Design of a counter-drone visualisation platform incorporating sensor-data fusion

Stylianos Lagaras, Danilo Amendola, David Anderson

European Commission, Joint Research Centre (JRC), Geel, Belgium
Email: stelios.lagaras@vito.be, {danilo.amendola; david.anderson}@ec.europa.eu

***Abstract*—This paper describes the development of a counter-unmanned aircraft system (C-UAS) platform for sensor-data fusion and visualisation. The paper outlines the design, implementation and integration of sensor inputs for the effective detection, identification and tracking (DTI) of both cooperative and non-cooperative unmanned aircraft, commonly known as drones, in airspace. With a focus on the direct remote identification sensing, the platform architecture is based on state-of-the-art web platform technologies to create a modern prototype. The objective is to develop a prototype solution that can be extended to deploy and test more sophisticated sensor-data fusion algorithms on real-time data for C-UAS. A first version of a sensor-data fusion algorithm is implemented for data from direct remote identification (DRI) sensors using dynamic time warping (DTW). The Python implementation of DTW is discussed, highlighting its efficiency and applicability in real-time operations. A frontend interface of the platform is provided, offering a user-friendly graphical user interface for monitoring and controlling DTI drone activity. Providing a full-stack solution utilizing open-source tools for data handling, processing, and visualisation. The frontend application incorporates the visualisation of the official geo-zones. Future developments are discussed, such as implementing geo-zoning and geo-fencing alerts.**

***Index Terms*—Counter-Unmanned Aircraft System (C-UAS), Unmanned Aerial Vehicle (UAV), Unmanned Aircraft System (UAS), drones, counter-drones critical infrastructure, counter-terrorism**

I. Introduction

ORIGINALLY developed for military purposes, drones are now frequently used for disaster response, infrastructure inspection, surveillance, remote sensing, photography, recreation and more. The number of drones in use in the EU is set to grow significantly in the coming years, and they will improve greatly in terms of speed, agility, maximum range, payload capabilities, precision of sensors, and use of artificial intelligence. These developments will lead to a wider range of legitimate and lawful uses for drones.

However, drones can be used for illegal and dangerous activities such as smuggling, aviation harassment, public space harassment, cyber-attacks and terrorist attacks. In order to realise the potential of the lawful use of drones, it is necessary to address the potential threat that can be posed by non-cooperative drones. A non-cooperative drone can be classified according to the nature of non-cooperation that could include criminal, illegal (intended regulatory breach) or amateur (ignorant, careless). In recent years, plans have been uncovered involving the use of drones for terrorist attacks, and reports of malicious use of drones appear regularly in the press* . There have also been sightings of suspicious drones around critical infrastructure, such as energy facilities, airports, and ports, indicating the potential misuse of drones for hostile information gathering. Drones are used by criminals engaged in smuggling across borders, or to facilitate other illicit operations, including drug trafficking and smuggling contraband into prisons (see Figure 1b)

(a) (b)

Fig. 1: (a) A British radio-controlled aircraft from the First World War, which was the first drone to fly under control in March 1917; (b) Image of a drone for smuggling contraband into a prison in 2023.

In this paper, we adopt the term unmanned aircraft system (UAS) which includes the unmanned aircraft (UA), a ground controlling system with an operator, and the communication channel. This term was adopted by the EU Delegated Regulation 2019/945 [1] and USA Department of Defence [2], and it is commonly used in the technical literature and adopted in the European Single European Sky ATM Research project [3].

Following the Russian invasion of Ukraine, the world has also witnessed the growing importance of UAS, which are

* Corresponding author: Danilo.Amendola@ec.europa.eu

ISBN 978-3-903176-68-3 © 2024 IFIP

*Examples include: (i) the plan of an inspired jihadist who was sentenced by a Spanish court in October 2022 for planning to attack a stadium during a major football match using a drone packed with explosives (Daily Mail, 2022); and (ii) a failed attempt to assassinate the Venezuelan President using explosives-laden drones (BBC, 2018).

often used in an unorthodox way, e.g. by attaching explosives, used as "kamikaze drones" or as surveillance tools. The potential of UAS to operate undetected, to be multi-functional and cost-efficient has rendered them critical weapons in multiple conflict areas. History has shown that innovations on the battlefield are subsequently translated to the civil context by criminals or terrorists. For this reason, there is currently heightened concern about the security of critical infrastructures, public spaces and borders, in terms of their vulnerability to novel attacks involving drones, and the pressing need to develop counter-UAS capability.

II. Related work

While Counter-Unmanned Aircraft System (C-UAS) research and commercialisation are rapidly expanding [4]–[10], current state of the art lies its foundations on [11], which has been of great use in the aviation industry to detection, tracking, and identification (DTI) multiple aircraft. Furthermore, such principles must be extended with new requirements related to Unmanned Aircraft System (UAS) vulnerabilities and their potential intrusion strategies [12] that are key for their DTI [13].

The European Commission's Joint Research Centre (JRC) is developing C-UAS knowledge and capability via implementation of its DRONE project [14] and has published two handbooks on physical hardening [15] and protection of critical infrastructure and public spaces [16]. Moreover, the JRC runs a framework contract for C-UAS services for research and protection of the infrastructure of European Commission in both Geel (Belgium) and Brussels and to establish a Living lab[†] in Geel.

The EU-funded COURAGEOUS project (Grant Agreement nr. 101034655), launched in 2021, focuses on developing a standardised test methodology to DTI nefarious drones using countermeasure systems to protect the low-altitude airspace. The project aims to create a fair qualitative and quantitative comparison between different C-UAS by establishing a series of standard user-defined scenarios representing a wide range of use cases. By addressing the need for a standardised test methodology, the COURAGEOUS project significantly contributes to advancing the field and ensuring effective protection of low-altitude airspace from unauthorised drone usage. In [9] the authors provide an overview of the trends for the design and the implementation of C-UAS solutions. Radiofrequency (RF), Radio Detection and Ranging (RADAR), and visible-light sensors are preferred, whereas acoustic, Light Amplification by Stimulated Emission of Radiation (LASER)-based, and infra-red (IR) ones are less common. While RF and RADAR are more likely to be used standalone, some solutions prefer the combination of RADAR, visible-light, and thermal sensors.

[†]Living labs are a modern way of creating user-centred environments that enable innovation, co-creation and start-up development. https://joint-research-centre.ec.europa.eu/living-labs-jrc_en

A comprehensive overview of cooperative scenarios, in which a C-UAS implementation is augmented with the use of UASs for patrolling, can be found in [17]. This contribution also addresses neutralisation techniques to mitigate airspace breaches. A crucial aspect for this extended system is a robust wireless network that is able to not interfere with the counter system and is able to defend itself from external attacks. The challenges in these aspects can be found in [13], [18], [19]. More generally, in [20] Laoudias et al. provide a review on network localisation, tracking, and navigation, in particular on cellular localisation systems and solutions based on wireless local area networks.

A. Scope of the project

Direct remote identification (DRI), also known as Remote Identification (RemoteID) of drones, is the ability of a drone in flight to provide identification and location information that can be received by other parties through a broadcast signal. To meet the European Commission Delegated Regulation 2019/945 [1] and the Commission Implementing Regulation 2019/947 [21], ASD-STAN has developed the prEN 4709-002 Direct Remote Identification specification [22]. Technical specifications exist also for USA (ASTM F3411 [23]), and for Japan.

Although DRI covers many of the simple, careless incidents, and most commercial drones will carry the DRI, a higher level of safety is needed for public spaces and critical infrastructures, where we consider malicious attackers performing more sophisticated techniques. For example, signal spoofing is a technique for mimicking a signal but with 'synthetic' data, e.g. a DRI packets to flood the receiver system with hundreds or thousands of ghost targets. JRC has already demonstrated the potential for a low-cost "spoofer" module that creates a huge number of false signals and can be used to overwhelm a C-UAS system and/or mask a real attack. For this reason, different kinds of sensor technologies are necessary to contribute to an effective C-UAS solution that can maximise true drones detections and minimise false alarms.

Counter-UAS solutions can utilise a combination of sensors, radars, cameras, and software algorithms to detect and track unauthorised or malicious drones in the airspace. These systems can identify the drone's location, trajectory, and other relevant information (e.g. payload type), allowing authorities to respond appropriately. In C-UAS it is important to distinguish two categories, cooperative and non-cooperative drones:

- cooperative UAS actively communicate their intentions and collaborate with infrastructure, unmanned air traffic control, and authorities;
- non-cooperative UAS do not actively contribute to their identification.

It is of crucial importance for C-UAS solutions to be able to manage both categories of UAS. The C-UAS solution can detect the cooperative and non-cooperative UAS by exploiting two distinct radio frequency (RF) communication channels, i.e. DRI broadcast messages and Data-Link channel. A C-UAS solution needs to be able to process and merge the data coming from many different RF sensors, and to avoid having false (or duplicate) detections of UAS. This means that it is necessary to have a sensor-data fusion module with the capability to process and fuse the data. Many techniques can be applied to solve the sensor-data fusion problem, e.g. deterministic algorithms or AI techniques. In this work, we will make a first version using the dynamic time warping algorithm (described later in section III).

The addition of multiple sensor channels inevitably introduces complexity as we receive observations referring to the same object from multiple sensors, and also from the same sensor we may not have a direct correlation for the punctual observations. Here arises one of the main challenges, to find a proper way to correctly fuse these data. This is not a trivial task, and a lot of experimentation is needed to reveal what works and what not, and to resolve issues like channel synchronisation, new position (output of the fusion) approximation, choice of appropriate algorithm and much more.

In this project, we started from scratch, hence before working on the sensor data fusion algorithms we had to create the overall infrastructure to collect, manage and visualise sensor data. We needed to create a full-stack architecture, gathering the data from the sensor containing the target observation, through a process to fuse, transform and enhance the target observations, and create a graphical user interface (GUI), i.e. a web application, to visualise the information on an interactive map (see Figure 2).

This paper describes the tools and the architecture of the resulting C-UAS platform, including design decision. A more detailed technical report related to this work has been published on the EU Publication Office website [24]. The solution allows for sensor-data fusion using the dynamic time warping (DTW) algorithm, and we present some results regarding execution time of the algorithm. The process of rendering the data on the map, the challenge of serving the data on an intranet environment and the containerisation of the application services using docker are also described.

B. Direct remote identification through passive radio frequency sensor

For the creation of this application, a range of different tools and technologies need to be orchestrated and integrated. Starting from the package signal, we use a commercial sensor to detect the DRI messages corresponding to cooperative drones. Different types of radio frequency (RF) sensors for C-UAS are available on the market; some RF sensors provide functionalities to track non-cooperative drones by decoding the data-link channel between UA and remote controller (RC). This kind of solution only works with specific manufacture and models. We used a sensor that is able to capture messages for Air Traffic Management (ATM) and Unmanned Traffic Management (UTM). It captures the messages and makes use of a messaging system (i.e., AMQP 0.9 or AMQP 1.0 protocols) to disseminate the information to the backend. In our solution, we make us of RabbitMQ message broker to transmit the sensor data to the backend.

III. C-UAS Sensor data Fusion

Sensor fusion involves integrating data from various sensors with the aim of generating situational awareness that is more reliable and precise than what could be achieved using each sensor separately. Sensor fusion can lead to a reduction in measurement uncertainty, and can sometimes generate new insight, e.g. depth perception from two-dimensional images when fused to form a stereoscopic image. The sources used in sensor fusion can include a mix of different types of sensors, knowledge about the sensors' accuracy, historical data, as well as external knowledge and human inputs. In C-UAS, there are different types of sensors, such as radar, acoustic sensors, electro-optical/infrared sensors and RF detectors, each with pros and cons. Adding sensors to a C-UAS system should be done judiciously, in order to ensure that the increasing complexity of the system is compensated for by an increase in performance. In its simplest form, sensor fusion can offer a universal picture of non-overlapping sensors, e.g. four radars covering a broad area in which the individual observations are merged to improve the overall monitoring process. There are several algorithms and techniques for implementing sensor fusion e.g. Kalman filter, neural network convolution, and Gaussian processes. In this work, we describe a first step in designing a sensor-fusion process by comparing the distance between multiple trajectories of points in 3D space, in order to judge if the trajectories should be fused or not. To this end, we use dynamic time warping (DTW), which is an algorithm that can generate distance measures between sequential time series, even if the data points are not aligned or synchronised in the time domain.

A. Dynamic Time Warping

Dynamic time warping (DTW) is an algorithm for comparing two time series [25]. It produces two pieces of information: i) a mapping of which points along the two different times series correspond with each other, and ii) a measure of the similarity between the two time-series. The main characteristic of the DTW is that it can compare time series even if they are slightly delayed or stretched in the time domain. DTW was originally developed for speech recognition applications, and it has since been applied to temporal sequences of video, audio, and graphics data.

The DTW algorithm is normally implemented with the following rules:

- every point from the first series must be matched with one or more from the second series, and vice versa;
- the first point from the first series must be matched with the first point from the second series (but it does not have to be its only match);
- the last point from the first series must be matched with the last point from the second series (but it does not have to be its only match);

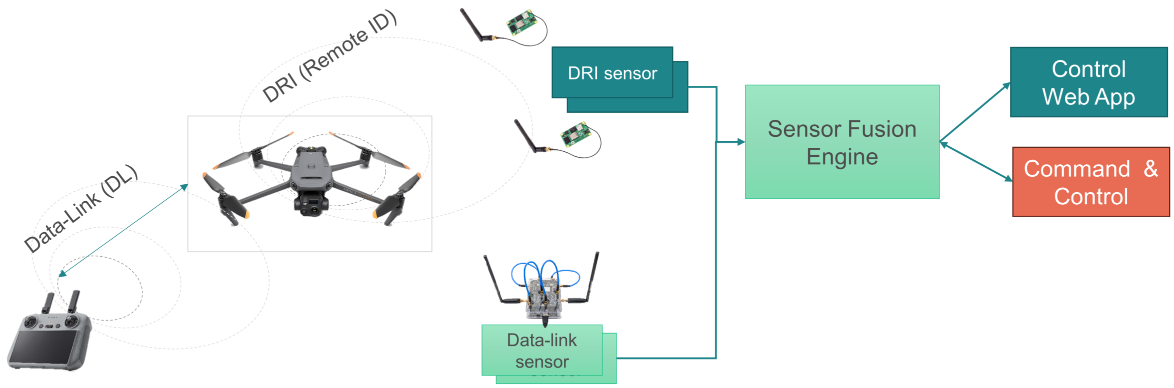


Fig. 2: Architecture of the solution.


```
{
    "vehicleID": {
        "serialNumber": "1581E276CH4MR0A050W",
        "vehicleType": 2
    },
    "timestamp": "2020-06-17T08:01:01.591Z",
    "operatorID": "ABC548411-123",
    "vehicleLocation": {
        "latitude": 50.699642,
        "longitude": 15.273084,
        "accuracy": 10
    },
    "pilotLocation": {
        "latitude": 50.699642,
        "longitude": 15.273084,
        "type": 0
    },
    "altitudes": {
        "geodeticAltitude": 110.5
    },
    "trackDirection": 172,
    "groundSpeed": 2.699,
    "trackID": 5326,
    "fusion": {
        "type": 2
        "likelihood": 100
    }
}
```


Fig. 3: Payload message of DRI sensor.

- the mapping of the points from the first series to the second series must be monotonically increasing, and vice versa.

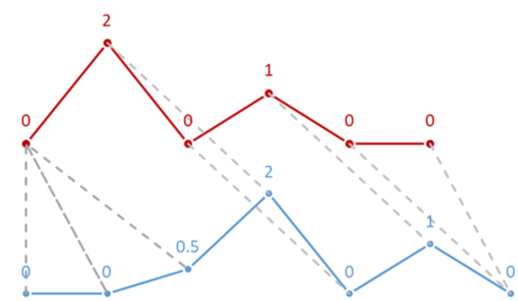
For two time series $x(i)$ and $y(j)$, with i in the range $(1, N)$ and j in the range $(1, M)$, the distance matrix is determined as follows:

1) a N x M distance matrix is initiated;
2) iterating through $i = (1, N)$ and $j = (1, M)$, the values of the distance matrix are calculated as the distance, d, between the points $d(x_i, y_j)$ plus the minimum of

$$[d(x_i - 1, y_(j - 1)), d(x_(i - 1), y_j), d(x_i, y_(j - 1))]$$

, which correspond to the matching, insertion, or deletion of points, respectively;
3) the mapping of points from one time series to another is obtained by starting from the upper-right corner of the matrix, $[N, M]$ and traversing back to $[0, 0]$ via the lowest of three neighbouring cells: $[i-1, j]$, $[i-1, j-1]$, or $[i, j-1]$;
4) the distance measure is calculated by normalising the sum of the distances along the path by the number of points on the path.

The concept is illustrated in Figure 4. In this example, the mapping path is $[0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0, 0]$, and the normalised distance metric is therefore $3/8 = 0.375$.



(a)

6	3	3	3.5	5	1.5	2.5	0.5
5	3	3	3	4.5	1.5	1.5	0.5
4	3	3	2.5	3	1.5	0.5	1.5
3	2	2	2	2.5	0.5	1.5	1.5
2	2	2	1.5	0.5	2.5	3.5	3.5
1	0	0	0.5	2.5	2.5	3.5	3.5
	1	2	3	4	5	6	7

(b)

Fig. 4: Illustration of DTW algorithm concept. Given two time series (a), the DTW algorithm provides a distance matrix (b), which in used to generate the mapping between points and the distance metric.

DTW Python Package	Mean Execution Time	Standard Dev.
DTAIDistance distance_fast	0.0097	0.0008
TSLearn	0.0164	0.0012
PyTS	0.0421	0.0043
DTW-Python	0.0879	0.0064
FastDTW	0.6758	0.0349
Dtaidistance	3.7540	0.1216

TABLE I: Comparison between different DTW Python libraries, results with 20 runs and time series length of 1,000 and 2,000. Source: Forecastegy.com, 2024

B. Fusing 3D trajectories using DTW

In this work, we extended the DTW algorithm to three dimensions (latitude, longitude, height). The DTW distance metric was used to determine if trajectories of sensor data points were separate trajectories of multiple drones (i.e. no fusion applied), or duplicate trajectories of a single drone (i.e. sensor data should be fused). A three-phase approach was implemented:

- the original distance table is calculated which corresponds to the output of DTW for each pair of time series (i.e. 3D trajectories);
- transform the non-measurement distance unit to a similarity value where similarity $\in [0, 1]$ with 1 means exact match;
- apply a Boolean cut-off value similarity, for the example 0.9, to determine which of the time series should be fused.

When this process finishes, we are ready to merge the data series by traversing the Boolean matrix. We traverse it row by row, and we fuse the tracks by calculating their mean values. The new track object, with various useful information like which sensor data series were fused, a new unique ID, possible alarms etc. are assigned in the new drone model. The new data are stored in a special stream named fused_stream.

An example of the 3D trajectory fusion is shown in Figure 5. On the left, we see the distance matrix in which each trajectory is compared to each other. Along the diagonal, the values are zero because the trajectories are identical. We can see that the green and yellow trajectories (dp1 & dp2) are close and have a distance of 0.249. In the middle image, the distance matrix is converted to a similarity matrix. On the right, we see the thresholded matrix, and in the graph underneath the green and yellow trajectories have been fused into a new trajectory, shown in purple.

C. Python implementation of DTW

When studying the application of DTW to the fusion of 3D trajectories, we examined the calculation time of different DTW libraries in Python. The results of the different libraries are summarised in Table I, in terms of mean execution time and standard deviation.

Based on these considerations, we decided to use the $DTAIDistance$ package [26] for our implementation of dynamic time warping. The $DTAIDistance$ package offers several benefits, including fast C implementation with parallel mode, the ability to apply the DTW algorithm to multi-dimensional time series, a distance-to-similarity matrix conversion, and agglomerative clustering with hierarchical tree plot output.

Our algorithm reads data points received in sensor streams, and for each sensor it executes a query on the time series to get the three relevant values for the specific sensor. The length of the time series read can be configured and in our experiment we used $DATAPOINTS\_NUMBER = 10$. Lastly, we create the three-dimensional location vectors for each sensor and we normalise them using the sklearn.preprocessing package (a submodule of scikit-learn). Subsequently, we have a three-dimensional matrix: i) number of drones, ii), $DATAPOINTS\_NUMBER$, iii) location vector.

From Table II, we can see that in a common scenario of less than 100 drones and a time series length of 20 with three dimensions, the execution time of the fusion is acceptable for real-time operation. There are also more operations taking place to prepare the data fusion, such as normalisation of the space and recreation of the tracks – which involves reading from the database.

The data fusion is a CPU-bounded process, and a critical service for our application, thus we need to exhaust every possibility for optimising the execution time of the algorithm. As it is known for developers, Python by design does not allow parallelisation between different threads due to the Global Interpreter Lock. So, if we want to take advantage of more CPU resources (cores) we need to run our fusion algorithm through a separate process using the Process class from the multiprocessing package. The drawback is that we need to handle this process inside our main script $receiver.py$ and use inter-process communication for triggering the fusion to start running. The solution to that, was to use the Event class and set the event when we start receiving some data points.

IV. Backend solution/architecture

An example of the setup used in our test with real sensors is shown in Figure 2. We have two DRI sensors, which are internally configured to publish the observations to a dedicated exchange broker that can routes all the messages to one or more destination queues. In our implementation, the message broker RabbitMQ exchange is named $sensors\_exchange$ and collects all the DRI sensor messages. The $sensors\_exchange$ can route to multiple queues, for instance, to better scale up. In our implementation, one queue is enough and this queue is named DRI_ALL. Finally the workflow is concluded with the consumer module that reads the messages from the queue. For our setup, we use an exchange with a specific type of routing to forward the messages to different queues according to their sensor name. We also make use of the convenient way to initialise the configuration, exchange types and queues of our RabbitMQ instance from the $rabbit.definitions.json$ file found inside the rabbitmq-docker folder.

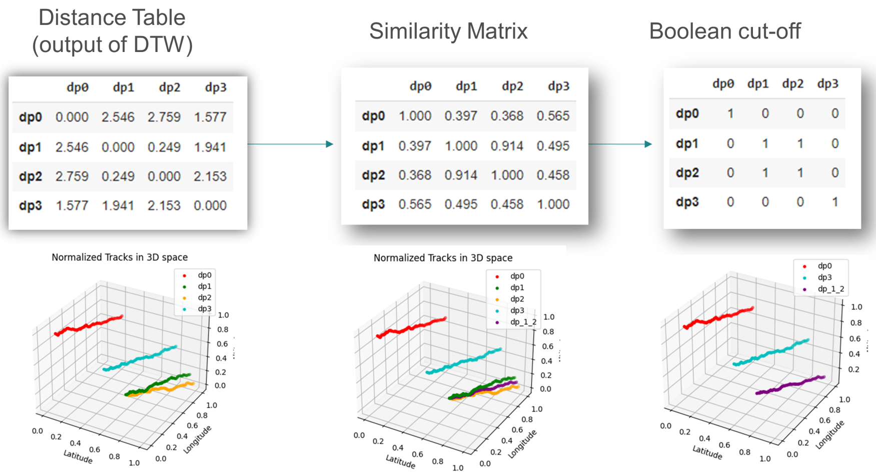


Fig. 5: Example of sensor data fusion of 3D trajectories using dynamic time warping

Number of time series	Length of time series	Dimensions	Average time for 100 iterations	Standard Deviation
1000	100	3	23.80 s	0.0809
100	100	3	0.2634 s	0.0401
100	[50-100] (1)	3	0.1681	0.1586
100	20	3	0.0144 s	0.0046

TABLE II: Investigation of influence of time series length on calculation time for $DTAIDistance$ fast distance and similarity matrix. The tests are conducted to determine the scalability of the DTW libraries and algorithms in Python using the following workstation: CPU Xeon E3-1200 v5/E3-1500 v5/6th Gen Core Processor Host.
(1) Non-homogenous length

A. Data processing backend

In this section, the backend application and its modules are described. As introduced in previous sections, the objective was to design a modular and scalable platform. Hence our backend implements a set of APIs, designed with the OpenAPI standard, to better isolate the presentation from the processing. The decision to implement the backend application in Python is mostly due to the fact that Python is a well-known language in the scientific community, and due to the large availability of libraries and modules. Due to previous experience in Python and Node JS, a decision was taken to use Python and the modern FastAPI web framework to prototype the APIs.

Consequently, we chose Pika as a Python for RabbitMQ message broker with a traditional blocking connection. Pika is a pure-Python implementation of the AMQP 0.9-1 protocol that tries to stay fairly independent of the underlying network support library. A blocking connection means that the thread that handles the communication will be blocked until it receives a new message from the queue. This has the trade-off from one side of simplicity, easier understanding and debugging of the application lifecycle, and on the other side the need to handle the concurrency of the threads to avoid unwanted blocking situations.

One of the main ideas for future experimentation would be to try asynchronous/asyncio type of Pika adaptors, with the potential benefits of using fewer threads due to the non-blocking behaviour of concurrency. It can be expected that the more sensors/channels used in the system, the more time improvement can be obtained via asynchronous programming compared to blocking connections.

B. Synthetic data generators

In our first steps to reproduce synthetic sensor data broadcasted from two or more sensors, we created a synthetic data generator comprising two different scripts that randomly change the next positions of data points simulating two sensors. Later, to experiment also with synchronisation, we used our RabbitMQ instance together with a controller script to orchestrate the generators so they send together their packages synchronised with controller's period. Finally, to decouple it from our source file, a separate folder, .generators, was created in the backend directory that can be used as a standalone package for creating synthetic data.

V. Frontend application

In counter-UAS, one of the most relevant activities is to command and control the operation and activities. A command and control (C2) system is designed to counter the threat posed by UAS, and it plays a crucial role in managing and coordinating C-UAS operations. It ensures effective communication, decision making, and coordination among various elements within the C-UAS architecture. In this project, it was not the intention to design a full C2 system, but rather to focus on designing a frontend application that is easy-to-use, customisable, and accessible from mobile and web.

The framework chosen to implement the frontend application was Nuxt, which is a JavaScript framework for building web applications based on Vue. The version 3 of Nuxt offers a useful layer of abstraction for Vue developers, simplifying the use of concepts like routing and auto-imports of components. In this work, we used Nuxt with the Vue-Leaflet package6F and the Nuxt UI library7F for the graphical components like cards and buttons. In Nuxt UI, cards are versatile elements that can contain various types of content such as text, images, links, buttons, and other components. The map library, Leaflet, offers interactive features to help the user moving in the virtual environment, for example:

- auto-focus on specific target by clicking in a drone card;
- lock the view following a specific drone;
- highlighting the path for a specific drones;
- enable/disable the stream connection from the server;
- enable/disable multi-sensors fused data visualisation.

VI. Future developments

Geo-zone or geo-fence commonly refer to a 2D virtual perimeter or boundary around a geographic area. Geo-zones are used in various applications to trigger specific actions when a person or object enters or exits the defined area, e.g. transportation applications like taxi. A geo-zone in the U-Space refers to a 3D geographical area defined by specific coordinates where U-Space services and regulations are applied. Geo-zones are portions of airspace where UAS operations are facilitated, restricted or excluded. These geo-zones can be used for various purposes, such as defining areas for drone operations, managing air traffic, or implementing safety measures. By creating geo-zones, U-Space aims to ensure the safe and efficient integration of UAS into the airspace.

In our solution, we have integrated the visualisation of official geo-zones for Belgium to help the users and facilitate the C-UAS operations. The Belgian geo-zones are collected from the official service Droneguide from Skeyes Belgian air traffic controller. The data are downloaded from the official map in GeoJSON format.

Adding the geo-fencing functionality to have automatic alerts for unauthorised UAS entering in a specific geo-zone is left for future development. However, we already designed a partial solution by adding a specific geolocation and making use of Tile38 geo-fencing tool. The main idea is that all the cooperative UAS are broadcasting the DRI messages, the location of each UASs is cross-checked with the geo-server, and if a drone cross a restricted area an alarm is shown on the GUI. Geo-fencing can simplify and speed up the phase of intervention to mitigate the risk coming from drone in restricted areas.

VII. Conclusion

This project has successfully established a comprehensive full-stack counter-UAS sensor-data fusion and visualisation platform that can integrate multiple sensor inputs for effective drone monitoring within the European Union's airspace. The platform is designed to address both cooperative and non-cooperative drones, given the rapid proliferation of drone technology and its varied applications. Through the implementation of advanced technologies such as message brokers, a Python data processing backend, and Redis in-memory database, the platform has significant potential as a foundational system for real-time data analysis and sensor fusion.

A key innovation within the project is the designing of a trajectory fusion algorithm with the application of the dynamic time warping (DTW) algorithm to fuse 3D trajectories of sensor data, which provides robust alignment of time series that are not synchronised temporally. The Python implementation of the DTW algorithm has proven to be efficient for real-time operations and forms the basis for future exploration and development of more complex sensor fusion algorithms.

The frontend interface, characterised by its user-friendliness, enables operators to monitor and control drone activity effectively. It represents an accessible and scalable solution that can adapt to various devices, enhancing the operational capabilities of those tasked with airspace security.

In conclusion, the project has laid a solid groundwork for future advancements in C-UAS technology. With the successful integration of official Belgian geo-zones within the frontend application, the platform is well-positioned to incorporate geo-zoning and geo-fencing alerts (a backend module would be implemented for that), thereby elevating its functionality. As drone technology continues to evolve, so too will the challenges of monitoring and securing airspace. The platform developed in this project stands as a versatile tool, ready to meet these challenges through continued innovation and integration of emerging technologies.

References

[1] "Delegated regulation - 2019/945 - EN - EUR-Lex," doc ID: 32019R0945 Doc Sector: 3 Doc Title: Commission Delegated Regulation (EU) 2019/945 of 12 March 2019 on unmanned aircraft systems and on third-country operators of unmanned aircraft systems Doc Type: R Usr_lan: en. [Online]. Available: https://eur-lex.europa.eu/eli/reg_del/2019/945/oj

[2] United States. Department of Defense, "Unmanned Aircraft Systems Roadmap 2005-2030," Tech. Rep., Jan. 2005. [Online]. Available: https://rosap.ntl.bts.gov/view/dot/18248

[3] Single European Sky ATM Research 3 Joint Undertaking (EU body or agency), *European ATM master plan: the roadmap for delivering high performing aviation for Europe : executive view : edition 2015*. Publications Office of the European Union, 2015. [Online]. Available: https://data.europa.eu/doi/10.2829/240873

[4] D. Rauschen, D. Gläsel, H. P. Such, P. Zimmermann, and M. Antweiler, "Commercial of the shelf counter uav," in *2019 International Conference on Military Communications and Information Systems (ICMCIS)*, 2019, pp. 1–5.

[5] J. Wang, Y. Liu, and H. Song, "Counter-unmanned aircraft system(s) (c-uas): State of the art, challenges, and future trends," *IEEE Aerospace and Electronic Systems Magazine*, vol. 36, no. 3, pp. 4–29, 2021.

[6] S. Park, H. T. Kim, S. Lee, H. Joo, and H. Kim, "Survey on anti-drone systems: Components, designs, and challenges," *IEEE Access*, vol. 9, pp. 42 635–42 659, 2021.

[7] G. Lykou, D. Moustakas, and D. Gritzalis, "Defending airports from uas: A survey on cyber-attacks and counter-drone sensing technologies," *Sensors*, vol. 20, no. 12, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/12/3537

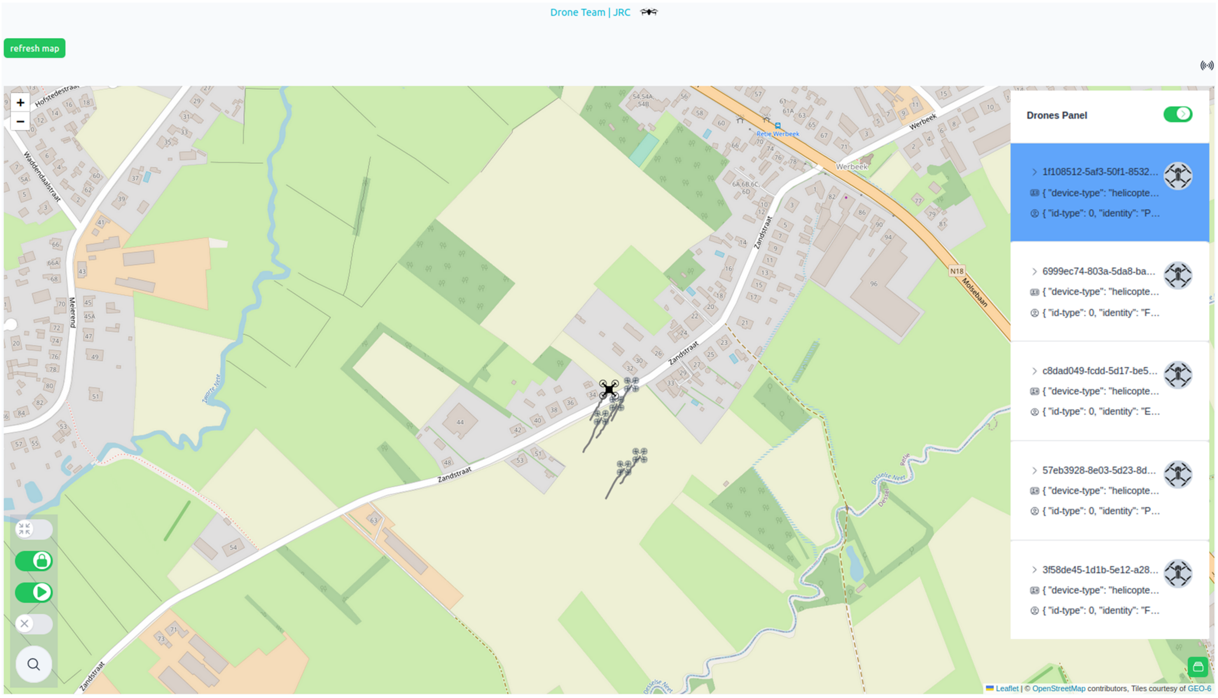


Fig. 6: Frontend application; GUI comprising map with detected drones (first drone is highlighted).

[8] A. Melnichuk, E. A. Kuzina, and N. K. Yurkov, "Methods and means for countering unmanned aerial vehicles," in *2020 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*, 2020, pp. 1–6.

[9] COURAGEOUS, "Review of current C-UAS Frameworks (Methods & Technologies) - D2.2 - V2.0," 2022.

[10] R. N. Kashi, A. Prashanth, S. R. Kashi, and G. Prabhakara, "A survey and analysis of drone detection systems using a systems approach superposed on scenarios," *Systems Engineering*, vol. 27, no. 3, pp. 598–636, 2024. [Online]. Available: https://incose.onlinelibrary.wiley.com/doi/abs/10.1002/sys.21735

[11] S. S. Blackman and R. Popoli, *Design and analysis of modern tracking systems*, ser. The Artech House radar library. Artech House, 1999.

[12] S. Hengy, M. Laurenzis, S. Schertzer, A. Hommes, F. Kloeppel, A. Shoykhetbrod, T. Geibig, W. Johannes, O. Rassy, and F. Christnacher, "Multimodal UAV detection: study of various intrusion scenarios," in *Electro-Optical Remote Sensing XI*, G. Kamerman and O. Steinvall, Eds., vol. 10434, International Society for Optics and Photonics. SPIE, 2017, p. 104340P. [Online]. Available: https://doi.org/10.1117/12.2278212

[13] C. G. L. Krishna and R. R. Murphy, "A review on cybersecurity vulnerabilities for unmanned aerial vehicles," in *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, 2017, pp. 194–199.

[14] "CRITICAL INFRASTRUCTURE PROTECTION & RESILIENCE - JRC DRONE Project to Counter UAS for CI." [Online]. Available: https://ec.europa.eu/newsroom/cipr/items/752427

[15] Joint Research Centre (European Commission), V. Karlos, and M. Larcher, *Protection against unmanned aircraf systems: handbook on UAS risk assessment and principles for physical hardening of buildings and sites*. Publications Office of the European Union, 2023. [Online]. Available: https://data.europa.eu/doi/10.2760/969680

[16] Joint Research Centre (European Commission), P. Hansen, and R. Pinto Faria, *Protection against Unmanned Aircraft Systems: handbook on UAS protection of critical infrastructure and public space : a five phase approach for C UAS stakeholders*. Publications Office of the European Union, 2023. [Online]. Available: https://data.europa.eu/doi/10.2760/18569

[17] V. U. Castrillo, A. Manco, D. Pascarella, and G. Gigante, "A review of counter-uas technologies for cooperative defensive teams of drones," *Drones*, vol. 6, no. 3, 2022. [Online]. Available: https://www.mdpi.com/2504-446X/6/3/65

[18] P. Boccadoro, D. Striccoli, and L. A. Grieco, "An extensive survey on the internet of drones," *Ad Hoc Networks*, vol. 122, p. 102600, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570870521001335

[19] R. Yuvaraj and S. Velliangiri, "A comprehensive study on unmanned aerial vehicle security issues," in *2023 IEEE 5th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA)*, 2023, pp. 544–550.

[20] C. Laoudias, A. Moreira, S. Kim, S. Lee, L. Wirola, and C. Fischione, "A survey of enabling technologies for network localization, tracking, and navigation," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3607–3644, 2018.

[21] "Implementing regulation - 2019/947 - EN - EUR-Lex," doc ID: 32019R0947 Doc Sector: 3 Doc Title: Commission Implementing Regulation (EU) 2019/947 of 24 May 2019 on the rules and procedures for the operation of unmanned aircraft (Text with EEA relevance.) Doc Type: R Usr_lan: en. [Online]. Available: https://eur-lex.europa.eu/eli/reg_impl/2019/947/oj

[22] "ASD-STAN prEN 4709-002 P1 | ASD-STAN." [Online]. Available: https://stan-shop.org/en/catalog/item/75040

[23] "F3411 Standard Specification for Remote ID and Tracking." [Online]. Available: https://www.astm.org/f3411-22a.html

[24] E. Commission, J. R. Centre, D. Amendola, D. Anderson, and S. Lagaras, *A counter-drone visualisation platform incorporating sensor-data fusion*. Publications Office of the European Union, 2024.

[25] O. Gold and M. Sharir, "Dynamic time warping and geometric edit distance: Breaking the quadratic barrier," *ACM Trans. Algorithms*, vol. 14, no. 4, aug 2018. [Online]. Available: https://doi.org/10.1145/3230734

[26] "DTAIDistance." [Online]. Available: https://zenodo.org/records/7158824