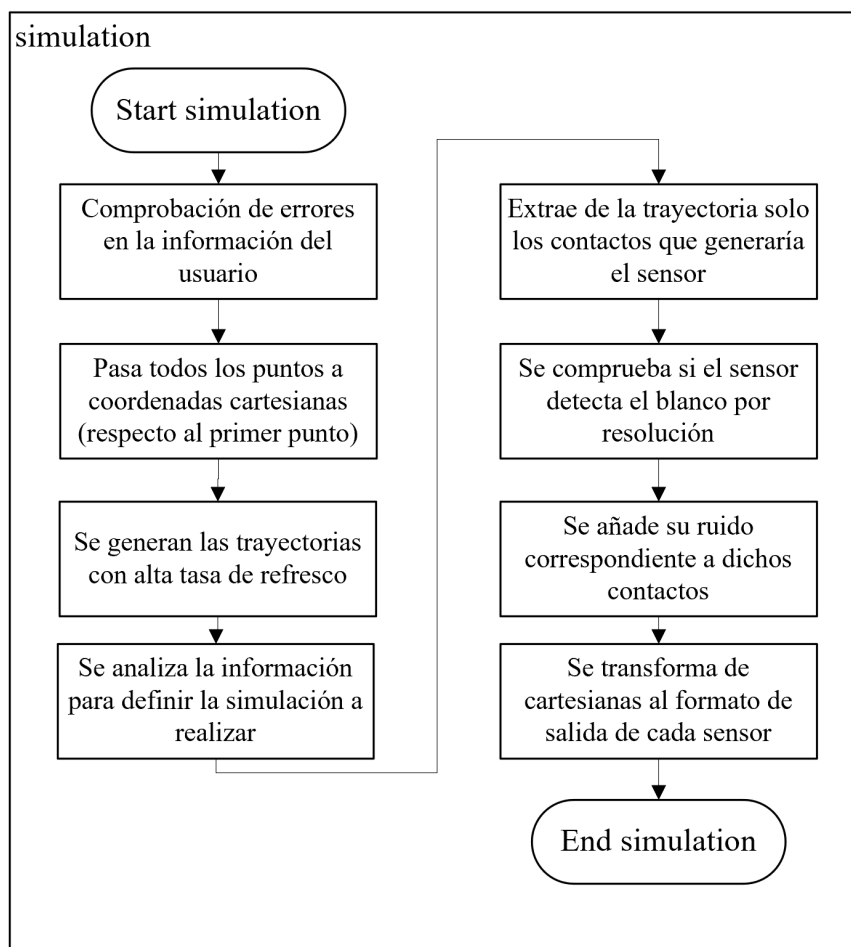


Fig.5.12: Algoritmo generador de contactos

- se comienza comprobando toda la información que ha insertado el usuario, tanto de cada blanco como de los parámetros del algoritmo, como si existe suficiente información para generar contactos.
- Una vez toda la información se ha validado, se transforma toda la información a coordenadas cartesianas. Se utiliza la primera posición

del blanco que genera contactos (en caso de ser la estación fija, dicho punto) y se transforman de WGS-84 (latitud y longitud) a cartesianas (X, Y, Z). Se almacenan dichos valores.

- A continuación se llega al generador de trayectorias. Teniendo en cuenta cada punto de paso del blanco, tanto sus posiciones como su velocidad deseada, como su tiempo de paso, etc. Con toda esta información, el algoritmo desarrollado es capaz de generar, de tres en tres puntos de paso, la trayectoria seguida desarrollando una curva b-spline, y posteriormente ajustar la velocidad para cuadrar tanto el tiempo como la velocidad de paso. Este proceso se realiza con todos los blancos, generando sus trayectorias, que posteriormente son dibujadas e insertadas en el mapa.
- El cuarto paso es el encargado ordenar la simulación a realizar, con el objetivo de pasar la información necesaria al siguiente paso. Para cada sensor del blanco que genera contactos, se recogen únicamente las trayectorias que dicho sensor debe utilizar. Además, si es un sensor dependiente, se le pasan los sensores de los receptores, que son los que emiten la información.
- Con la trayectoria del blanco actual, las características de cada sensor, y los blancos que puede detectar, se pasa a un método específico para cada tipo de sensor que reporta solo los contactos que obtendría de manera real el sensor. Por ejemplo, un RADAR recibe solo cuando está apuntando hacia el blanco a detectar (esté rotando o no), pero un ADSB recibe cada medio segundo la información.
- Una vez se tienen los contactos que se detectan, se comprueba si hay dos blancos en la misma dirección desde el blanco que genera contactos. Si lo están (con un cierto margen que define el sensor) genera un problema de resolución, haciendo que el sensor no sea capaz de interpretar la respuesta. En estos casos, se ha decidido eliminar ambos contactos.
- Una vez se tienen todos los contactos, tras eliminar con conflictos de resolución, se introduce el ruido a los contactos (hasta ahora exactos respecto a la trayectoria). Se cuenta con distintos tipos de ruido a generar. El primero de ellos es la generación para sensores dependientes. En estos casos, los parámetros del sensor del blanco detectado modelan el valor de ruido al 95% de confianza, directamente en metros, imitando el funcionamiento del GPS. El otro tipo de ruido es el de los sensores independientes, se cuenta con un ruido de dirección y de distancia. Al igual que antes, con el 95% de confianza se calcula la dirección real con ruido, y la distancia en línea recta con ruido. Con estos valores, se calcula de nuevo la posición cartesiana, con el ruido incorporado.

- Finalmente, una vez se tiene en cada contacto la posición con un cierto ruido aleatorio, se generan las variables que se necesitan para cada sensor.

5.3 Sistema de fusión

Por otro lado, se desarrolla la segunda herramienta, la destinada a probar un sistema de fusión de datos con las características deseadas y los contactos introducidos.

5.3.1 Algoritmos implementados

Como ya se ha explicado, cada componente de un sistema de fusión multisensor puede utilizar un algoritmo para su funcionamiento. Para permitir las pruebas sobre el sistema de fusión desarrollado, se han implementado los siguientes algoritmos para cada componente.

- Componente de Estimación de estado:
 - Filtro de Kalman Extendido (EKF)
 - Filtro de Kalman “Unscented” (UKF)
 - Filtro de Kalman Ensemble (EnKF)
 - Filtro de partículas (PF)
 - Filtro de Múltiples modos interoperantes (IMM)
- Componente de Asociación de trayectorias:
 - Vecino más próximo (NN)
 - Asociación Probabilística de Datos (PDA)
 - Seguimiento con Hipótesis Múltiples (MHT)
 - Algoritmo de Munkres

Indicar de nuevo que estos algoritmos ya estaban desarrollados, con otra estructura. Únicamente fue necesario adaptarlos a la interfaz y la nueva estructura de datos.

5.3.2 Métricas de evaluación implementadas

Además, para agregar funcionalidad al análisis de los resultados de cada test, además de permitir comparar los resultados de estimación y asociación en el mapa (de manera similar a la mostrada en la otra herramienta), se calculan las siguientes métricas, el usuario puede utilizar para complementar su análisis:

- Tiempo de cómputo. Medido por cada iteración del algoritmo.
- Innovación cuadrática normalizada (NIS)
- Error cuadrático de estimación normalizado (NEES)
- Raíz del error cuadrático medio (RMSE)

5.3.3 Funcionamiento de la aplicación

La aplicación cuenta con un funcionamiento de la interfaz similar al de la primera aplicación.

5.3.3.1 Vista general

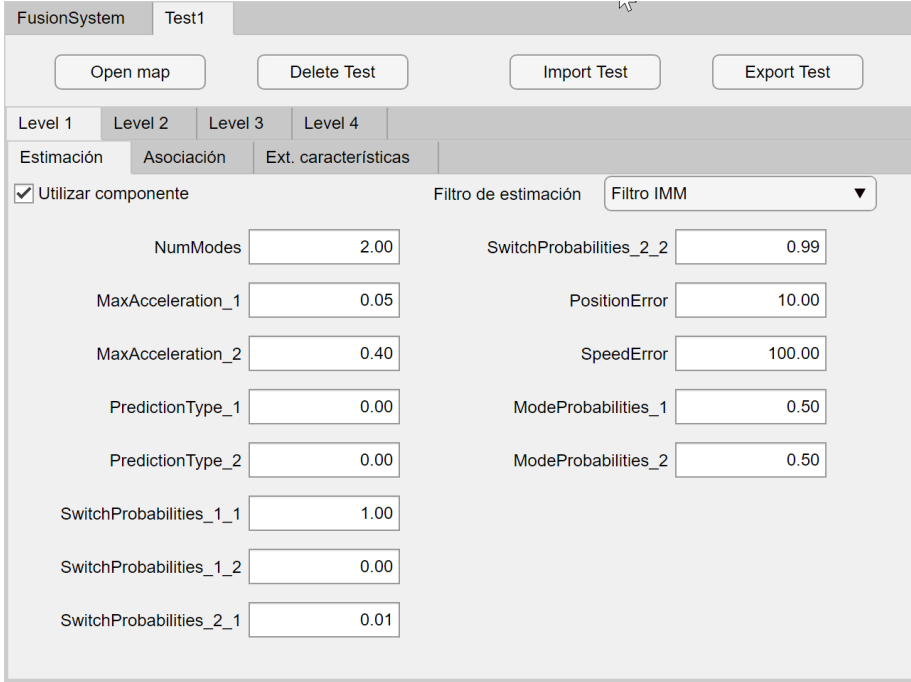


Fig.5.13: Vista de prueba. Estimación

Inicialmente se muestra la pantalla principal, el sumario. En el se pueden hacer las acciones principales, similares a la del generador de contactos.

5.3.3.1.1 Nueva prueba de fusión El usuario puede crear todas las pruebas que desee realizar en esta ejecución del sistema de fusión.

5.3.3.1.2 Cargar escenario Es necesario contar con una función para importar todo el escenario desarrollado en la herramienta anterior, tantos los contactos generados por el centro de fusión por cada sensor, como la trayectoria definida por el mismo. Esta función, como toda la información que se exporta, se lee mediante un archivo CSV con una interfaz común.

5.3.3.1.3 Importar / exportar conjunto de pruebas Al igual que en el sistema de generación de contactos, esta aplicación permite exportar todas las características definidas en la aplicación para poder reutilizarla en el futuro. Como siempre, este proceso genera / lee toda la información en un fichero CSV con una interfaz común.

5.3.3.1.4 Ejecución de las pruebas Una vez definidas las pruebas, se cuenta con el botón de ejecución de las pruebas definidas. Este pone en marcha la algoritmia definida en Fig.5.18. Este proceso, al ser lo suficientemente costoso en cómputo, utiliza el mensaje de progreso explicado anteriormente.

5.3.3.1.5 Guardar resultados Otra funcionalidad similar es la que permite exportar todos los resultados de todas las pruebas generadas en un fichero CSV. De esta manera, se pueden observar en otra herramienta, como Microsoft Excel. Además, en esta aplicación se pueden utilizar en la siguiente funcionalidad.

5.3.3.1.6 Importar resultados Una funcionalidad nueva de esta aplicación es la capacidad de importar los resultados generados, exportados en CSV, de nuevo a la aplicación. Esto permite observar de nuevo unos resultados concretos en la aplicación y realizar la comparativa, sin tener que ejecutar todas las pruebas del sistema de fusión de datos multisensor, las cuales pueden tardar bastante tiempo.

5.3.3.1.7 Comparativa de resultados La zona utilizada para mostrar las trayectorias y contactos en el mapa, ahora se utiliza para seleccionar qué resultados se quieren evaluar.

Como se observa en la figura Fig.5.14, ahora se cuenta más entradas que antes, al mostrar tanto la estimación de cada serie de contactos y blanco (LocalTrack, traza local), como su asociación (GlobalTrack, traza global). Esta serie de entradas permite comparar directamente en el mapa, o en una figura utilizando sus coordenadas cartesianas, donde se muestra más información (como se observa en la figura Fig.5.15, con un ejemplo de asociación en la herramienta).

Además, se introducen en esta lista de evaluación las métricas generadas, las cuales tienen un comportamiento similar, permitiendo introducir tantas

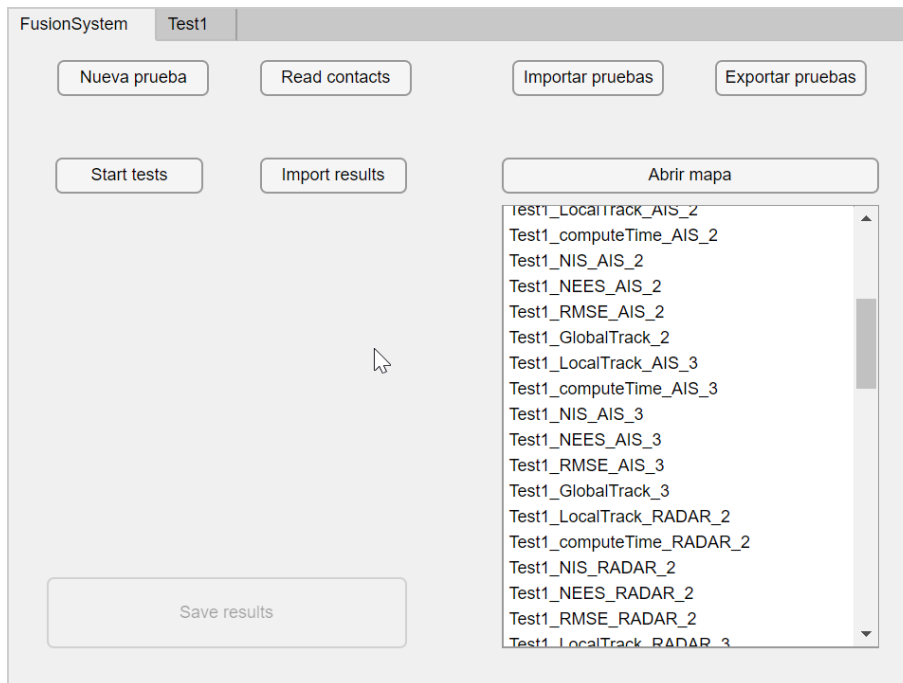


Fig.5.14: Listado de resultados para evaluar

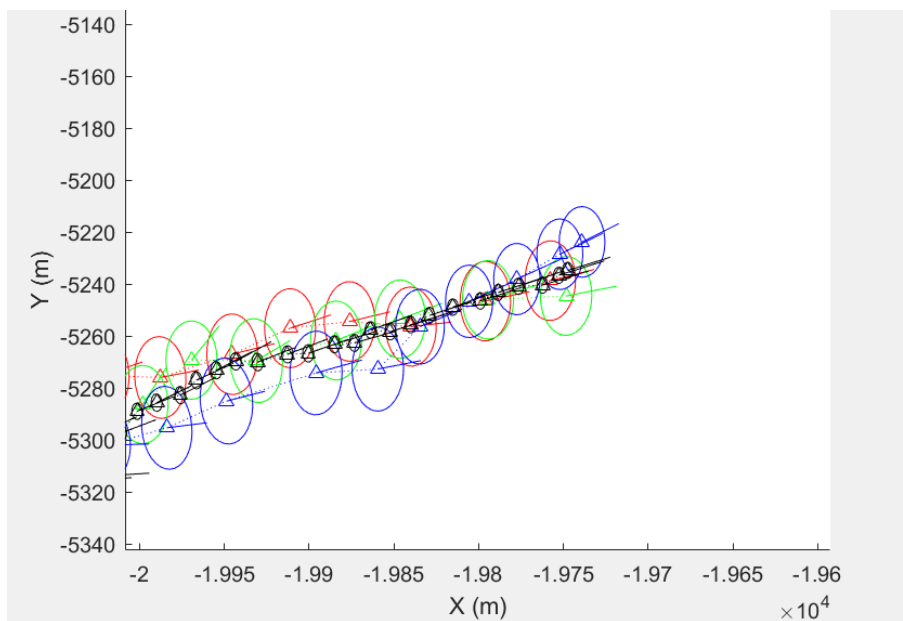


Fig.5.15: Ejemplo de asociación en la herramienta. En rojo, verde y azul, las trazas locales de varios sensores del blanco número 3, y en negro su traza global

entradas como se deseen, en una figura nueva (puesto que no son coordenadas, son valores que varían respecto al tiempo). En la figura Fig.5.16 se muestra un ejemplo de un gráfico comparativo de una métrica.

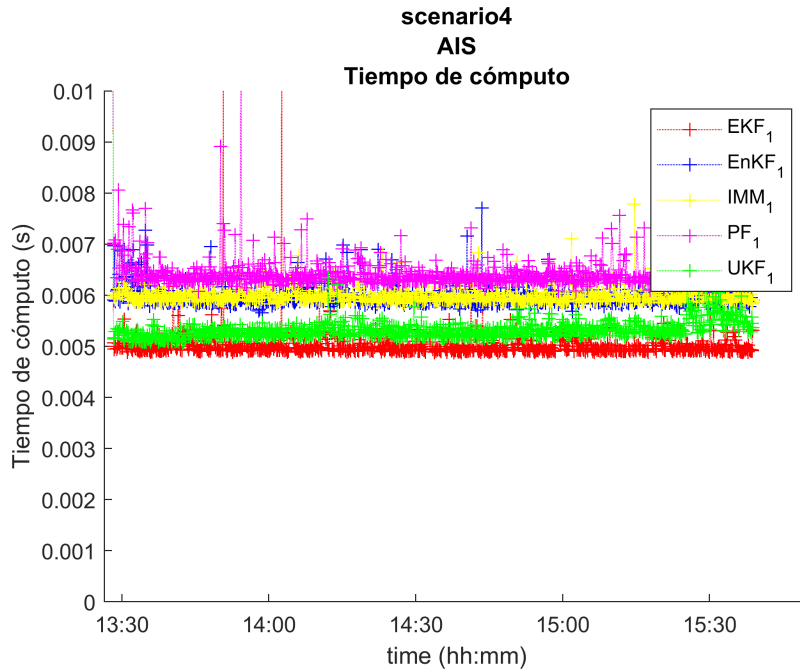


Fig.5.16: Ejemplo de métrica en la herramienta. Comparativa de tiempos de cómputo de la estimación del mismo blanco, con filtros de estimación distintos

5.3.3.2 Vista de prueba

Una vez el usuario genera una nueva prueba, se genera una nueva pestaña con la interfaz que se muestra en la figura Fig.5.17.

Como se puede observar, se cuentan con los botones habituales en la parte superior, para eliminar la prueba actual, e importar o exportar esta prueba actual si se desea. Por debajo se encuentran los componentes de un sistema de fusión de datos multisensor implementados en la herramienta, divididos para una mejor comprensión por la clasificación de niveles de JDL. Dentro de un cada componente, se cuenta con un botón para activar o desactivar el componente, si es posible para dicho componente, puesto que hay algunos (como en este caso el estimador) que deben aplicarse siempre.

En la parte derecha se incorpora la selección de los distintos algoritmos de este componente, mostrados mediante un desplegable, el cual cambia todas las variables a configurar mostradas justo debajo, en función de la selección optada.

Level 1	Level 2	Level 3	Level 4
Estimación Asociación Ext. características			
<input checked="" type="checkbox"/> Utilizar componente			
Filtro de estimación		Filtro IMM	
NumModes	2.00	SwitchProbabilities_2_2	0.99
MaxAcceleration_1	0.05	PositionError	10.00
MaxAcceleration_2	0.40	SpeedError	100.00
PredictionType_1	0.00	ModeProbabilities_1	0.50
PredictionType_2	0.00	ModeProbabilities_2	0.50
SwitchProbabilities_1_1	1.00		
SwitchProbabilities_1_2	0.00		
SwitchProbabilities_2_1	0.01		

Fig.5.17: Vista de prueba. Estimación

5.3.4 Algoritmia de la aplicación

La parte más importante de esta aplicación es la algoritmia que tiene detrás para simular el funcionamiento del ciclo de fusión de datos multisensor. En la figura Fig.5.18 se muestra el funcionamiento del ciclo de fusión implementado, realizando tanto la estimación como la asociación.

- Se comienza comprobando que todas las entradas del usuario son aceptadas y se permite realizar las pruebas sin fallos.
- En ese instante, se recopila toda la información de las pruebas, facilitando posteriormente el acceso a las variables y los componentes activos en el ciclo de fusión.
- A partir de todos los contactos, almacenados hasta ahora por separado, se intercalan ordenados por tiempo, simulando la llegada de manera real.
- Se comienza con cada prueba, de manera ordenada.
- Para la prueba i , se comienza con el primer contacto. Se transforma este a coordenadas cartesianas respecto al centro de fusión (ya sea estático o en movimiento). Los contactos que no se pueden transformar, como el IRST, se salta hasta la asociación.

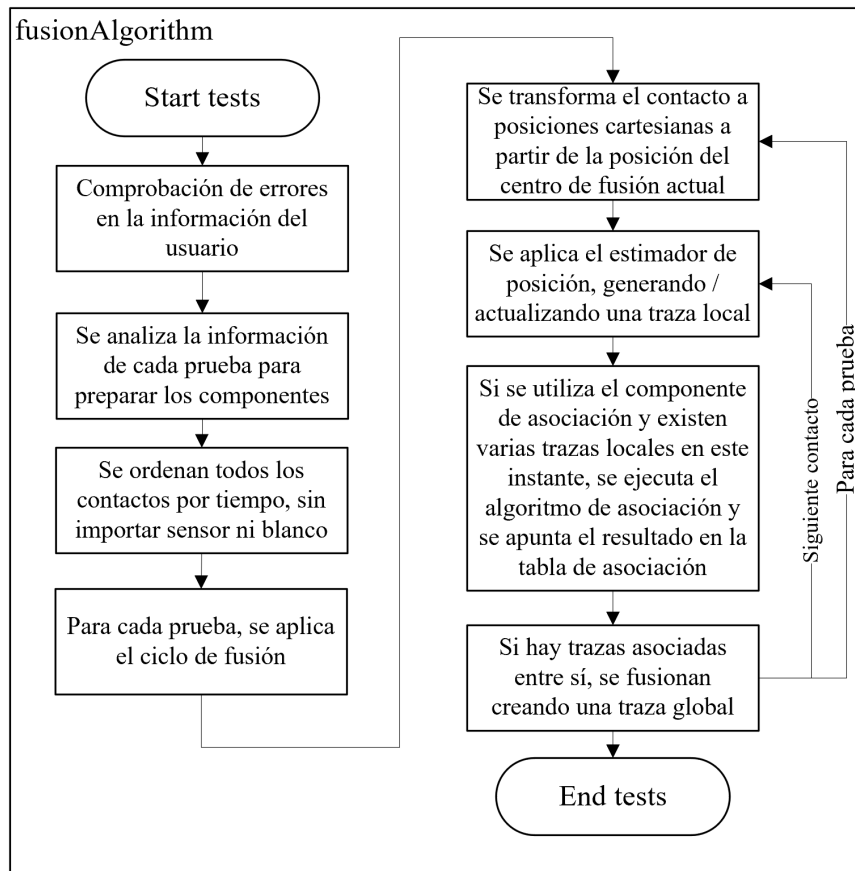


Fig.5.18: Algoritmo fusión de datos multisensor

```

global_track_id;local_track_id;sensor_id;
6001;3001;3;
6002;3002;3;
6002;9002;9;
6003;10001;10;
6004;3003;3;
6004;10003;10;
6004;9003;9;

```

Fig.5.19: Ejemplo de tabla de asociación

- Con el contacto en una interfaz comparable, se utiliza su estimador de estado, de manera que se pueda conocer con precisión y confianza su posición, velocidad y aceleración. Se utiliza el algoritmo seleccionado por el usuario, con los valores que el ha introducido. Se construye así una traza local para un blanco concreto.
- Con la estimación realizada se comprueba si existe alguna traza con

la que asociarla, si hasta ahora no se había asociado. Para ello se cuenta con la tabla de asociaciones, que identifica las relaciones entre las trazas locales (seguimiento de un sensor a un blanco concreto) y las trazas globales (fusión de trazas locales que siguen al mismo blanco). En la figura Fig.5.19 se muestra un ejemplo de la tabla de asociación.

- Si aún no ha sido asociado, se ejecuta el algoritmo de asociación. Este devuelve la traza global con la que debe asociarse esta traza local. Si no existía ninguna traza global hasta ahora, se creará una nueva.
- Una vez se cuenta con trazas globales, cada vez que se actualice su una de sus trazas locales asociadas, se actualiza la traza global. Para ello, se utiliza una unión mediante el método de los mínimos cuadrados, generando la unión más adecuada entre todas las trazas locales.

Como se comenta, se realiza el proceso sobre todos los contactos y todas las pruebas diseñadas por el usuario. De esta manera, una vez termina se cuenta con una gran cantidad de entradas para seleccionar en el mapa, de manera que el usuario pueda comparar y comprobar la exactitud de la trayectoria fusionada en cada prueba.

Capítulo 6

Pruebas del sistema

Tras explicar la herramienta desarrollada, toca el turno de exponer la validación realizada para comprobar si el software cumple las expectativas definidas al comienzo del trabajo. Para ello, en la sección 6.1 se introduce el método de validación utilizado en este proyecto. Seguido, se muestra la validación de cada uno de los escenarios en la sección 6.2.

6.1 Introducción

Una de las partes más importantes en un proyecto software es realizar un testeo sobre el mismo, para garantizar que cumple los requisitos que buscaba satisfacer. De esta manera, con coherencia con el resto del trabajo, se prueba la validez del software desarrollado observando si los requisitos desarrollados, plasmados en las aplicaciones, permiten satisfacer los escenarios definidos.

Cabe destacar que las pruebas sobre el funcionamiento de los algoritmos de los componentes de fusión no son necesarias, puesto que los mismos están desarrollados en un trabajo de fin de máster previo [36], con sus pruebas accesibles. Por ello, únicamente es necesario, al haberse probado el funcionamiento de toda la usabilidad en el capítulo 5, del cumplimiento de los escenarios definidos.

6.2 Comprobación escenarios

6.2.1 Comprobación del escenario 1

Observando el escenario número 1 (ver 4.3.0.1), Javier con la herramienta de generación de contactos tendría todo lo que necesita para realizar el trabajo deseado. Además, la herramienta cuenta con una alta usabilidad que le facilitaría la comprensión de los distintos términos y el acceso al código para indagar más y realizar las modificaciones que considere oportunas.

6.2.2 Comprobación del escenario 2

Luisa por su parte, conseguiría solventar los dos problemas de su escenario 4.3.0.2. Para empezar, teniendo una herramienta podría ofrecer trabajos de fin de grado más interesantes y sencillos a sus alumnos. De esta manera, aumentaría la cantidad de investigación que realiza y cumpliría sus objetivos.

Por otro lado, conseguiría una herramienta con la que desarrollar sus pruebas de sistemas de fusión multisensor. La parte más laboriosa del ciclo de fusión y la generación de contactos ya está desarrollada, por lo que para hacer pruebas más complejas únicamente debería codificar componentes y añadirlos en la pestaña correspondiente.

6.2.3 Comprobación del escenario 3

El primer escenario de Luke (ver 4.3.0.3), por su parte, ofrece una interfaz y estructuras de datos ya implementadas que podrían utilizar todo su equipo de desarrollo para tener una herramienta común. Lógicamente, tendría un alto trabajo migrar todo su trabajo actual a estas herramientas, pero con el tiempo conseguirían una herramienta muy completa y eficaz para esas necesidades. De esta manera, además, se eliminaría la imprescindibilidad de cada empleado, al ser un código muy explicable, comentado y con una documentación auxiliar clara en esta memoria.

6.2.4 Comprobación del escenario 4

El segundo escenario de Luke (ver 4.3.0.4) también se satisface. Gracias a la herramienta, se puede exportar las pruebas y replicarlas en cualquier momento. Además, la capacidad de generación de comparación de pruebas y gráficos le otorga la capacidad de probar cualquier variación del sistema de fusión y mostrarlo con claridad. No solo la comparación de la posición del vehículo en el mapa, sino la generación de gráficos con métricas de análisis del comportamiento de cada algoritmo. Al exportar los resultados con un sistema CSV, facilita la compatibilidad con otras herramientas para generar otro tipo de gráficos más complejos.

Capítulo 7

Conclusiones y líneas futuras

En este último capítulo de la memoria se concluye el proyecto. En la sección 7.1 se introducen las conclusiones extraídas de este proyecto. A continuación, en la sección 7.2 se comprueba la exactitud de la planificación realizada. Finalmente, se muestran los próximos caminos que ha abierto este trabajo en la sección 7.3.

7.1 Conclusiones

Este proyecto tiene como principal objetivo acercar y facilitar el acceso de los sistemas de fusión de datos multisensor tanto a nuevos investigadores como a personas expertas. Esta tarea se ha cumplido con éxito, generando una aplicación con una usabilidad más que correcta con una herramienta poco habitual para ello, MATLAB. Además, ha servido para sentar las bases de un proyecto que puede terminar siendo muy completo si se continúa con un ritmo de trabajo similar al de esta planificación.

7.1.1 Cumplimiento de los objetivos

Para comprobar si el proyecto de ingeniería se ha cumplido de manera adecuada, el proceso a seguir es repasar los objetivos marcados al comienzo de la tesis. Estos eran:

- De manera teórica, es necesario comprender el funcionamiento teórico de los sistemas de fusión multisensor, especialmente los destinados a entornos 3D.
 - Como se ha realizado en la sección 2.1, se han analizado todos los componentes de un sistema de fusión de datos multisensor de manera teórica, haciendo énfasis en los algoritmos de los componentes implementados en este trabajo.
- Junto a este, se necesita estudiar los algoritmos que dan soporte a estos sistemas de fusión para poder comprender su funcionamiento,

sus variables y ser capaz de implementarlo y analizarlo.

- Se han implementado todos los algoritmos indicados, siendo completamente intercambiables entre ellos, y con variables accesibles por el usuario para ajustarlo a sus necesidades y pruebas. La algoritmia de cada uno está disponible en las referencias citadas en la sección 2.1. No se considera necesaria en este trabajo, al haberse realizado un desarrollo específico sobre ellas en el trabajo de fin de máster [36], ni ser el objetivo principal conocer dicha algoritmia en este proyecto.
- A su vez, es importante conocer el funcionamiento detrás de cada tipo de sensor que sirve de entrada para los sistemas de fusión.
 - En la sección 2.2 se analizan de manera detallada la gran mayoría de los sensores aplicados a los sistemas de fusión de datos multi-sensor en el ámbito marítimo. Además, se categorizan en función de diversos parámetros y se comparan de manera minuciosa.
- Por otro lado, en el aspecto de la usabilidad de la herramienta a desarrollar, hay que analizar cuales son los problemas que tienen los usuarios para así poder comprender sus necesidades y desarrollar un proyecto de ingeniería que los resuelva.
 - El capítulo 4 cumple específicamente este objetivo. Se establece el alcance inicial del proyecto, con el se diseñan los artefactos persona que representan grupos de usuarios de la aplicación. Con ellos, se establecen escenarios en los que la aplicación puede satisfacer a los mismos. Tras analizar toda esta información, se definieron los requisitos del proyecto.
- Una vez conocidas las necesidades, se debe realizar este proceso de ingeniería, estableciendo una metodología de trabajo, desarrollar una planificación adecuada y un presupuesto válido.
 - Este objetivo se sacia en el capítulo 3. Se define una metodología adaptada al proyecto, se genera una planificación que, como se puede observar en la sección 7.2, ha sido bastante ajustada y un presupuesto adecuado visto el acierto en la planificación y el resultado del software.
- Este proyecto de ingeniería debe estar validado antes de comenzar con el desarrollo. Para ello se debe utilizar una técnica de validación para contrastar las necesidades de los usuarios y la utilidad de este proyecto.
 - Una vez establecidos los requisitos y por ello la propuesta de proyecto, se valida el proyecto con dos entrevistas con usuarios

potenciales de la aplicación. Ambos consideraron la aplicación útil para el trabajo que desarrollan actualmente, por lo que se confirma la utilidad de la misma.

- Una vez validado el prototipo, se procederá a desarrollar la herramienta. Para ello, se debe generar tanto la interfaz de usuario final, como la estructura de datos de cada aplicación, como la algoritmia que tienen por detrás.
 - El desarrollo de la herramienta se ha cumplido con creces. La interfaz gráfica es más que solvente para estar desarrollada en MATLAB. Se ha dejado preparado el terreno para ampliar con nuevos componentes y algoritmos, y se han implementado y adaptado los algoritmos necesarios para dotar a la herramienta actual de una funcionalidad que satisface los requisitos marcados.
- Todo este desarrollo debe ir acompañado de una explicación adecuada y solvente, tanto en esta memoria como en el código para ser reutilizable y ampliable en el futuro.
 - La documentación proporcionada en el capítulo 5, así como la limpieza y comentarios del código, junto a los aspectos que se acaban de comentar, hacen que la herramienta sea fácilmente ampliable en el futuro.

Todos los objetivos se han cumplido, por lo que se clausura el proyecto como exitoso.

7.1.2 Conclusiones personales

A nivel personal este es el proyecto más ambicioso que he desarrollado. Además de satisfacer una necesidad existente, esta herramienta, a diferencia de otros trabajos que he realizado, sí será útil en un futuro cercano, al menos para mí en el doctorado de Ciencia y Tecnología Informática. Integrar todos los componentes, que hasta ahora se tenían de manera dispersa en distintos proyectos ha sido sencillo, pero no así otorgarles a todos ellos una interacción sencilla pero completa, además de permitir realizar las pruebas de manera rápida. La herramienta utilizada no ofrecía ninguna facilidad para el desarrollo de una interfaz gráfica amigable, pero aún así creo que este aspecto ha quedado más que correcto.

Este proyecto me ha servido para darme cuenta de la importancia de algunos conceptos aprendidos a lo largo del máster, y como estos han hecho que este desarrollo se haga de manera más lógica y sistemática. La importancia de analizar la regulación del software que realizas, la importancia de gestionar un equipo de trabajo compuesto por personas, la necesidad

de proveer un código mantenible en el tiempo, y un largo etcétera han sido conceptos que sin este máster no habrían tenido este impacto.

7.2 Comparación tiempos de trabajo

Como se comentó anteriormente, se ha utilizado la aplicación TimeCamp para medir el tiempo de trabajo de cada rol. De esta manera, se puede calcular con precisión si el tiempo estimado de trabajo, y la planificación, se ha cumplido o ha sido errónea, provocando un problema con el presupuesto solicitado.

TimeCamp permite organizar las tareas realizadas a lo largo del día en bloques, de manera que obtener únicamente el tiempo productivo, obviando la cuenta de tiempos en aplicaciones o webs que no han aportado al proyecto. De esta manera, una entrada diaria puede resumirse en una serie de tareas realizadas. Se muestra un ejemplo en la figura Fig.7.1.

Name	Time	Time (%)	Category	Productivity	Type	
matlabwindow	0,04	40	No category	0	app	2019-04-16 comenzando sensores
MATLAB	0,02	17	Developer Tools	1	app	
mendeleydesktop	0,01	14	Productivity	1	app	
google.com	0,01	8	Reference & Search	0	www	
code	0,00	3	Developer Tools	1	app	
icao.int	0,00	3	No category	0	www	
es.mathworks.com	0,00	3	Office	1	www	
revistas.udistrital.edu.co	0,00	2	No category	0	www	
eurocontrol.int	0,00	2	Reference & Search	0	www	
multilateration.com	0,00	1	No category	0	www	
radartutorial.eu	0,00	1	No category	0	www	
skybrary.aero	0,00	1	No category	0	www	
revistahub.com	0,00	1	No category	0	www	
github.com	0,00	1	Developer Tools	1	www	
indracompany.com	0,00	1	No category	0	www	
code.nasa.gov	0,00	1	No category	0	www	
Name	Time	Time (%)	Category	Productivity	Type	Nombre de la tarea
code	0,05	31	Developer Tools	1	app	2019-04-18 leyendo sobre sensores
docs.google.com	0,02	15	Office	1	www	
matlabwindow	0,02	14	No category	0	app	
mendeleydesktop	0,01	7	Productivity	1	app	
Google Chrome	0,01	7	Reference & Search	0	app	
en.wikipedia.org	0,01	5	Reference & Search	0	www	
multilateration.com	0,01	3	No category	0	www	
icao.int	0,00	2	No category	0	www	
Acrobat Reader	0,00	2	Productivity	1	app	
es.wikipedia.org	0,00	2	Education	0	www	
art.torvergata.it	0,00	2	No category	0	www	
radartutorial.eu	0,00	2	No category	0	www	
oa.upm.es	0,00	1	No category	0	www	
ads-b.com	0,00	1	No category	0	www	
gssc.esa.int	0,00	1	No category	0	www	
thebalancecareers.com	0,00	1	Office	1	www	
MS PowerPoint	0,00	1	Productivity	1	app	
web.mit.edu	0,00	1	Education	0	www	
indracompany.com	0,00	1	No category	0	www	
skybrary.aero	0,00	0	No category	0	www	
upcommons.upc.edu	0,00	0	No category	0	www	

Fig.7.1: Ejemplo entrada TimeCamp

Contabilizando el tiempo dedicado a cada tarea, según el día de la misma, se puede observar si se ha cumplido con la planificación inicial del proyecto. De media, se han dedicado al proyecto 3 horas y 25 minutos, en un total de

132 días, desde el 1 de abril hasta el 6 de septiembre. El número de días varía respecto a la planificación, al haberse utilizado algunos fin de semana para completar el proyecto, no contados inicialmente en la herramienta Microsoft Project. Esto indica que el número de días estimado era algo incorrecto, pero esto es debido a que no se ha alcanzado la media esperada de trabajo inicial, fijada en 4 horas al día. A pesar de ello, se considera una variación mínima.

Si nos centramos, por ejemplo, en el tiempo de desarrollo de las herramientas, el tiempo empleado en las herramientas "matlabwindow", "MATLAB" y "es.mathworks.com" son 35 días de trabajo con el tiempo medio por día, distribuidos en 52 entradas de TimeCamp. Esto indica que la aproximación era medianamente correcta, puesto que sobre este tiempo se obvian el acceso a la aplicación donde el programador lee los requisitos o dudas que ha tenido que buscar en Internet y en la ayuda de MATLAB. Si se introdujese todo ese tiempo no contabilizado, el tiempo de desarrollo estaría más cerca de los 48 días marcados.

7.3 Líneas futuras de investigación

La aplicación producida en este trabajo es de utilidad, como se ha podido observar. A pesar de ello, existen varios caminos evidentes para continuar con este trabajo y mejorar así sus características.

El más claro es ampliar la cantidad de componentes de fusión disponibles en la aplicación de las pruebas del sistema de fusión multisensor. Aunque parezcan pocos los componentes, se ofrecen bastantes algoritmos de cada uno, implementados y testeados para funcionar de manera independiente. Como se explicó en el estado del arte (capítulo 2), quedan componentes del nivel 2, 3 y 4. Estos componentes ya dejan de ser tan genéricos y concretos, pasando a ser soluciones más ad-hoc, pero sería interesante desarrollarlas.

Además, sobre los componentes implementados existen una infinidad de algoritmos propuestos en la literatura los cuales serían de agrado tener usables en la aplicación.

Respecto a la aplicación de generación de contactos, también existen mejoras que realizar. Existen una gran cantidad de algoritmos de generación de trayectorias, sobre todo más centradas en un vehículo específico, teniendo en cuenta las capacidades de maniobra y aceleración del mismo. Sería interesante introducir un modelo de vehículo sobre el blanco, para simular su trayectoria de una manera más adecuada, no tan genérica. No se mueve igual un dron que un avión caza. Aunque se han modelado una gran cantidad de sensores, existen más sensores que los utilizados, y modelarlos ofrecería más capacidades a la herramienta. Además, es común incorporar sensores redundantes, por lo que permitir doblar los sensores o añadir tantos como se desee por blanco podría ser otra adición.

Por otro lado, aunque la interacción con la aplicación es suficientemente buena, existen funciones que serían agradecidas por los usuarios. Por ejemplo, la interfaz no está pensada para escalar, al no estar App Designer pensado siquiera para la adición de componentes de manera dinámica. Aun así, es posible realizarlo y sería útil para los usuarios.

Anexos

Anexo A

Entrevistas

En este anexo se aportan las entrevistas desarrolladas a dos expertos en la temática de este proyecto, la fusión de datos orientada a la vigilancia aérea. En la sección A.3.2.1 se expone la estructura de la entrevista a realizar, seguida por las preguntas a realizar en cada fase A.2. Finalmente, en la sección A.3, se aporta una transcripción de las respuestas de cada uno.

A.1 Estructura de las entrevistas

Las entrevistas son uno de los métodos de extracción de conocimiento más utilizados con los clientes. Esto es debido a que permite obtener una gran cantidad de información útil a partir de cada entrevistado. Además, este sector no es tan amplio como para acceder a la información mediante otras técnicas, como cuestionarios.

La entrevista a realizar está desarrollada y por tanto enfocada directamente a estos dos entrevistados, los cuales son expertos en la materia. El modelo de entrevista cuenta con cinco fases y se permite un tiempo flexible sobre cada una ellas, en función del desarrollo de la misma.

- **Introducción:** se explica al entrevistado el objetivo de esta entrevista y se le pregunta sobre las condiciones de la misma (duración, posibilidad de grabación en audio, etc.).
- **Calentamiento:** antes de profundizar en aspectos técnicos, se le pregunta sobre su perfil y su camino hasta llegar a ser experto en la materia. También se le pregunta sobre sus motivaciones con la misma, el tipo de proyectos en los que ha estado vinculado, etc. Con estas preguntas se clasifica al usuario según su perfil, de manera que utilizar su respuesta en mayor u menor medida para el proyecto.
- **Fase principal:** En esta fase se profundiza en los conceptos de interés para el proyecto. Se utilizan preguntas preparadas para obtener la información deseada sobre su experiencia en dos principales bloques.

Los bloques son, el primero enfocado a su relación con el cliente y facilidad de generar pruebas y el segundo más enfocado a los sistemas que utilizan actualmente en su grupo de trabajo.

- Fase de descanso: se le explica al entrevistado la idea del proyecto actual, de manera que pueda comprenderla y dada su experiencia validar la idea o realizar apuntes que mejoren el producto. También se le muestran los prototipos desarrollados, para que comprenda de mejor manera la idea explicada.
- Fase final: Se da las gracias al entrevistado y se cierra la entrevista.

A.2 Preguntas de las entrevistas

A continuación, se exponen las preguntas de cada fase de la entrevista.

A.2.1 Introducción

1. Antes de comenzar debo preguntarte si puedo grabar la conversación para que no se me pase ningún detalle.
2. Además, preguntarte ¿cuánto tiempo tenemos para la entrevista?, para organizar y hacer hincapié en algunas preguntas y no en otras.
3. ¿Tienes alguna pregunta antes de que comencemos la entrevista?
4. Comenzar a grabar.

A.2.2 Calentamiento

Cuéntame sobre ti,

1. ¿cuántos años tienes?
2. ¿qué estudios tienes?
3. ¿están relacionados con tu trabajo actual?
4. ¿cuál es tu trabajo actual?
5. ¿qué tipo de proyectos realizas?
6. ¿tienes empleados a tu cargo en tu puesto de trabajo actual?

Respecto a tu opinión sobre los sistemas de fusión de datos:

1. ¿qué nivel de JDL consideras el más interesante? ¿por qué?

2. ¿cuál es la aplicación de los sistemas de fusión que más te interesa a día de hoy?
3. ¿cómo ha evolucionado esta rama de investigación desde que estás en ella?
4. ¿cuál ha sido el avance más importante que has vivido?

A.2.3 Fase principal

A continuación, vamos a comenzar con preguntas generales, menos técnicas, sobre tu trabajo:

1. Vuestros proyectos, ¿surgen por la necesidad de un cliente, o son proyectos independientes?
2. En los proyectos para un cliente, ¿os encontráis con clientes que no tiene claras sus necesidades?
3. En estos casos, ¿es habitual desarrollar varias opciones para que el cliente escoja?
4. ¿En estos proyectos, es sencillo el proceso de testeo del software? ¿El cliente comprende las pruebas que realizáis?
5. ¿El cliente genera sus propias pruebas? ¿Tenéis un punto en común al realizar las pruebas?
6. ¿El cliente suele pedir tener ciertos parámetros que pueda personalizar sobre el sistema de fusión?

Pasemos a otro bloque de preguntas, más centrado en los componentes de un sistema de fusión de datos:

1. ¿Hasta qué nivel de JDL llegan los sistemas de fusión que implantáis?
2. ¿Lo desarrolláis siempre desde cero? ¿o partís de alguna base ya desarrollada y la portáis a otro entorno?
3. ¿Vuestros sistemas de fusión cuentan con muchas variables y parámetros que deben ser optimizados en función del escenario? ¿De qué manera lo ajustáis?
4. ¿Tenéis algún software para comprobar el funcionamiento del sistema de fusión en escenarios concretos? ¿Me puedes describir sus características?
5. ¿Estos escenarios, de donde salen? ¿los obtenéis vosotros / el cliente, los descargáis de Internet, o los generáis vosotros?

6. En el caso de utilizar una herramienta para generar, ¿qué características tiene la misma? ¿Es configurable? ¿En 2 o 3 dimensiones? ¿Se pueden seleccionar distintos sensores? ¿Se pueden ajustar los sensores?
7. ¿Esta herramienta tiene interfaz gráfica? ¿Le sería útil tenerla?
8. Por otro lado, para probar el sistema de fusión, contáis con algún software para observar su funcionamiento a lo largo del escenario? ¿qué características tiene?

A.2.4 Fase de descanso

A continuación, te voy a explicar la idea original de este proyecto, para que puedas valorarla y aportar cualquier sugerencia en alguno de sus aspectos, así como calificar si sería útil para tu trabajo.

- Se le pasan los prototipos desarrollados, impresos en papel -

Este proyecto, con el objetivo de generar un framework ampliable, capaz de probar cualquier tipo de configuración de un sistema de fusión, de manera que ser el centro de trabajo para optimizar parámetros y facilitar la tarea tanto a desarrolladores de sistemas de fusión, sus algoritmos, como para testear su funcionamiento en entornos reales. Con una herramienta así:

1. ¿Te sería útil una herramienta / conjunto de herramientas para generar posibles escenarios de manera gráfica?
2. ¿Te sería útil una herramienta de fusión de datos con una interfaz gráfica que permita la selección de cada componente, de manera que permitir la comparación entre ellos?
3. ¿Permitiría realizar labores de vuestros proyectos con mayor velocidad?
4. ¿Facilitaría la comprensión del sistema por parte del cliente?

A.2.5 Fase final

Muchas gracias por tu tiempo. Hemos terminado la entrevista, -apagar la grabadora-.

A.3 Transcripción de las entrevistas

En esta sección se recopila cada entrevista íntegra transcrita a texto.

A.3.1 Transcripción de la entrevista 1

La primera entrevista se realizó el día 24 de mayo del 2019, a las 18:24 horas.

A.3.1.1 Introducción

1. **Antes de comenzar debo preguntarte si puedo grabar la conversación para que no se me pase ningún detalle.**
Claro, puedes grabarlo.
2. **Además, preguntarte ¿cuánto tiempo tenemos para la entrevista?, para organizar y hacer hincapié en algunas preguntas y no en otras.**
Unos 20 minutos estaría bien.
3. **¿Tienes alguna pregunta antes de que comencemos la entrevista?**
Nada, vamos a ello.
4. **Comenzar a grabar.**

A.3.1.2 Calentamiento

Cuéntame sobre ti,

1. **¿cuántos años tienes?**
Tengo 34 años, soy del 85.
2. **¿qué estudios tienes?**
Soy ingeniero informático por la Universidad de Cádiz. Me gradué en el 2009. También he hecho un máster de Big Data aquí en Madrid hace unos años.
3. **¿están relacionados con tu trabajo actual?**
Sí claro.
4. **¿cuál es tu trabajo actual?**
Ahora mismo dirijo proyectos del departamento de sistemas de mi empresa.
5. **¿qué tipo de proyectos realizas?**
Los proyectos que dirijo principalmente se centran en la integración de datos para vehículos marítimos de defensa.
6. **¿tienes empleados a tu cargo en tu puesto de trabajo actual?**
Sí, tengo un equipo entero al que gestionar. Actualmente, son 8 personas. Analistas, desarrolladores...

Respecto a tu opinión sobre los sistemas de fusión de datos:

1. **¿qué nivel de JDL consideras el más interesante? ¿por qué?**
Yo no estoy demasiado informado sobre la teoría de los sistemas de fusión. La parte en la que centramos más recursos actualmente es la integración adecuada de todos los sensores que tenemos en el proyecto actual (Nivel 1).
2. **¿cuál es la aplicación de los sistemas de fusión que más te interesa a día de hoy?**
Lo más interesante realmente es lo que viene a continuación, la extracción de conocimiento, cuando ya tienes todos los barcos colocados y conoces su situación. El qué hacer y qué van a hacer los demás.
3. **¿cómo ha evolucionado esta rama de investigación desde que estás en ella?**
Yo no llevo demasiado tiempo con estos sistemas, más o menos cinco años. Por ahora no he notado un gran cambio.
4. **¿cuál ha sido el avance más importante que has vivido?**
Lo mismo te digo, no he percibido un gran avance. Siguen los mismos sensores, los algoritmos tampoco han cambiado...

A.3.1.3 Fase principal

A continuación, vamos a comenzar con preguntas generales, menos técnicas, sobre tu trabajo:

1. **Vuestros proyectos, ¿surgen por la necesidad de un cliente, o son proyectos independientes?**
Los nuestros siempre surgen por las necesidades de los clientes.
2. **En los proyectos para un cliente, ¿os encontráis con clientes que no tiene claras sus necesidades?**
El cliente siempre quiere lo último, pero no sabe qué es lo último. Por eso a veces hay que hablar más veces de las necesarias con ellos, para aclarar el alcance del proyecto.
3. **En estos casos, ¿es habitual desarrollar varias opciones para que el cliente escoja?**
Varias opciones en paralelo no, pero sí hemos tenido que cambiar de algoritmos porque a mitad de desarrollo no era lo que buscaban.
4. **¿En estos proyectos, es sencillo el proceso de testeo del software? ¿El cliente comprende las pruebas que realizáis?**
El cliente siempre nos pide una serie de pruebas unitarias, de manera que conozca los límites del software. También quiere ver funcionar el software, pero esto es más complicado, al necesitar de todos los sensores implementados en el vehículo, haberlo probado e integrado antes,

etc. El cliente no llega a comprender realmente todo lo que está detrás. No se involucra en exceso en la parte técnica.

5. **¿El cliente genera sus propias pruebas? ¿Tenéis un punto en común al realizar las pruebas?**

Sí claro, la empresa del cliente siempre tiene sus ingenieros que después revisan todo el código y prueban el software, de manera que lo optimicen para el funcionamiento real. Es complicado llegar a un punto común para el desarrollo de las pruebas con el cliente, puesto que ellos no te explican todas las características técnicas que requieres saber para imitar un escenario "real".

6. **¿El cliente suele pedir tener ciertos parámetros que pueda personalizar sobre el sistema de fusión?**

Sí claro, ellos quieren que el software sea customizable, para que sus ingenieros lo puedan ajustar de manera sencilla. Con un fichero de configuración poder cambiar el funcionamiento es lo más habitual.

Pasemos a otro bloque de preguntas, más centrado en los componentes de un sistema de fusión de datos:

1. **¿Hasta qué nivel de JDL llegan los sistemas de fusión que implantáis?**

No sé decirte cuál es. Yo te digo qué labores hace nuestro software a grandes rasgos y me dices. El software recibe las señales de los sensores, las almacena, las procesa, realiza la estimación y asociación de vehículos cercanos. Con esta información se saca el barco con la traza fusionada, y con esta información se generan probabilidades de varias decisiones a tomar. Conclusión: nivel 3.

2. **¿Lo desarrolláis siempre desde cero? ¿o partís de alguna base ya desarrollada y la portáis a otro entorno?**

Los implementamos desde cero. Al final, lo habitual es utilizar los mismos algoritmos, con ligeras variaciones, por lo que al final sabemos hacerlo. Es posible que si miras varios distintos parezcan salir de una misma plantilla, eso puede ser (risas). Al final aprendes a hacerlo de una manera.

3. **¿Vuestros sistemas de fusión cuentan con muchas variables y parámetros que deben ser optimizados en función del escenario? ¿De qué manera lo ajustáis?**

Sí, tenemos decenas de algoritmos trabajando en paralelo en el sistema, te puedes imaginar cuantas variables hay que ajustar. El sistema tiene un algoritmo para ajustar parámetros automáticamente, pero siempre es mejor colocar los parámetros a mano por un experto, en tiempo real.

4. **¿Tenéis algún software para comprobar el funcionamiento del sistema de fusión en escenarios concretos? ¿Me puedes describir sus características?**

No existe ningún software especial para testear el funcionamiento del sistema. Se pueden realizar pruebas introduciendo al sistema las entradas oportunas y comprobando su salida, pero nada más específico.

5. **¿Estos escenarios, de donde salen? ¿los obtenéis vosotros / el cliente, los descargáis de Internet, o los generáis vosotros?**

Las entradas al sistema de fusión, o escenario como dices tú, las generamos tanto en la realidad, con sensores y vehículos, como simulando trayectorias de vehículos.

6. **En el caso de utilizar una herramienta para generar, ¿qué características tiene la misma? ¿Es configurable? ¿En 2 o 3 dimensiones? ¿Se pueden seleccionar distintos sensores? ¿Se pueden ajustar los sensores?**

Para simular únicamente colocamos puntos a lo largo de una trayectoria inventada y le aplicamos un ligero ruido. Es bastante rudimentario, al ser pruebas simples. Cuando se requieren pruebas de verdad se generan como te he dicho.

7. **¿Esta herramienta tiene interfaz gráfica? ¿Le sería útil tenerla?**

No tiene no. Podría estar bien para seleccionar la trayectoria a mano.

8. **Por otro lado, para probar el sistema de fusión, contáis con algún software para observar su funcionamiento a lo largo del escenario? ¿qué características tiene?**

No se cuenta con ninguna herramienta concreta. La salida del sistema de fusión queda almacenada, por lo que se puede analizar. Además, el sistema genera un log el cual detecta ciertos comportamientos.

A.3.1.4 Fase de descanso

A continuación, te voy a explicar la idea original de este proyecto, para que puedas valorarla y aportar cualquier sugerencia en alguno de sus aspectos, así como calificar si sería útil para tu trabajo.

- Se le pasan los prototipos desarrollados, impresos en papel -

Este proyecto, con el objetivo de generar un framework ampliable, capaz de probar cualquier tipo de configuración de un sistema de fusión, de manera que ser el centro de trabajo para optimizar parámetros y facilitar la tarea tanto a desarrolladores de sistemas de fusión, sus algoritmos, como para testear su funcionamiento en entornos reales. Con una herramienta así:

1. **¿Te sería útil una herramienta / conjunto de herramientas para generar posibles escenarios de manera gráfica?**

Por supuesto. Actualmente lo realizamos de manera muy rudimentaria. Además, si se pueden escoger como comentas distintos sensores y customizarlos, lo hace una herramienta muy útil para realizar pruebas antes de implementarlo en realidad.

2. **¿Te sería útil una herramienta para lanzar distintas pruebas sobre un sistema de fusión de datos con una interfaz gráfica, de manera que permitir la comparación entre ellos?**

Sí. Precisamente para mostrar a los clientes cuál es el mejor sistema/algoritmo para un escenario concreto sería muy útil. Si de manera gráfica son capaces de observar porqué uno y no otro, sería una gran mejora para ahorrar tiempo y comprender mejor el problema.

3. **¿Permitiría realizar labores de vuestros proyectos con mayor velocidad?**

Sin duda. Agilizaría todo el proceso de pruebas de los algoritmos y facilitaría la comprensión de cualquier posible fallo.

4. **¿Facilitaría la comprensión del sistema por parte del cliente?**

Sí, como te he dicho. Sería ideal contar con una herramienta así.

A.3.1.5 Fase final

Muchas gracias por tu tiempo. Hemos terminado la entrevista, -apagar la grabadora-.

A.3.2 Transcripción de la entrevista 2

La segunda entrevista se realizó el día 03 de junio de 2019, a las 12:33 horas.

A.3.2.1 Introducción

1. **Antes de comenzar debo preguntarte si puedo grabar la conversación para que no se me pase ningún detalle.**

Sin problema, te vendrá bien.

2. **Además, preguntarte ¿cuánto tiempo tenemos para la entrevista?, para organizar y hacer hincapié en algunas preguntas y no en otras.**

Lo que necesites.

3. **¿Tienes alguna pregunta antes de que comencemos la entrevista?**

Nada.

4. Comenzar a grabar.

A.3.2.2 Calentamiento

Cuéntame sobre ti,

1. **¿cuántos años tienes?**

42 años, ya estoy mayor.

2. **¿qué estudios tienes?**

Soy ingeniero de telecomunicaciones, e hice un doctorado en la UPM sobre sistemas de fusión.

3. **¿están relacionados con tu trabajo actual?**

Sí, actualmente soy investigador de la Universidad de Cádiz, dedicado a la ciencia y tecnología informática.

4. **¿cuál es tu trabajo actual?**

Ya me he adelantado jaja. Imparto clase a alumnos de informática y telemática, investigo y de vez en cuando sale algún que otro proyecto de sistemas de fusión.

5. **¿qué tipo de proyectos realizas?**

De vez en cuando sale un proyecto de alguna empresa aeronáutica, para hacer o revisar una parte de sus sistemas de fusión. Nada demasiado grande.

6. **¿tienes empleados a tu cargo en tu puesto de trabajo actual?**

No, tengo compañeros en el grupo de investigación y algún que otro alumno becado que nos ayuda.

Respecto a tu opinión sobre los sistemas de fusión de datos:

1. **¿qué nivel de JDL consideras el más interesante? ¿por qué?**

Lo más interesante se encuentra en los niveles dos o tres. La inteligencia a extraer a partir de los escenarios es donde está el meollo ahora mismo. La precisión de los sensores y su fusión a día de hoy es suficientemente buena.

2. **¿cuál es la aplicación de los sistemas de fusión que más te interesa a día de hoy?**

Lo que te comento, la predicción de los escenarios.

3. **¿cómo ha evolucionado esta rama de investigación desde que estás en ella?**

Ha evolucionado bastante rápido, a la par que la tecnología. Los sensores son mejores, se pueden hacer algoritmos mucho más potentes...

4. **¿cuál ha sido el avance más importante que has vivido?**

¿En sistemas de fusión de datos? Los filtros de partículas quizá son lo más interesante. Además, se utilizan bastante a día de hoy. Al final funcionan prácticamente solos y la computación ahora permite muchísimas partículas, por lo que funcionan muy muy bien.

A.3.2.3 Fase principal

A continuación, vamos a comenzar con preguntas generales, menos técnicas, sobre tu trabajo:

1. **Vuestros proyectos, ¿surgen por la necesidad de un cliente, o son proyectos independientes?**

Proyectos con empresas siempre por sus necesidades. Al final en la universidad tienes que crearte tus propios proyectos y tratar de colocarlos. Así que, un poco de todo.

2. **En los proyectos para un cliente, ¿os encontráis con clientes que no tiene claras sus necesidades?**

En nuestra situación, suelen tener bastante orientado el proyecto, al ser como te he comentado, pequeñas porciones del sistema las que nos corresponden.

3. **En estos casos, ¿es habitual desarrollar varias opciones para que el cliente escoja?**

No, nosotros siempre hacemos exclusivamente el trabajo que nos han solicitado.

4. **¿En estos proyectos, es sencillo el proceso de testeo del software? ¿El cliente comprende las pruebas que realizáis?**

En la mayoría de nuestros casos sí, porque no tenemos que hacerlo nosotros (risas). Nos suelen pedir que desarrollemos el software y nos pasan un escenario concreto para observar que funciona correctamente. Nada más.

5. **¿El cliente genera sus propias pruebas? ¿Tenéis un punto en común al realizar las pruebas?**

Entiendo que sí, que después lo revisarán ellos claro. Como te he dicho, sí, hacemos lo que nos piden.

6. **¿El cliente suele pedir tener ciertos parámetros que pueda personalizar sobre el sistema de fusión?**

Claro, ya sabes que estos software deben serlo. Siempre hay unas variables que deben ser accesibles para toquetearlo un poco. En ningún caso se cambia de algoritmo, eso no.

Pasemos a otro bloque de preguntas, más centrado en los componentes de un sistema de fusión de datos:

1. ¿Hasta qué nivel de JDL llegan los sistemas de fusión que implantáis?

Hemos hecho de todo. Desde el nivel 1 hasta el 3. Estimación de posición, asociación, predicción, ...

2. ¿Lo desarrolláis siempre desde cero? ¿o partís de alguna base ya desarrollada y la portáis a otro entorno?

Tenemos el código desarrollado en MATLAB, por lo que al final portamos el algoritmo solicitado y luego hacemos los cambios oportunos para el cliente.

3. ¿Vuestros sistemas de fusión cuentan con muchas variables y parámetros que deben ser optimizados en función del escenario? ¿De qué manera lo ajustáis?

Mediante las variables, como te he comentado. Realmente no tenemos un sistema de fusión montado per sé. Tenemos los algoritmos y cuando lo necesitamos, los unimos.

4. ¿Tenéis algún software para comprobar el funcionamiento del sistema de fusión en escenarios concretos? ¿Me puedes describir sus características?

No tenemos ningún software que lo compruebe. Tenemos el log y los resultados, los cuales analizamos montando gráficas y las tablas de MATLAB.

5. ¿Estos escenarios, de donde salen? ¿los obtenéis vosotros / el cliente, los descargáis de Internet, o los generáis vosotros?

Los escenarios como te he comentado, nos los pasan los clientes. Nosotros no tenemos la capacidad de obtener los datos.

6. Por otro lado, para probar el sistema de fusión, contáis con algún software para observar su funcionamiento a lo largo del escenario? ¿qué características tiene?

No, no tenemos nada.

A.3.2.4 Fase de descanso

A continuación, te voy a explicar la idea original de este proyecto, para que puedas valorarla y aportar cualquier sugerencia en alguno de sus aspectos, así como calificar si sería útil para tu trabajo.

- Se le pasan los prototipos desarrollados, impresos en papel -

Este proyecto, con el objetivo de generar un framework ampliable, capaz de probar cualquier tipo de configuración de un sistema de fusión, de manera

que ser el centro de trabajo para optimizar parámetros y facilitar la tarea tanto a desarrolladores de sistemas de fusión, sus algoritmos, como para testear su funcionamiento en entornos reales. Con una herramienta así:

1. **¿Te sería útil una herramienta / conjunto de herramientas para generar posibles escenarios de manera gráfica?**

Claro. Aparte de su utilidad para todo el mundo que desarrolle este tipo de sistemas. A nosotros en concreto no nos aportaría directamente, porque no generamos escenarios. Pero seguro que hay a mucha gente a quién una herramienta trabajada por y para generar contactos de escenarios le pueda facilitar su labor.

2. **¿Te sería útil una herramienta para lanzar distintas pruebas sobre un sistema de fusión de datos con una interfaz gráfica, de manera que permitir la comparación entre ellos?**

Esta otra herramienta le veo utilidad a la hora de comparar varios algoritmos. Además, puede ser útil para enseñar a los alumnos el funcionamiento y la complejidad de estos sistemas.

3. **¿Permitiría realizar labores de vuestros proyectos con mayor velocidad?**

Más que rapidez, veo que aportaría claridad a la comprensión de los conceptos. Al final, esto tiene fundamentos matemáticos, que aún con una interfaz no ves de manera clara.

4. **¿Facilitaría la comprensión del sistema por parte del cliente?**

Si está mascado y claro, lo verá y comprenderá de manera sencilla, sí.

A.3.2.5 Fase final

Muchas gracias por tu tiempo. Hemos terminado la entrevista, -apagar la grabadora-.

Anexo B

Requisitos

En este anexo se aportan los requerimientos finales del proyecto, introducidos en la sección 4.6.

B.1 Requisitos funcionales

B.1.1 Requisitos del simulador

Identificador	RF.1
Nombre	Selección del tipo de simulación
Descripción	La aplicación permite generar los contactos de la simulación de un vehículo en concreto, de una estación base en tierra, o de todos los vehículos integrantes del escenario
Razón	Esta opción permite a la herramienta ser más completa. Es útil para usuarios con necesidades de niveles JDL elevadas, para simular tanto los contactos recibidos por vehículos propios como enemigos

Tab.2.1: Requisito funcional 1

Identificador	RF.2
Nombre	Gestión de blancos
Descripción	La aplicación permitirá al usuario crear, editar y borrar blancos en el simulador de contactos
Razón	Es necesario permitir al usuario capacidad de ampliar, ajustar y corregir el escenario que desea realizar

Tab.2.2: Requisito funcional 2

Identificador	RF.3
Nombre	Creación de puntos de paso en un blanco
Descripción	La aplicación permite al usuario crear los puntos de paso de un blanco en el simulador de contactos, utilizando un mapa para establecer su posición
Razón	Es necesario facilitar al usuario la colocación de los puntos de paso para conocer la trayectoria del mismo

Tab.2.3: Requisito funcional 3

Identificador	RF.4
Nombre	Parámetros de cada punto de paso
Descripción	La aplicación permitirá en cualquier momento modificar los siguientes valores de cada punto de paso de un blanco: <ul style="list-style-type: none"> • Latitud • Longitud • Altitud • Tiempo • Velocidad • Aceleración • Retraso
Razón	Cada punto de paso modela el vehículo en dicho momento, por lo que el usuario debe poder modelar la trayectoria de cada vehículo en dicho instante

Tab.2.4: Requisito funcional 4

Identificador	RF.5
Nombre	Gestión de puntos de paso en un blanco
Descripción	La aplicación permite al usuario editar y borrar los puntos de paso existentes en un blanco en el simulador de contactos. Además, se muestran el punto de paso en el mapa, y resaltado el seleccionado
Razón	Es necesario permitir al usuario capacidad de corregir errores y ajustar el escenario a sus necesidades

Tab.2.5: Requisito funcional 5

Identificador	RF.6
Nombre	Selección de sensores en cada blanco
Descripción	Cada vehículo cuenta con una serie de sensores concreta. De todos los sensores implementados, el usuario puede seleccionar los que desea utilizar para recibir contactos
Razón	El usuario debe poder elegir qué sensores se están utilizando en el escenario que está realizando

Tab.2.6: Requisito funcional 6

Identificador	RF.7
Nombre	Selección de cooperación en sensores cooperativos con otros blancos
Descripción	Existen sensores cooperativos que requieren de la cooperación del blanco detectado. Cada blanco puede cooperar mediante dicho sensor, o no hacerlo
Razón	El usuario debe poder desarrollar de manera específica el escenario deseado

Tab.2.7: Requisito funcional 7

Identificador	RF.8
Nombre	Customización de cada sensor en cada blanco
Descripción	Cada sensor de cada blanco puede ser distinto, por lo que el usuario debe tener la opción de ajustar cada parámetro de cada sensor a su gusto
Razón	Cada sensor tiene características muy distintas según el lugar donde se implementa y su necesidad a saciar

Tab.2.8: Requisito funcional 8

Identificador	RF.9
Nombre	Exportar blanco
Descripción	La aplicación permite exportar las características de un blanco concreto, para incorporar tanto sus puntos de paso como sus sensores en futuros escenarios

Razón	Esta funcionalidad permite al usuario facilitar su trabajo futuros proyectos
Tab.2.9: Requisito funcional 9	
Identificador	RF.10
Nombre	Importar blanco
Descripción	La aplicación permite importar las características de un blanco concreto, para incorporar tanto sus puntos de paso como sus sensores en futuros escenarios
Razón	Esta funcionalidad permite al usuario facilitar su trabajo futuros proyectos
Tab.2.10: Requisito funcional 10	
Identificador	RF.11
Nombre	Exportar características de la generación de contactos
Descripción	La aplicación permite exportar las características de toda la generación de contactos, tanto para continuar posteriormente como para servir de base de posibles simulaciones
Razón	Esta funcionalidad permite al usuario facilitar su trabajo futuros proyectos
Tab.2.11: Requisito funcional 11	
Identificador	RF.12
Nombre	Importar características de la generación de contactos
Descripción	La aplicación permite importar las características de toda la generación de contactos, tanto para continuar posteriormente como para servir de base de posibles simulaciones
Razón	Esta funcionalidad permite al usuario facilitar su trabajo futuros proyectos
Tab.2.12: Requisito funcional 12	
Identificador	RF.13

Nombre	Control de los valores de cada blanco
Descripción	La aplicación comprueba todos los valores antes de comenzar la generación de los contactos, mostrando un aviso al usuario en caso de encontrar algún fallo en los mismos
Razón	Esta funcionalidad permite al usuario detectar problemas en los parámetros y ayudarlo a encontrarlos

Tab.2.13: Requisito funcional 13

Identificador	RF.14
Nombre	Generación de los contactos
Descripción	El usuario tiene la opción de generar los contactos a partir de la información y selecciones que ha realizado
Razón	La aplicación ejecuta el algoritmo de generación de los contactos y le presenta los resultados al usuario en el mapa

Tab.2.14: Requisito funcional 14

Identificador	RF.15
Nombre	Trayectorias en el mapa
Descripción	La aplicación muestra la información de la simulación en un mapa con el que el usuario puede interactuar, acercarse y moverse. Se muestran tanto los puntos de paso, como la trayectoria generada de cada blanco, como los contactos de cada sensor sobre cada contacto
Razón	Esta funcionalidad permite al usuario entender y obtener de manera gráfica la generación que está realizando

Tab.2.15: Requisito funcional 15

Identificador	RF.16
Nombre	Información auxiliar en el mapa

Descripción	Cada trayectoria mostrada en el mapa, se acompaña de información extra si el usuario lo selecciona en el mapa. Esta información indica el identificador del blanco y sensor, así como información del punto concreto, como el tiempo del mismo o su velocidad
--------------------	---

Razón	Esta funcionalidad permite al usuario entender y obtener de manera gráfica la simulación realizada
--------------	--

Tab.2.16: Requisito funcional 16

Identificador	RF.17
----------------------	-------

Nombre	Gestión del mapa
---------------	------------------

Descripción	La aplicación cuenta con un componente auxiliar para modificar el comportamiento del mapa y mostrar únicamente la información deseada
--------------------	---

Razón	Esta funcionalidad facilita al usuario la comprobación de la información que desea observar, y/o comparar diferentes trayectorias entre sí
--------------	--

Tab.2.17: Requisito funcional 17

Identificador	RF.18
----------------------	-------

Nombre	Guardado de los contactos
---------------	---------------------------

Descripción	La aplicación deberá permitir al usuario almacenar los contactos generados en un fichero, así como la trayectoria del centro de fusión, para utilizarlos en la herramienta de fusión de datos
--------------------	---

Razón	Esta funcionalidad permite al usuario, una vez se ha terminado la simulación, guardar la misma en el fichero y utilizar ese escenario en la herramienta de simulación
--------------	---

Tab.2.18: Requisito funcional 18

B.1.2 Requisitos de la app de fusión de datos

Identificador	RF.19
----------------------	-------

Nombre	Importar contactos
---------------	--------------------

Descripción	La aplicación permitirá cargar los contactos generados con la herramienta de simulación en el sistema de fusión. Además, se muestran en el mapa
Razón	Con esta funcionalidad cumple una labor imprescindible para el funcionamiento del sistema

Tab.2.19: Requisito funcional 19

Identificador	RF.20
Nombre	Gestión de pruebas de fusión
Descripción	La aplicación permite al usuario crear, editar y borrar las pruebas a realizar sobre el conjunto de contactos importado
Razón	Es necesario permitir al usuario capacidad de corregir errores y ajustar cada prueba del sistema de fusión a sus necesidades

Tab.2.20: Requisito funcional 20

Identificador	RF.21
Nombre	Selección de componentes
Descripción	En cada prueba definida, la aplicación permite seleccionar una serie de componentes del sistema de fusión a utilizar. Algunos de ellos son de uso obligado
Razón	Con esta funcionalidad se permite al usuario customizar cada prueba del sistema de fusión

Tab.2.21: Requisito funcional 21

Identificador	RF.22
Nombre	Selección de algoritmos de componente
Descripción	La aplicación permite seleccionar, para cada componente del sistema, entre distintos algoritmos implementados, así como la configuración del mismo, si la tiene
Razón	Con esta funcionalidad se facilita la labor al usuario de la aplicación, comparar entre distintos funcionamientos

Tab.2.22: Requisito funcional 22

Identificador	RF.23
Nombre	Exportar prueba
Descripción	La aplicación permite exportar la configuración de una prueba del sistema de fusión, de manera que esta misma sea reutilizable en el futuro
Razón	Con esta funcionalidad se facilita la labor al usuario de la aplicación
Tab.2.23: Requisito funcional 23	

Identificador	RF.24
Nombre	Importar prueba
Descripción	La aplicación permite exportar la configuración de una prueba del sistema de fusión, de manera que esta misma sea reutilizable en el futuro
Razón	Con esta funcionalidad se facilita la labor al usuario de la aplicación
Tab.2.24: Requisito funcional 24	

Identificador	RF.25
Nombre	Exportar configuración del conjunto de pruebas de fusión
Descripción	La aplicación permitirá exportar la configuración actual del sistema de fusión, de manera que esta misma sea reutilizable en el futuro
Razón	Con esta funcionalidad se facilita la labor al usuario de la aplicación
Tab.2.25: Requisito funcional 25	

Identificador	RF.26
Nombre	Importar configuración del conjunto de pruebas de fusión
Descripción	La aplicación permitirá importar la configuración actual del sistema de fusión, de manera que esta misma sea reutilizable en el futuro

Razón	Con esta funcionalidad se facilita la labor al usuario de la aplicación
	Tab.2.26: Requisito funcional 26
Identificador	RF.27
Nombre	Control de los valores de cada prueba
Descripción	La aplicación comprueba todos los valores antes de comenzar la ejecución de cada sistema de fusión a probar, mostrando un aviso al usuario en caso de encontrar algún fallo en los mismos
Razón	Esta funcionalidad permite al usuario detectar problemas en los parámetros y ayudarle a encontrarlos
	Tab.2.27: Requisito funcional 27
Identificador	RF.28
Nombre	Implementación de algoritmos de estimación
Descripción	La aplicación cuenta debe contar con diversos algoritmos del componente de estimación de estado implementados y accesible para las pruebas del sistema de fusión. En concreto, se deben implementar: EKF, EnKF, UKF, IMM y PF
Razón	Esta funcionalidad permite al usuario realizar pruebas con el sistema de fusión
	Tab.2.28: Requisito funcional 28
Identificador	RF.29
Nombre	Implementación de varios algoritmos de asociación
Descripción	La aplicación cuenta debe contar con diversos algoritmos del componente de estimación de estado implementados y accesible para las pruebas del sistema de fusión. En concreto, se deben implementar: Vecino más próximo (NN), Asociación Probabilística de Datos (PDA), Seguimiento con Hipótesis Múltiples (MHT) y Algoritmo de Munkres
Razón	Esta funcionalidad permite al usuario realizar aún más pruebas con el sistema de fusión

Tab.2.29: Requisito funcional 29

Identificador	RF.30
Nombre	Estructura preparada para ampliación con más componentes de sistemas de fusión multisensor
Descripción	La aplicación debe contar con una estructura de datos y lógica robusta para permitir una futura ampliación de componentes de manera sencilla
Razón	El usuario potencial de la aplicación es un investigador que desarrollará y testeará diversos algoritmos en la aplicación

Tab.2.30: Requisito funcional 30

Identificador	RF.31
Nombre	Implementación de varias métricas de rendimiento
Descripción	La aplicación cuenta con el algoritmo Filtro de partículas (PF) implementado y accesible para las pruebas del sistema de fusión. En concreto, se deben implementar: Tiempo de cómputo, Innovación cuadrática normalizada (NIS), Error cuadrático de estimación normalizado (NEES) y Raíz del error cuadrático medio (RMSE)
Razón	Esta funcionalidad permite al usuario comprender mejor las pruebas que realice con el sistema de fusión

Tab.2.31: Requisito funcional 31

Identificador	RF.32
Nombre	Ejecución de las pruebas desarrolladas
Descripción	La aplicación permite al usuario ejecutar el algoritmo del sistema de fusión de datos con las características que ha seleccionado
Razón	Se consigue uno de los objetivos del trabajo

Tab.2.32: Requisito funcional 32

Identificador	RF.33
Nombre	Comparación de los resultados entre pruebas
Descripción	Una vez ejecutadas las pruebas, la aplicación permite al usuario observar toda la información que se ha generado y compararlas con gráficos específicos para cada métrica, o directamente en el mapa
Razón	Se facilita el análisis a realizar por el usuario

Tab.2.33: Requisito funcional 33

Identificador	RF.34
Nombre	Guardar resultados
Descripción	La aplicación permitirá, una vez realizada una simulación, si guardar los resultados para observarlos otra vez en el futuro
Razón	Con esta funcionalidad se permite una labor imprescindible para el análisis de distintos comportamientos de la herramienta

Tab.2.34: Requisito funcional 34

B.2 Requisitos de datos

Identificador	RD.1
Nombre	Unidades en Sistema Internacional
Descripción	Todos los datos almacenados y procesados por ambas aplicaciones se trabajan en Sistema Internacional
Razón	Facilita la comprensión por parte de todo experto de fusión de datos

Tab.2.35: Requisito de datos 1

Identificador	RD.2
Nombre	Almacenaje de datos
Descripción	Toda la información generada se almacena en ficheros CSV, para facilitar su lectura, tanto por la aplicación como otros software.

Razón	No se obliga a utilizar la herramienta en todos los entornos
--------------	--

Tab.2.36: Requisito de datos 2

Identificador	RD.3
----------------------	------

Nombre	Aviso sobre el tipo de dato
---------------	-----------------------------

Descripción	Se permitirá al usuario conocer cada variable y sus unidades desde la misma aplicación
--------------------	--

Razón	Facilita la comprensión y uso de las aplicaciones por parte del usuario, al contar con demasiadas variables con unidades diferentes
--------------	---

Tab.2.37: Requisito de datos 3

Identificador	RD.4
----------------------	------

Nombre	Accesibilidad de la información
---------------	---------------------------------

Descripción	Toda la información utilizada en la aplicación es accesible y modificable desde la misma, permitiendo al usuario conocer todas las variables que entran en sus pruebas
--------------------	--

Razón	Facilitar la comprensión del trabajo realizado
--------------	--

Tab.2.38: Requisito de datos 4

B.3 Requisitos de ambiente

Identificador	RA.1
----------------------	------

Nombre	Climatología del entorno
---------------	--------------------------

Descripción	El lugar de utilización esperado para la aplicación es un espacio interior sin climas adversos. Aun así, la aplicación trata de ofrecer una interfaz con alto contraste para facilitar la visión
--------------------	--

Razón	De esta manera se ofrece al usuario una interfaz adecuada para la visión en todas condiciones
--------------	---

Tab.2.39: Requisito de ambiente 1

Identificador	RA.2
Nombre	Idioma de la aplicación
Descripción	La aplicación se utilizará en cualquier lugar del mundo. Por ello, el idioma escogido para la interfaz es el inglés. No obstante, se permitirá modificar el idioma
Razón	Los expertos de fusión de datos se comunican en inglés, pero se utilizará también en español

Tab.2.40: Requisito de ambiente 2

Identificador	RA.3
Nombre	Compartir información
Descripción	Se espera que el usuario comparta información / escenarios / resultados con otros usuarios de la aplicación
Razón	Los expertos de fusión de datos comparten información y debaten entre ellos

Tab.2.41: Requisito de ambiente 3

B.4 Requisitos de usuario

Identificador	RU.1
Nombre	Facilidades para usuario básico
Descripción	En las herramientas se cuenta con avisos, mensajes y correcciones para facilitar la experiencia y aprendizaje al usuario básico
Razón	Los usuarios menos expertos necesitan comprender las funciones que se ofrecen en la herramienta

Tab.2.42: Requisito de usuario 1

Identificador	RU.2
Nombre	Distintos caminos para usuarios expertos
Descripción	De igual manera, al usuario más experto en la materia y las aplicaciones, se le otorgan atajos para realizar pruebas de manera más veloz

Razón	El usuario experto necesita llegar a su objetivo de una manera más rápida
--------------	---

Tab.2.43: Requisito de usuario 2

B.5 Requisitos de usabilidad

Identificador	RX.1
Nombre	Interfaces similares
Descripción	Ambas aplicaciones deben contar con una interfaz sencilla y común, para facilitar la comprensión por parte del usuario
Razón	El objetivo de la aplicación es analizar escenarios, no aprender a utilizar la interfaz. De esta manera, se enfoca el trabajo en el análisis

Tab.2.44: Requisito de usabilidad 1

Identificador	RX.2
Nombre	Facilidad de aprendizaje
Descripción	La aplicación debe ser sencilla de aprender, de manera que un usuario entienda su funcionamiento de manera rápida
Razón	El objetivo de la aplicación es analizar escenarios, no aprender a utilizar la interfaz. De esta manera, se enfoca el trabajo en el análisis

Tab.2.45: Requisito de usabilidad 2

Identificador	RX.3
Nombre	Aplicaciones informativas
Descripción	Ambas aplicaciones, en sus tareas más pesadas, informan sobre su estado respecto a la tarea, para informar al usuario de que el procesamiento está siendo correcto y estime el tiempo restante
Razón	Se mantiene al usuario informado y de esta manera obtiene un trato agradable con la aplicación

Tab.2.46: Requisito de usabilidad 3

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] 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.
- [3] G. S. Santos and C. Müller, “Limits and Future GnsS Solutions for Aircraft Landing in Low Visibility Conditions,” no. December, 2013.
- [4] IBM, “10 Key Marketing Trends for 2017,” *Ibm.com*, p. 18, 2016. [Online]. Available: [https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=WRL12345USEN\(13.10.2017\)](https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=WRL12345USEN(13.10.2017))
- [5] F. E. White, “JDL, Data Fusion Lexicon,” *Technical Panel For C3*, vol. 15, no. 0704, p. 15, 1991.
- [6] D. L. Hall and J. Llinas, “An introduction to multisensor data fusion,” *Proceedings of the IEEE*, vol. 85, no. 1, pp. 6–23, 1997.
- [7] 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>
- [8] B. V. Dasarathy, “Sensor fusion potential exploitation-innovative architectures and illustrative applications,” *Proceedings of the IEEE*, vol. 85, no. 1, pp. 24–38, 1997.
- [9] 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.

- [10] 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.
- [11] Federico Castanedo, "A Review of Data Fusion Techniques," *The Scientific World Journal*, vol. 2013, p. 19, 2013.
- [12] 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>
- [13] 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>
- [14] P. C. Mahalanobis, "On the generalized distance in statistics," *Proceedings of the National Institute of Sciences (Calcutta)*, vol. 2, pp. 49–55, 1936.
- [15] 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>
- [16] Y. Bar-Shalom and E. Tse, "Tracking in a cluttered environment with probabilistic data association," *Automatica*, vol. 11, no. 5, pp. 451–460, 1975.
- [17] 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.
- [18] 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/>
- [19] 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.
- [20] J. Munkres, "Algorithms for the Assignment and Transportation Problems," *Journal of the Society for Industrial and Applied Mathematics*, 1957.

- [21] 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.
- [22] 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.
- [23] 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.
- [24] J. Gomez-Romero, M. Patricio, J. García, and J. Molina, “Ontological representation of context knowledge for visual data fusion,” *Information Fusion (FUSION)*, 2009, 2009.
- [25] 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>
- [26] 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.
- [27] P. Borkowski, “The ship movement trajectory prediction algorithm using navigational data fusion,” *Sensors (Switzerland)*, vol. 17, no. 6, 2017.
- [28] 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.
- [29] Toyon, “Simulation of the Locations and Attack of Mobile Enemy Missiles (SLAMEM),” *NAVAIR Public Release*, p. 1, 2019. [Online]. Available: <http://www.toyon.com/products/slamem>
- [30] N. N. Okello, D. Tang, and D. W. McMichael, “TRACKER: A sensor fusion simulator for generalised tracking,” *IDC 1999 - 1999 Information, Decision and Control, Data and Information Fusion Symposium, Signal Processing and Communications Symposium and Decision and Control Symposium: Proceedings*, pp. 359–364, 1999.
- [31] Ternion Corporation, “FLAMES Simulator Framework.”

- [32] M. J. Carlotto and I. Kadar, “A Multi-source Report-level Simulator for Fusion Research,” *Signal Processing, Sensor Fusion, and Target Recognition XII*, vol. 5096, no. August 2003, pp. 364–378, 2003.
- [33] 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.
- [34] P. A. Thomas, J. Barr, B. Balaji, and K. White, “An open source framework for tracking and state estimation (‘Stone Soup’),” *Signal Processing, Sensor/Information Fusion, and Target Recognition XXVI*, vol. 10200, p. 1020008, 2017.
- [35] BOE, “Real Decreto Legislativo 1/1996, de 12 de abril. Ley de Propiedad Intelectual,” 1996.
- [36] D. Amigo, “Técnicas avanzadas de estimación y filtrado: análisis y evaluación en entornos marítimos,” Ph.D. dissertation, Carlos III, 2019.