

En Profundidad

Sistema SAFETERM: Safe Flight Termination System

Autor: Javier Ferrero Micó, GMV Aerospace and Defence.

Palabras clave: inteligencia artificial, estandarización, UAS, emergencia, visión por computador, aprendizaje profundo, terminación del vuelo.

Líneas I+D+i ETID relacionadas: 2.5.2, 7.3.1, 7.3.2, 7.3.5, 7.4.1, 7.4.2.

Introducción

GMV ha desarrollado, bajo contrato con la Agencia de Defensa Europea (EDA) y con la colaboración de AERTEC, el sistema SAFETERM (*Safe Autonomous Flight Termination*) para la mejora de los actuales sistemas de terminación de vuelo de aeronaves pilotadas por control remoto (UAS) de media altitud y larga duración y el concepto general de recuperación de emergencia (ER). El sistema SAFETERM se fundamenta en el uso de tecnologías de inteligencia artificial de última generación para aumentar el nivel de seguridad en situaciones específicas de emergencia, que conducen a una terminación de vuelo. Este concepto es especialmente beneficioso para emergencias que se produzcan en el modo de enlace de datos de pérdida de mando y control (C2). Es un proyecto piloto que ha sido utilizado por autoridades de estandarización, como caso de uso para el diseño de las guías para la certificación y aeronavegabilidad de sistemas basados en inteligencia artificial embarcados en Vehículos Aéreos No Tripulados (UAS por sus siglas en inglés) y en aviación tripulada.

Contexto y objetivo

Los Sistemas de Recuperación de Emergencia (ER) y de Terminación de Vuelo (FTS) de los UAS de mediana altitud y larga duración (MALE, por sus siglas en inglés), que actualmente gestionan los fallos del sistema en el modo de pérdida del enlace C2, se basan en procedimientos preprogramados en

función de posibles sucesos adicionales (es decir, condiciones de fallos combinados con la pérdida del enlace C2). Se requerirá, por tanto, una predefinición de Áreas de Terminación de Vuelo (FTAs) durante las actividades de planificación de la misión, teniendo en cuenta los posibles escenarios de fin de vuelo. Aunque en algunos casos esta información puede actualizarse durante el vuelo, esto limita la capacidad de los UAS para reaccionar ante sucesos imprevistos. Además, si se produce una situación imprevista y no se puede alcanzar una FTA planificada, se producirá un accidente incontrolado en el terreno. En la figura 1 se puede ver una representación gráfica del caso de uso.

El centro de vuelo dispone de una pista primaria de asfalto de 600 m de longitud y 18 m de anchura y una secundaria de hierba de 400 m de longitud y 15 m de anchura, con orientación 07-25 y elevación de 443,8 m (1456 pies). Está equipado con un radar primario para rastrear las operaciones de los vehículos aéreos no tripulados y detectar posibles intrusos, tripulados o no, en la zona de operaciones.

Las instalaciones de ATLAS están rodeadas por una TSA (*Temporary Segregated Area*) de 1.000 km², conocida como TSA30, que puede ser activada por el personal de ATLAS mediante la emisión de un NOTAM (*Notice To Air Missions*), y permite

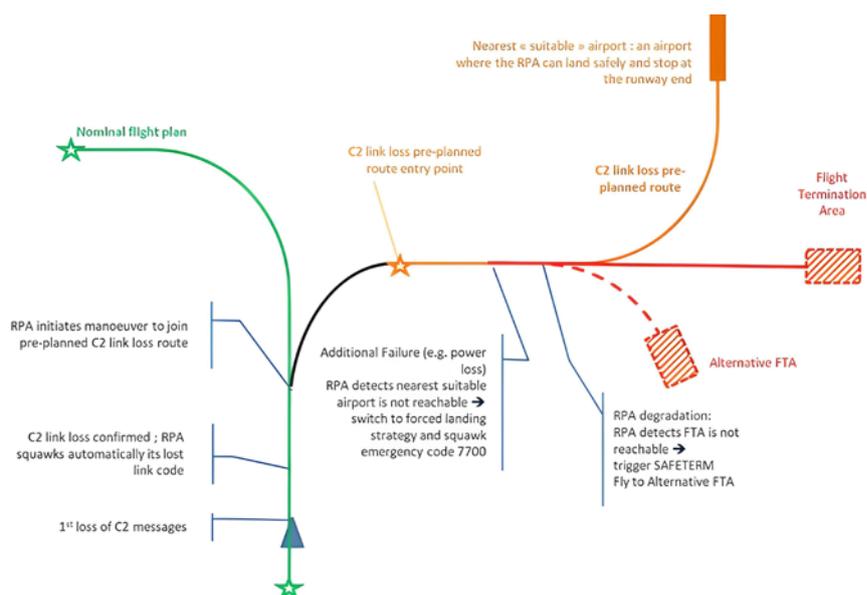


Figura 1. Representación gráfica del caso de uso. (Fuente: Consorcio GMV-AERTEC)

El SAFETERM pretende resolver esta carencia calculando áreas de terminación de vuelo alternativas basadas en la autonomía de vuelo y en el reconocimiento de patrones mediante inteligencia artificial, visión por computador y aprendizaje profundo.

Los vuelos de captación de datos y validación se realizaron en el Centro Atlas, en Villacarrillo, Jaén. Estas instalaciones han sido concebidas para la operación de sistemas RPA de clase I y II, de hasta 650 kg de MTOW.

realizar operaciones de vuelo más allá de la línea visual.

Esta será la zona de estudio, tanto para el entorno sintético como para las pruebas con datos captados por sensores embarcados en un RPAS. El SAFETERM podría desplegarse en escenarios semejantes y se podría esperar resultados equivalentes. Aquí entra una de las premisas fundamentales del proyecto, de cara a plantear que un sistema basado en inteligencia artificial se pueda certificar: ¿un dato de entrada produce de manera

determinista el mismo dato de salida? De ser demostrable, tendríamos solucionado uno de los primeros escollos respecto a la implementación de este tipo de sistemas en aviación, su certificabilidad. A este efecto, el SAFETERM se está usando como caso de uso por las autoridades de estandarización.

Además, si para una misma entrada obtenemos la misma salida, tendremos que adaptar el sistema, *off-line*, para que pueda responder adecuadamente a los distintos escenarios de despliegue. Un ejemplo claro sería: querer desplegar SAFETERM, entrenado en ATLAS, en Europa del norte. La distribución de los datos del entrenamiento inicial será diferente de los de despliegue y, por tanto, el sistema tendrá un funcionamiento limitado. La pregunta entonces sería: ¿qué coste tiene una adaptación del SAFETERM a un escenario cuyos datos captados tengan una distribución diferente del escenario de entrenamiento?

Desarrollo

Para solventar las preguntas anteriormente planteadas, el SAFETERM se ha desarrollado en cuatro etapas, todas supervisadas por *Stakeholders* de las diversas entidades implicadas

en la estandarización y potencial certificabilidad de los sistemas basados en inteligencia artificial a nivel europeo.

Tras el establecimiento del concepto de operaciones, se diseñó un entorno de entrenamiento y validación basado en bases de datos públicas, compuestas por imágenes satelitales en blanco y negro. Dichos datos sirvieron para dimensionar y estudiar las distintas arquitecturas punteras del momento y obtener modelos entrenados que permitieran comparar los distintos resultados. Este conjunto de datos se acercaba al caso de uso, pero la distribución de los datos no era la misma. Al ser imágenes satelitales tienen una vista cenital que no se correspondería con el caso de uso. En el caso del UAS, la cámara irá apuntando hacia adelante para poder tener información de contexto y hacer una previsión con cierto margen de maniobra de las zonas hábiles para aterrizar.

Por este motivo, a partir de la segunda fase, ya se planteó el uso de datos con una distribución equivalente a la que tendría un UAS desplegado, generados sintéticamente. Mediante herramientas de simulación de entornos sintéticos, se diseñó todo el entorno

alrededor de las instalaciones del centro de pruebas ATLAS, donde más tarde se obtendrían imágenes reales para la tercera fase y finalmente se volaría el UAS TARSIS 75, con el SAFETERM embarcado.

A lo largo de las distintas fases se realizaron pruebas para la optimización de los tiempos de ejecución, se desplegó el sistema sobre plataformas *hardware* certificables (arquitectura ARM) y se probaron plataformas *hardware* de aceleración (*Graphical Processing Units*).

Tal y como vemos en la figura 2, se presenta un resumen de las fases 2, 3 y 4, realizadas con el conjunto de datos con una distribución adecuada. En dichas fases, se trabajó tanto con imágenes RGB, como con imágenes FWIR.

En las distintas fases se utilizaron combinaciones de datos sintéticos y reales. En la primera, datos puramente sintéticos se emplearon para probar la arquitectura (fase 1 de validación). También se confrontó el algoritmo entrenado con datos puramente sintéticos, a un conjunto de datos real de validación, para cuantificar la capacidad de generalización.

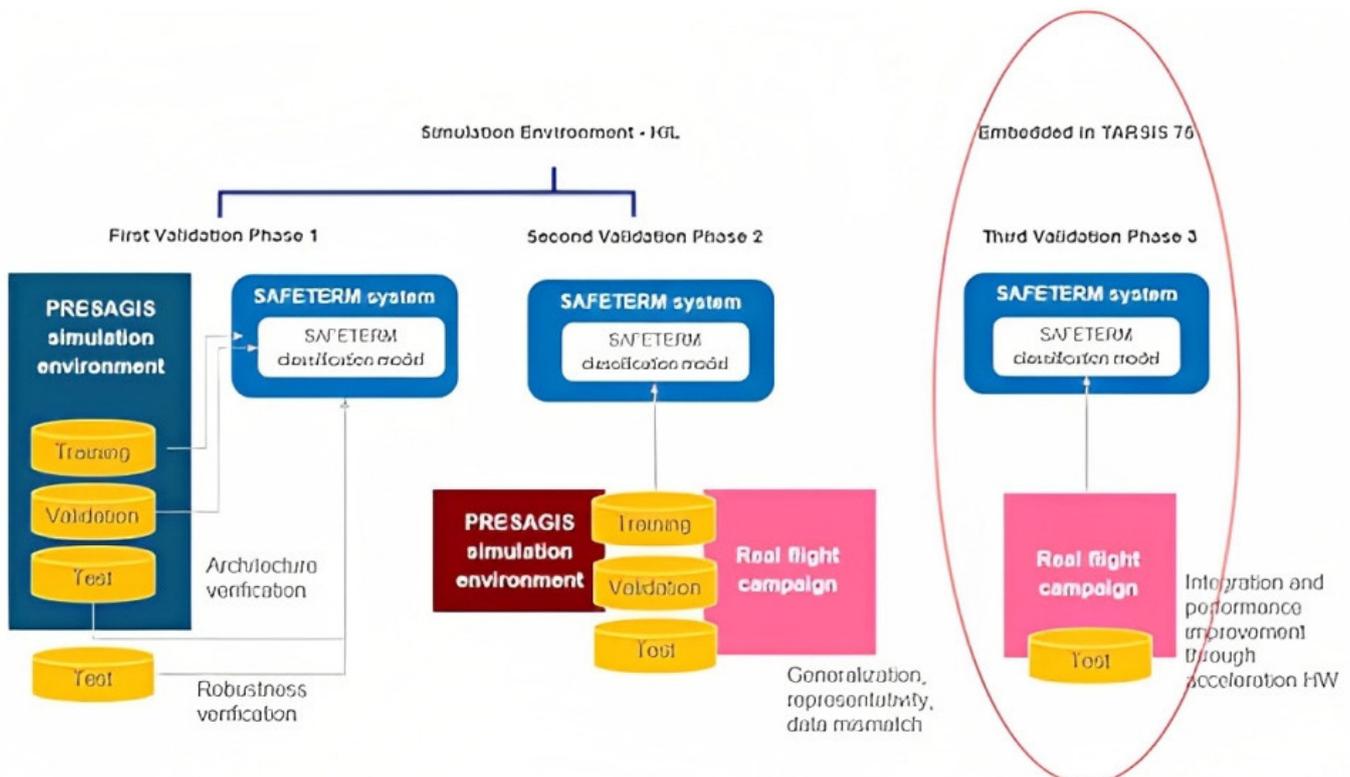


Figura 2. Fases de validación. (Fuente: Consorcio GMV-AERTEC)

En profundidad

Como cabría esperar, la representación sintética del centro ATLAS, aun siendo de calidad, dista en nivel de detalle a la representación real del entorno. Esto es entendible, puesto que, al hablar de imágenes, detalles tales como el nivel de luminosidad, los destellos del sol sobre la superficie del agua o la configuración de las sombras de los árboles son detalles trascendentales para el modelo y muy difíciles de plasmar adecuadamente, sin incurrir en costes que superasen con creces el alcance del proyecto.

En nuestro caso, a efectos de un entorno operacional real, se requiere una adaptación mediante la incorporación de datos reales, para terminar de encajar el modelo con la distribución de datos final.

Esta combinación de datos de distintas procedencias es interesante. Se utilizaron distintas técnicas de aprendizaje para tratar de mitigar esta singularidad: aprendizaje profundo –automático tradicional y de transferencia– y se ha estudiado el impacto

en la generalización y corrección de errores.

El aprendizaje automático requiere un entrenamiento desde cero, que es computacionalmente caro y requiere una gran cantidad de datos para lograr un alto rendimiento. Sin embargo, el aprendizaje por transferencia es un método automático en el que reutilizamos un modelo preentrenado como punto de partida para un modelo en una nueva tarea. Como ya se ha dejado intuir anteriormente,



Figura 3. Ejemplos y preproceso de datos reales vs sintéticos de paisaje urbano y masas de agua. (Fuente: Consorcio GMV-AERTEC)

SAFETERM es un sistema que, para ser operativo, debe ser fácilmente adaptable y con impacto y costes relativamente reducidos. La utilización del aprendizaje de transferencia y la evaluación de los resultados tuvo como finalidad ver hasta qué punto era posible optimizar los procesos de despliegue en distintos escenarios al de entrenamiento.

A través del aprendizaje de transferencia se pretendía adaptar el modelo preentrenado con datos sintéticos, mediante datos reales.

En el entrenamiento tradicional se emplearon distintas proporciones de datos reales y sintéticos. De esta forma, se contrastaron diferentes distribuciones con el fin de localizar las proporciones óptimas para el problema de SAFETERM. Esta sería la opción empleada para la versión final del sistema.

Para la cuarta fase (Fase de Validación 3), el sistema se integró y embarcó en el UAS TARSIS 75 de AERTEC. El modelo se optimizó para que el tiempo de ejecución de todo el algoritmo se ejecutase en tiempo real y se pudieran extraer predicciones con una frecuencia suficiente para que el piloto automático tuviera datos constantemente actualizados.

Una vez expuestos los pasos y la estructura, debemos destacar las tecnologías utilizadas y los objetivos con respecto a qué se pretendía validar con SAFETERM. En este proyecto se han parametrizado, entre otros, para el caso de uso particular, la influencia que tiene el origen de los datos (real *versus* sintético), el tipo de sensores y su efecto en los resultados. Esto se analizó, de cara a un despliegue real, para minimizar costes a la hora de conseguir datos de una zona de difícil acceso (al ser operaciones militares, no siempre vamos a poder hacer campañas de captación de datos).

Debemos resaltar que, para garantizar un aprendizaje adecuado, los conjuntos de datos para entrenamiento y validación deben estar desacoplados. Aun teniendo la misma distribución, una imagen no deberá estar presente en ambos conjuntos. Asimismo deberemos garantizar un cierto nivel de complejidad, variabilidad y

representatividad de estos: zonas geográficas reservadas, cambios orográficos, distintas horas del día o distintas condiciones meteorológicas han sido algunas de las medidas adoptadas.

En la figura 3 se pueden ver ejemplos de los distintos tipos de datos reales y sintéticos utilizados en el entrenamiento y validación del sistema. En estas imágenes se muestra un ejemplo de dato real y sintético de entorno urbano (par superior de imágenes), otro ejemplo de datos real y sintético de masas de agua (par intermedio) y una simulación de cómo se preprocesaría tanto una imagen real como una imagen sintética (último par inferior). Mientras que la zona de entorno urbano se trata de un entorno restringido para realizar un aterrizaje seguro, la zona de agua será la categoría deseable, por la baja probabilidad de obstáculos. Otras zonas evaluadas fueron terrenos llanos y boscosos, donde el UAS sí podría aterrizar, y zonas con carreteras, que también se deberían evitar para no causar daños a personas, vehículos o infraestructuras.

¿Cómo se han trabajado los datos? Para empezar, estas imágenes que vemos en la figura anterior han necesitado ser preprocesadas. El modelo final transforma las capturas del vídeo de tres canales a un único canal de luminancia, y, posteriormente, dichas imágenes se descomponen en pequeños recortes de 100x100 píxeles. Cada recorte corresponderá a una única categoría o clase: agua, llanura, bosque, carretera o urbano. Esto permite que las inferencias sean mucho más rápidas, puesto que el modelo tiene una entrada más pequeña y, por tanto, un número de operaciones más reducido. Asimismo, este peso reducido facilita el despliegue del modelo entrenado en plataformas *hardware* con reducida potencia computacional y memoria, acercándonos a unas especificaciones de SWaP realistas para embarcar el sistema en un UAS.

Tras categorizar cada imagen como una matriz de predicciones, se realiza una consolidación mediante algoritmia basada en reglas, para reconstruir un mapa de segmentación. Esta estrategia se pensó así por varios motivos: aislar al máximo las funciones de la inteligencia artificial, reducir el peso al máximo de dicha partición

y simplificar el tipo de operaciones. Por supuesto, hay herramientas hoy que pueden solventar este problema mucho mejor, pero debemos tener en cuenta que estamos trabajando en un entorno donde la potencia computacional es escasa y el volumen de datos en memoria no puede ser excesivo. Algunas alternativas, si el *hardware* certificable fuera más potente, serían: redes neuronales mucho más complejas que segmentan directamente las imágenes, pero requieren de datos etiquetados más complejos y mucha más capacidad de cómputo. También se podría preparar una capa de entrada que trabajase como una ventana flotante *convolucional*, pero Tensorflow para Microcontroladores no daba soporte a dicha funcionalidad. Posteriormente se darán más detalles acerca de esta herramienta.

Finalmente, tras entrenar y desplegar el modelo en el entorno de inferencia, se realizaron pruebas con vídeos grabados durante operaciones reales, dentro y fuera de la zona ATLAS y los expertos evaluaron la tasa de precisión del algoritmo a la hora de detectar zonas alternativas de terminación de vuelo.

En la figura 3 se ha planteado el entrenamiento y verificación por clases, donde cada dato de entrada tiene una única clasificación. Sin embargo, los recortes son parte de un todo más grande que, tras pasar por todo el proceso, debe ser reconstruido para componer una imagen del tamaño original, pero con más información: una imagen segmentada semánticamente. Para construir dicho mapa de clases, es necesario contextualizar los recortes y consolidar mediante un algoritmo basado en reglas, cuyo principal dato de entrada es el solape entre recortes, las predicciones realizadas por el algoritmo.

En la figura 4, un experto categoriza aquellas áreas que un piloto humano consideraría como hábiles. En verde, aquellas áreas disponibles para el aterrizaje seguro, por tipología y tamaño; en rojo, las que *a priori* no podrían ser utilizadas para la terminación del vuelo (carretera y entorno con edificios). En naranja pueden verse áreas consideradas como demasiado pequeñas para ser una zona segura. A la derecha, el resultado final que

En profundidad

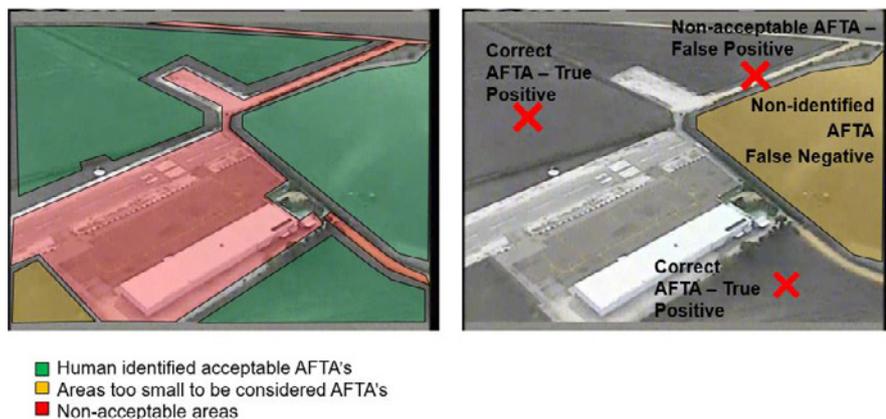


Figura 4. Ejemplo de consolidación de métricas. (Fuente: Consorcio GMV-AERTEC)

podría arrojar el sistema SAFETERM: una serie de puntos o centroides, cuyas coordenadas se establecen como los centros de las áreas disponibles para aterrizar.

Como vemos, hay un área que no fue detectada, pintada de naranja, aunque debería haberlo sido. Observamos dos «Verdaderos Positivos», donde el algoritmo provee dos puntos de aterrizaje aptos y que coinciden con el criterio del experto. Observamos también un punto sobre la carretera, que sería un «Falso Positivo», puesto que el experto ha categorizado dicha zona como no apta para la terminación del vuelo.

Finalmente, ni el área de la esquina inferior izquierda ni la zona con edificios han sido detectadas como zona segura y, por tanto, no se ha colocado un punto para el aterrizaje sobre ellas.

De aquí se pueden extraer diversas métricas: la ratio de verdaderos positivos respecto al total de zonas aptas para el aterrizaje, la ratio de falsos positivos, la ratio de áreas no detectadas o ratio de falsos negativos y la ratio de verdaderos negativos. La métrica crucial será la ratio de falsos positivos, puesto que podemos tardar más o menos en encontrar una zona apta para el aterrizaje, pero jamás debemos aterrizar en una zona prohibida, puesto que pondremos en riesgo a personas e infraestructuras.

Para asegurar la solidez del modelo, se realizó un elevado número de pruebas para parametrizar la capacidad de generalización, así como de ataques adversarios, con imágenes equivalentes a las anteriormente mostradas, pero con efectos de emborronamiento

o ruidos gaussianos, simulando una entrada defectuosa de las cámaras. Los resultados obtenidos fueron realmente interesantes, con tasas de falsos positivos reducidas.

Durante las pruebas en vuelo, los resultados también fueron remarcables. En la página web del proyecto^[1], el vídeo *SAFETERM-Real Flight Demonstration* muestra los resultados obtenidos en tiempo real durante un vuelo del RPAS TARSIS 75, con SAFETERM embarcado e integrado.

Unidad de aceleración del procesamiento y marco de ejecución

Uno de los principales aprendizajes de este proyecto, ha sido el despliegue de un modelo de inteligencia artificial, que tuviera que ser ejecutado en tiempo real, en un *hardware* embarcable. En las primeras iteraciones, se desplegó entero en una FPGA sin aceleración, donde se utilizó una arquitectura ARM tanto para la gestión del código basado en reglas y las funcionalidades I/O, como para el tratamiento de las imágenes. Se trabajó con las librerías de Tensorflow, Tensorflow Lite y Tensorflow para Microcontroladores, para que todo el sistema estuviera desplegado en C/C++.

Tras computar la capacidad de la CPU, se decidió probar otro tipo de *hardware*, dado que las métricas temporales no eran viables para un despliegue en tiempo real. Conociendo las dificultades de la certificación de hardware para aviónica, se decidió utilizar una GPU embarcable para medir la mejora en tiempos y resultados, que fueron notables. Siendo

viable, en términos de SWaP, embarcar un SAFETERM potenciado con GPU, los siguientes pasos, para GMV, serán estudiar la posibilidad de incorporar este tipo de algoritmos a FPGAs que puedan dar las prestaciones necesarias. Si se consigue demostrar que un *software* basado en inteligencia artificial es determinista, no adaptable durante inferencia y, por tanto, seguro, tendrá que ser desplegado en una plataforma capaz de realizar los cálculos necesarios, de forma igualmente determinista.

Conclusiones

Las conclusiones del proyecto SAFETERM pueden ser aplicadas a proyectos con alcances y objetivos similares, pero proyectos muy diferentes pueden requerir procedimientos distintos.

El objetivo principal de las campañas de validación era asegurar un cierto nivel de complejidad que hiciera que las lecciones aprendidas fueran fiables y escalables a aplicaciones más reales, dentro del mismo contexto.

En la medida de lo posible, los métodos se aproximaron a aquellos que se utilizarían en un entorno de aplicaciones certificables. Contestando a la primera pregunta expuesta en la introducción, se mantuvo un contacto estrecho con diversas organizaciones estandarizadoras, así como otras entidades como la EASA^{[2], [3]} que han participado en los talleres organizados para evaluar el avance del proyecto. Su intervención y monitorización aportó gran valor a la hora de aterrizar los conceptos y los procesos y avanzar hacia la definición de una «Inteligencia Artificial Determinista». SAFETERM, además, se ha utilizado como ejemplo de caso de uso para las primeras guías de estandarización desarrolladas por SAE/EUROCAE, para sistemas embarcados basados en inteligencia artificial, para aviación^{[4], [5], [6]}.

En general, se han obtenido unos resultados realmente prometedores, tanto en relación con la ejecución en tiempo real de algoritmos basados en inteligencia artificial, como en relación con la aplicación de esta tecnología en el campo aeronáutico. A este particular, el tiempo de ejecución con una superposición de los

recortes (ver figura 3) del 50 % (88 inferencias) alcanzó una tasa óptima de 1,5 segundos, que para un entorno de ejecución en tiempo real es muy aceptable. Esta tasa se consiguió al incorporar la GPU dentro del proceso. Sin GPU, la tasa era de nueve segundos al ejecutar SAFETERM sobre la FPGA, usando solamente CPU y con todo el sistema basado en Tensorflow para Microcontroladores. La diferenciación de capacidades de cómputo se vuelve más evidente al ejecutar

electroóptico, además, obtuvo mejores métricas generales que el flujo FWIR. Este fue un primer paso para contestar la segunda pregunta planteada en la introducción: mediante datos sintéticos y una pequeña proporción de datos reales, se podrá adaptar SAFETERM, con un coste relativamente reducido, a distintos tipos de escenarios operacionales.

El proyecto culminó con una campaña de vuelo, donde SAFETERM se inte-

En la figura 5, se puede ver la foto del sistema embarcado.

Con una tasa de falsos positivos muy cercana a cero, el sistema es lo suficientemente robusto como para ser considerado una prueba de concepto exitosa. Este tipo de sistemas tienen un gran potencial que, a futuro, GMV espera poder aprovechar iterando sobre esta base, mejorando las prestaciones, incorporando *hardware* más avanzado y



Figura 5. Fotografía SAFETERM embarcable. (Fuente: Consorcio GMV-AERTEC)

para un 80 % de superposición (256 inferencias) La GPU necesitó dos segundos, mientras que la combinación de FPGA con Tensorflow para Microcontroladores necesitó casi nueve minutos para completar el análisis de una imagen completa. Es evidente, por tanto, que se necesita una FPGA, certificable, que soporte la paralelización de procesos para obtener tiempos razonables.

Otro tipo de métricas reseñables fue la proporción óptima de datos sintéticos y reales para la fase de entrenamiento. En el flujo electroóptico, se necesitó un 25 % de datos reales para captar una distribución de los datos cercana a la realidad, mientras que con el flujo FWIR solamente se requería el 15 %. Aumentar el número de datos reales no incrementa la precisión del sistema. El flujo

gró en el TARSIS 75 de AERTEC, y se realizaron diversos vuelos a lo largo de una semana en las dependencias del centro ATLAS, para demostrar las capacidades del sistema embarcado al procesar, mediante inteligencia artificial, un flujo de vídeo continuo y analizar el entorno, segmentando en tiempo real las imágenes captadas. Se llevaron a cabo dos planes de vuelo diferenciados. Uno, en el que UAS iba directo hacia una ciudad y, al activarse las condiciones de inicio, se obtuvieron áreas alternativas de aterrizaje fuera del entorno urbano. El segundo plan de vuelo permitió ver cómo el sistema inicialmente detectaba áreas de aterrizaje en zonas boscosas y de campo despejado y, al aparecer masas de agua, les daba ponderaciones mayores, superponiéndose en tiempo real a las anteriores y actualizando el estado para encontrar la zona más segura.

ampliando su espacio de aplicación, para desplegarlo en nuevos escenarios tácticos y para distintos tipos de misiones.

Bibliografía

- [1] Consorcio GMV-AERTEC. Página web oficial del proyecto.
- [2] EASA. (2020). EASA. *Artificial Intelligence Roadmap 1.0* EASA.
- [3] EASA. (2021). EASA. *Concept Paper: First Usable Guidance for Level 1 Machine Learning Applications*.
- [4] EUROCAE. (2019). ED-253. *OSD for Automation and Emergency Recovery*.
- [5] EUROCAE. (2020). ED-281. *MASPS for RPAS Automation and Emergency Recovery*.
- [6] EUROCAE. (2021). ER-022 *Artificial Intelligence in Aeronautical Safety-Related Systems Statement of Concerns*.