

Assessment of the Impact of U-space Faulty Conditions on Drones Conflict Rate

Anamta Khan^{1*}, Carlos A. Chuquitarco Jiménez², Morcillo-Pallarés Pablo²,
Naghmeh Ivaki¹, Juan Vicente Balbastre Tejedor², and Henrique Madeira¹

¹ CISUC, Department of Informatics Engineering, University of Coimbra, Portugal

`{anamta,naghmeh,henrique}@dei.uc.pt`

² Universitat Politècnica de València, Spain

`{carchuji,pabmorpa,jbalbast}@upv.edu.es`

Abstract. Unmanned Aerial Vehicles (UAVs) have gained notable importance in civil airspace. To ensure their safe operation, U-space services are being defined. We argue that the target level of safety of UAS can be threatened by faults, abnormal conditions, and security attacks. In this paper, we propose a fault-injection-based approach to build a framework to allow the safety assessment of UAS under these conditions. We created a simulation-based setup to mimic a realistic scenario to study the impact of these conditions on UAS conflict metrics and surveillance performance metrics. Results show a correlation between the faults (impairments) introduced and the degradation of some performance metrics indicating the quality of the measured trajectory by the U-space. These metrics can be used by the UAV conflict management service to ensure UAS safe operation.

Keywords: UAS · U-space · Target Level of Safety · Fault Injection

1 Introduction

Although Unmanned Aircraft Systems (UAS) [7] were significantly used in some military operations in the 1990s (Gulf War I), it was not until the beginning of the 21st century when the interest in this technology showed up in the civil world. Nowadays, there is a huge expectation of a variety of potential professional uses in many fields (e.g., urban mobility, delivery, public safety and security).

One of the main hazards to the operation of UAS is nearby UAS and manned aircraft with which they may collide, especially in urban scenarios where a high density of UASs is likely. To mitigate the risk of mid-air collisions, the UAS traffic management (UTM) concept was conceived in the early 2010s "to support the real-time or near-real-time organization, coordination, and management of UA operations" [15]. The UTM concept is currently being developed and implemented in many countries around the World.

* Corresponding author.

The European implementation of the UTM concept is the U-space, a set of services relying on a high level of digitalization and automation of functions, whether they are onboard, or are part of the ground-based environment [23], allowing the safe and efficient operation of a large number of UASs (especially, but not only, in urban VLL (very low level)). The maturity achieved by several U-space services [24] led the European Commission to lay down a regulatory framework for the U-space [1], [2], [3], hereinafter referred to as "the U-space regulation". The U-space regulation introduces the "U-space airspace" concept as a designated portion of the airspace where U-space services are provided by U-space service providers (USSP) and used by UASs operators when planning and conducting flights therein.

1.1 Conflict Management By The U-space

Conflict management is one of the safety pillars in aviation, *aiming at limiting the risk of collision between aircraft and hazards to an agreed level deemed as acceptable* [17]. When the hazard is another aircraft, a conflict is defined as *a predicted converging of aircraft in space and time which constitutes a violation of a given set of separation minima*, which are the *minimum distance between aircrafts that maintain the risk of collision at an acceptable level of safety* [16].

The U-space tackles conflicts between aircraft at two levels **(1) strategic** and **(2) tactical**. At the strategic level, UASs operators have to submit their flight plans, including the 4D trajectory of the UAS during the entire mission, to the *Flight Authorisation Service*, which checks whether it intersects both in space and time with trajectories in already approved flight plans.

At the tactical level, UASs have to submit their position, speed, and orientation to the *Network Identification Service* in real-time. The *Traffic Information Service* receives UAS information from the *Network Identification Service* and makes this information available to UASs operators, which have to *take the relevant action to avoid any collision hazard*. Although not required by the U-space regulation, a conflict detection capability can be added to the *Traffic Information Service* to serve as a safety net alerting pilots when a predefined separation threshold is infringed. In future U-space implementations, it is expected that tactical conflicts will be tackled by a *Tactical Conflict Resolution service* [7] that will apply a stepped process for conflict resolution.

1.2 Safety Assessment In The U-space

Safety is the cornerstone of aviation. The usual way to express the safety goal is the **Target Level of Safety (TLS)**, which represents the level of risk as the number of risk events divided by an exposure unit. In manned aviation, the risk event used to express TLS is *accidents*, whereas the exposure units can be either *flight-hours* for en-route aircraft or *movements* for taking-off, landing, and taxiing aircraft. Defining TLS for UASs is still controversial, and there is no common agreement either on the risk event or the exposure time. We will use

mid-air conflicts per flight hour as the primary safety measure. Although other risks (e.g., direct ground impacts) could also affect safety, mid-air conflicts per flight hour is the common TLS measure often used by U-space projects.

The ultimate purpose of the U-space is to guarantee safe and efficient access of UASs to the airspace. From the safety perspective, the U-space raises mitigation barriers between UASs and hazards both at strategic and tactical levels, as described in section 1.1. When designating a part of the national airspace as U-space airspace, competent authorities are required by the U-space regulation to assess the effectiveness of these barriers in that local scenario by means of a risk assessment meant. This assessment is a twofold process encompassing both **(1) a success approach** and **(2) a failure approach**. The success approach aims at demonstrating that U-space services can mitigate risks posed by *pre-existing hazards*, i.e., those arising during usual aviation operations under *normal* or *abnormal* conditions (rare *external* events that can negatively affect safety). The failure approach is conducted to assess how *system-generated failures* affect safety. Due to the complexity of the aviation system, most of this risk assessment is qualitative and based on experts' judgment. However, whenever a quantitative approach is possible, it should be applied.

This paper presents a fault-injection-based approach to quantitatively assess the effect of abnormal and faulty conditions (some of them caused by security issues) to the effectiveness of the tactical barrier provided by the U-space to mitigate the collision risk. The research is focused on the detection of the conflict, which is the first and most critical step in the tactical conflict resolution process as described in section 1.1 (non-detected conflicts will never be solved). Hence, the number of conflicts will be used instead of the number of collisions as a measure of the risk. The results show the importance of considering abnormal, faulty, and security conditions in the safety assessment of UASs and U-space services. Moreover, the results prove the effectiveness of our approach.

2 Related Work

This section reviews the related work on the safety assessment of UAVs and, in particular, fault-injection-based assessment of UAVs.

2.1 Safety Assessment Of UAVs

For assessment of UAVs operations management systems like the U-space services, the Joint Authorities on Rulemaking for Unmanned Systems (JARUS) [18] developed a risk-based methodology as Specific Operations Risk Assessment (SORA), which determines the safety level required for these operations considering both drone and ground stakeholders [21]. Then, U-space safety assessment (MEDUSA) is developed, which identifies and mitigates the relevant risks of drone operations supported by U-Space services by integrating the SESAR safety principles for the overall airspace system with the SORA approach that is focused on risk assessment of individual missions [4].

In addition to the above analytical approaches, we can find efforts in the literature that experimentally assess the safety (or security) of UAVs operations. A UAVs safety assessment approach is presented in [8], where a series of attacks (e.g., DoS) were injected into a real commercial drone to show how easily one can remotely control or bring the UAV down. In a similar study [13], a De-Authentication attack is emulated to demonstrate that anyone with access to a computer could potentially take down a drone. Similar studies are performed in [26] and [14] aiming to assess the security of commercially available drones. They present several security vulnerabilities found in drones and exploit them through a series of attacks (e.g., De-Authentication attack and buffer overflow).

2.2 Fault Injection For Safety Assessment

In a study [19] to create fault tolerance for UAVs, the authors created a Hardware in Loop Simulation (HILS) environment and studied three subsystems (Navigation, GPS, and transmission) injecting two types of faults (failure and signal strength), thus in a total of 6 faults to identify mitigation and define recovery mechanism for them. A similar study [12] using Simulink identified two common difficulties: a) in general, there are not enough fault samples in historical data to make it possible to cover most fault modes in UAVs; b) test flights does not offer a realistic way to identify and study these faults. The approach in this paper helps to solve both of the identified challenges in this study by using simulations to inject faults and study the impacts without using real vehicles. In a similar approach as previous works [20], the authors analyze the effects of GPS spoofing on drones through a series of tests in a HILS environment.

An attempt to study the impact of faults and also verify HILS environments is presented in [25]. The authors created their own simulator for fault injection purposes and also verified its accuracy, showing that the results of fault injection in simulation models (such as done in our paper) can be considered very close to real-world scenarios. Another recent fault injection platform [5], is using a very similar simulation model as in our approach, with PX4, Dronekit, and ArduPilot instead of Gazebo. It considers nine fault types including GPS faults (5 faults types) and Actuator Faults (4 faults types).

3 Approach And Experimental Setup

To verify and validate the behaviour of UAVs in faulty conditions, this work aims at creating a software-based fault injection framework providing all necessary tools for running experiments in faulty conditions. To achieve that, the following steps are required: i) definition and characterization of the system under assessment (SUA) and its environment (this is required for the definition of the missions); ii) identification and characterization of failure scenarios, helping to create a representative and fault model which is as complete as possible; iii) definition of safety assessment metrics for analysis of the obtained results, and

finally iv) design and implementation of fault injection framework that can be served in diverse types of UAV systems within diverse missions.

To realize the first step, it is important to identify diverse representative drone models in terms of hardware and software suitable for different application scenarios. It is also necessary to analyze, compare and identify the critical components of autonomous drones. Then, for each application scenario, it is needed to identify the most important environmental parameters. These help to define a set of realistic and representative missions, which are used as the workload for the experiments.

In the second step, a field data analysis is performed to identify and characterize the failure scenarios for the SUA, resulting in the definition of a fault model.

The third step is to identify the metrics allowing us to qualify and quantify the impact of each fault/failure/threat. This step is done by adapting the concept of surveillance performance monitoring from manned aviation to the U-space framework. As a result, we defined a set of metrics that evaluate the surveillance system performance quality. In parallel, a computation of the number of conflicts is done to assess and set safety thresholds for these metrics.

Finally, the last step is focused on the definition, design, and implementation and required techniques and tools for i) defining fault injection campaigns (e.g., software faults, security attacks, and network issues), ii) running the missions, iii) injecting the faults, iv) obtaining an analyzing the results.

When these essential requirements are met, experiments can be run. To assess the impact of faulty conditions, two sets of experiments need to be executed: **fault free runs (Gold runs)** and **Faulty runs**. The results obtained from gold runs are used as an oracle to assess the impact level of the injected faults.

3.1 Scenarios And Missions

In order to define a representative scenario with representative missions, we need to define the characteristics of the scenario, including i) dimensions of the area, ii) number of UASs per hour, and iii) type of trajectory to be followed. These

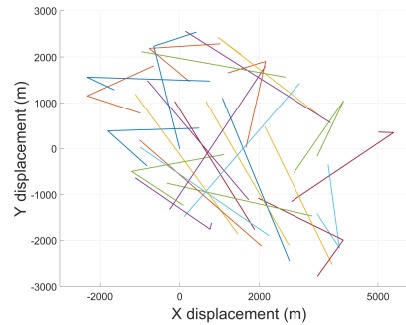


Fig. 1: General view of the scenario generated for the assessment

features are introduced in a trajectory generation software, namely BB-Planner, which generates a trajectory for each selected UASs. A file with all the necessary parameters and way-points (WPs) is generated for each trajectory.

In this work, we generated one scenario for running the experiments. This scenario is defined in an area of high-density controlled air traffic in the urban center of Valencia, Spain. It was designed to represent one of the most representative scenarios defined in [6] and also to meet with a TLS of $10E-6$ fatalities/hour. The simulated zone covers an area of 25 km^2 with a height limit of 120 meters (VLL) and a density of 28 UAS/h (i.e., 28 missions). Multirotor UAS of several categories with different power and velocity were used. Figure 1 shows a general view of the scenario generated for this study, where each line represents a UAS mission. The whole scenario takes about 1 hour to be completed.

3.2 Fault/Failure Model

Nowadays, GPS is the only surveillance system regulated for drones. Also, it has been shown that GPS is a highly vulnerable component [22] whose failures can lead to many conflicts, crashes, and even casualties. In the U-space framework, GPS information encapsulated in a surveillance data packet transmitted through the Network Identification Service is the only source of surveillance available for UTM. For these reasons, we selected GPS as the most critical component of UAV systems to be used as the target for our fault injection campaigns.

In this experiment, 14 **fault/failure types** were identified, taking into consideration 3 different condition types, i) *Faulty Conditions* (i.e., internal GPS failure or reception of incorrect data); ii) *Abnormal Conditions* caused by external factors (e.g., unavailability of GPS signals), and iii) Security conditions (e.g., hijacking), which are the most common conditions that a drone's GPS may face. The list of 14 fault/failure types is presented below:

1. **Fixed Valid Values:** Fixed valid value as GPS sensor input for Latitude, Longitude and Altitude (*faulty and security conditions*)
2. **Fixed Invalid Values:** Fixed invalid value as GPS sensor input for Latitude, Longitude and Altitude (*faulty and security conditions*)
3. **Missing Values:** Not receiving input values from GPS sensor (*faulty and Abnormal conditions*)
4. **Freeze Values:** Receiving same frozen GPS sensor input values for Latitude, Longitude and Altitude (*faulty and Abnormal conditions*)
5. **Random Value:** Receiving valid random GPS sensor input values for Latitude, Longitude and Altitude (*faulty and Abnormal conditions*)
6. **Min Value:** Receiving valid minimum GPS sensor input values for Latitude, Longitude and Altitude individually per fault (*faulty conditions*)
7. **Max Value:** Receiving valid maximum GPS sensor input values for Latitude, Longitude and Altitude individually per fault (*faulty conditions*)
8. **Fixed Noise:** Receiving a fixed value of noise in GPS sensor input values for Latitude, Longitude and Altitude (*faulty and Abnormal conditions*)

9. **Random Noise:** Receiving a random value (in range) of noise in GPS sensor input values for Latitude and Longitude (*faulty* and *Abnormal conditions*)
10. **Random Latitude:** Receiving a random value in GPS sensor input values for Latitude (*faulty* and *security conditions*)
11. **Random Longitude:** Receiving a random value in GPS sensor input values for Longitude (*faulty* and *security conditions*)
12. **Random Position:** Receiving a random value in GPS sensor input values for Latitude, Longitude and Altitude (*faulty* and *security conditions*)
13. **Slow Force Landing:** Forcing a drone to land by slowly increasing it's GPS sensor input values for Altitude (*security conditions*)
14. **Hijack:** Forcing the drone to move/land by tampering with its GPS sensor input values for Latitude, Longitude and Altitude (*security conditions*)

For each fault/failure type, the fault injection experiment is conducted for 4 different durations (namely, 2, 5, 10, and 30 seconds). Each fault instance was injected in the tactical phase at a random time between 30 seconds to 60 seconds after the completion of take-off. More than one test case was tested for some fault types (e.g., Random Noise). The result is a total of 74 cases to be studied.

3.3 Safety Assessment Metrics

As described in section 1.2, evaluating the TLS is fundamental in civil aviation safety. Since it is not possible to obtain this variable directly (as one metric), this study focuses on metrics related to conflict detection. In this study, two sets of safety assessment metrics are defined: i) conflict metrics and ii) surveillance performance metrics. The **conflicts metrics** are defined to assess and set safety thresholds (e.g., separation minima) and the surveillance performance metrics are considered to assess and evaluate the effect of the impairment on the aeronautical surveillance service. The conflict metrics defined are as follows:

- **Number of conflicts:** Is calculated as the number of times the separation volume of each UAS is intersected by the separation volume of a different UAS. This calculation is performed pairwise, and the number of conflicts is independent of the duration of the conflicts, 5 seconds being the minimum duration to consider a positive conflict.
- **Frequency of conflicts (conflicts/h):** Is calculated as the ratio of the total number of conflicts of the selected scenario divided by the sum of the total flight time of all UAS. These values indicate, together with the traffic density and the selected area, how effective the selected separation is.

To calculate the **surveillance performance metrics**, a Surveillance Performance Monitoring Tool is used, which works based on some generic specifications and requirements for Air Traffic Control (ATC) surveillance systems [11]. The performance-based surveillance approach aims to evaluate surveillance systems in a technology-agnostic way.

The tool compares the data provided by the surveillance systems (in this case, it is just the system that provides the U-space Network Identification surveillance service) with the high-quality trajectories computed from the received telemetry. To statistically obtain relevant performance metrics, a minimum of 50,000 position data cases are needed.

The telemetry received from the UAVs must be suitable in various aspects for the UTM system that want to use it. The data must be delivered with certain minimum conditions of completeness, codification, precision, update rate, latency, and integrity. These requirements can be reduced to a set of performance metrics [10] [9] that are defined as follows:

- **Probability of Update (PU):** This metric refers to the probability that a True Target Report (TR) is associated with a reference trajectory within an Update Interval (UI) defined by the user (one second in this case). A True Target Report is a target report whose positioning distance between the position measurement in a time and the position in the reference trajectory at the same time is below a threshold. The UI is a requirement of the application in which this data is used, in this case, 1 second.
- **Probability of Long Gap (PLG):** This metric refers to the probability of not receiving a True TR during a number of UIs greater than or equal to n .
- **Probability of False Track (PFT):** This metric refers to the probability of having a false track in a trajectory. A false track is defined as a consecutive number of false target reports (3 TRs in this study) correlated in the 3D space. A False Target Report is a target report in which the distance between the position of a UAV in a faulty trajectory and its position in the reference trajectory is higher than a certain threshold.

3.4 Experimental Framework

Figure 2 presents a general view of the process followed to run the whole experiment from the definition of flight plans to analysis of the results.

The BBPlanner generates flight plans/missions for a given scenario. The generated missions are executed within the fault injection environment. The generated telemetry in this environment is then transmitted to the Conflict Computation module (running on BBPlanner) and the Surveillance Performance Monitoring Tool for calculation of the safety assessment metrics. This setup intends to emulate a non-real-time U-space [6] architecture with the minimum modules and interfaces needed to make a safety assessment. This justifies the existence of Tracker between the telemetry source and the conflict detection module.

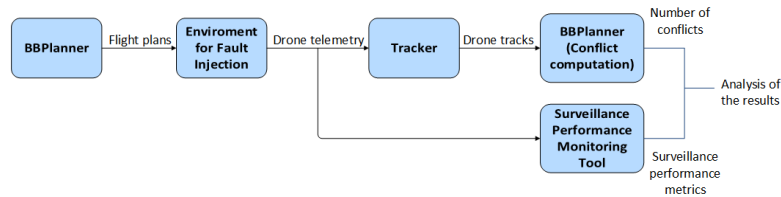


Fig. 2: General experimentation setup

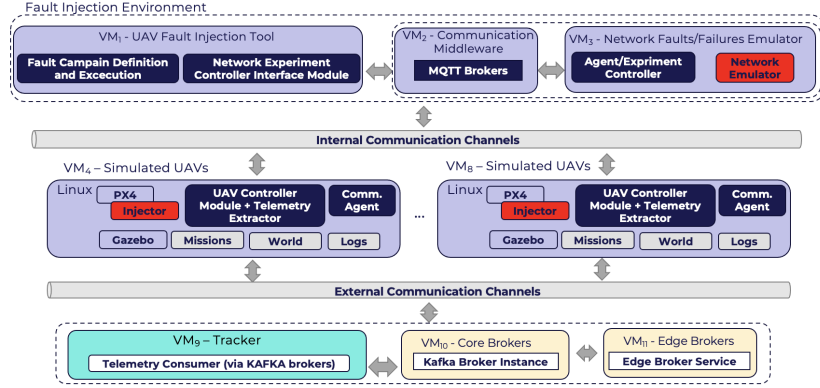


Fig. 3: Detailed View of the Fault Injection Environment.

Fault Injection Environment, whose overall view is presented in Figure 3, contains all required components for the definition of a fault injection campaign, the injection of faults, the execution of the missions on several UAVs, the extraction of flight logs, and the analysis of the trajectories. All these components are completely developed, deployed, and running within a VMware ESXi virtualized environment.

The fault injection campaigns are defined on the fault inject tool (VM_1). The defined faults will either be injected into the UAVs flight controller (currently only PX4 is supported by our fault injection environment) or into communication network (not done in this study) through the network emulator (VM_3) which is triggered after receiving the fault injection command through a communication middleware (VM_2). The UAVs are all running on a simulation environment created using Gazebo and PX4. Currently five machines (VM_4 to VM_8), each one with the capacity of running multiple UAVs, are dedicated to the UAVs. The tracking system (including a tracker, core brokers, edge brokers are running on VM_9 to VM_{11}) consumes the generated telemetry of UAVs for the calculation of the safety assessment metrics.

4 Results And Analysis Of The Results

The effect of the impairments injection can be directly seen comparing the faulty trajectories against the gold trajectories. Although a straightforward and quantitative analysis of the impact of faults could be done, this may not represent the real effects of impairments from a UTM point of view. A more thorough study in two perspectives in line with the U-space concept of operations [7] was carried out. Thus, this approach intends, on one hand, to assess and analyse the impairment effects on a UAS conflict detection tool and on a surveillance performance evaluation tool and, on the other hand, to relate the assessment done for each tool to find a preliminary set of metrics values in the surveillance performance metrics that may lead to a safety action concerning UAS separation management/conflict management.

4.1 Assessment Of The Impact On The Conflicts

In order to make the trajectories as realistic as possible, they are processed by a Kalman filter before being consumed by the conflict-counting algorithm. We do it to simulate the behaviour of a real UAS tracker. The effects of impairment/faults are reflected in the results in which the fault injection duration is of 2 seconds, where the difference between the faulty and gold run are practically negligible and do not affect the system (because the Kalman filter absorbs the effect of short duration failures). In Figure 4, the difference in the number of conflicts between the faulty run and the Gold run, does not exceed 2 conflicts in any of the studied cases. This maximum value occurs in the case of minimum altitude error, the rest of values varies between 1 or 0.

The results obtained for fault injection duration of 5 and 10 seconds (Figures 5 and 6) shows that the number of conflicts increases significantly when *Minimum Lat/Lon* and *Maximum Lat/Lon* fault types are injected. This is because, at the same instant of time, all the drones move in the same direction, resulting in drones' trajectories approaching each other and thus increasing the number of conflicts. In the cases in which the position of the drones is modified, such as *random lat/lon* and *random position* fault types, a similar effect is observed, but the number of conflicts is not as high as in the previous cases. Since the values selected for these cases are random, the directions the drones will take are highly dependent on these points.

The results for the fault injection duration of 30 seconds (Figure 7) show that the impact of previously mentioned fault types (*Minimum Lat/Lon* and *Maximum Lat/Lon*) becomes even more significant when the fault is injected for a longer period of time. For instance, in the case of *Random Lat/Lon*, the number of conflicts increases from 21 to 45 with a fault duration of 30 seconds.

From the results presented in Figure 7, we also observed that the fault type of *Missing Values* causes a significant decrease in the number of conflicts when compared to the gold run results. This happens due to the fact that the PX4 position estimate falls below acceptable levels, which is caused by GPS loss (injected fault). This triggers the Position Loss Failsafe, causing UAVs to descend to the ground, aborting the mission.

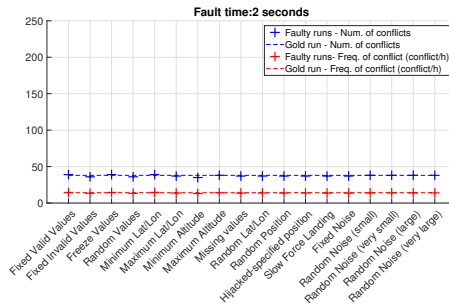


Fig. 4: Impact on conflict metrics (fault injection duration: 2 seconds)

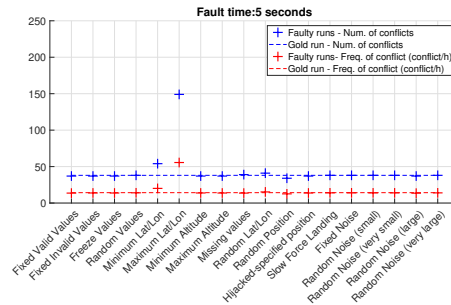


Fig. 5: Impact on conflict metrics (fault injection duration: 5 seconds)

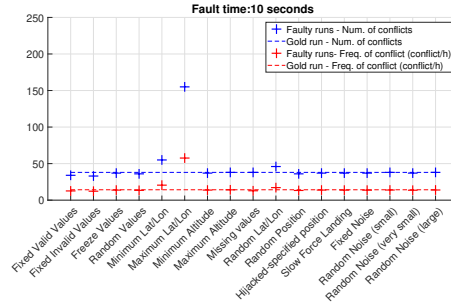


Fig. 6: Impact on conflict metrics (fault injection duration: 10 seconds)

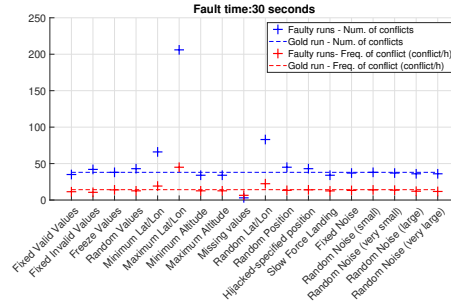


Fig. 7: Impact on conflict metrics (fault injection duration: 30 seconds)

The rest of the fault types shown in the figures are not affecting the conflict metrics to the same extent as those analyzed in this section. This is due, on the one hand, to smoothing of the trajectories by the Kalman filtering that Tracker uses internally and, on the other hand, to the conflict algorithm, in which a conflict must last for at least 5 seconds to be considered as a positive conflict.

4.2 Assessment Of The Impact On The Surveillance Performance

Figure 8 shows the impact of injected fault types on the Probability of Update for various fault injection durations. The probability of update depends on several factors: lost/incomplete/damaged reports/packages, latency, jitter, and, to a less important extent, positioning error. Here, this metric is mainly affected by GPS errors caused either by faulty, abnormal, or security conditions.

The results show that most of the fault types (Valid/Invalid Fixed Values; Random values, position, latitude, and longitude; Min/Max latitude, longitude and altitude; Random noise (large), and the Hijack), even when injected for a short period of time, had an impact on this metric, and the impact increased by increasing the fault injection duration.

The degradation of the performance metrics happens due to the fact that these impairments increase the positioning error greatly. The impact on this metric is even more significant when the fault injection duration goes above 30 seconds. In these cases the PU falls below 90%. In contrast, *Freeze Values*, *Minimum Altitude*, *Slow Force Landing*, *Small Fixed Noise*, and *Small Random Noise* faults do not affect this metric critically.

Figure 9 presents the impact of impairments on Probability of Long Gap (PLG). When the "gap" between true TRs becomes beyond 3UI or 4UI, we consider it a sensitive situation as the UI is equal to 1 second, and the maximum speed considered for each drone in this study is up to 20 m/s. The results show a similar impact on PLG when compared to PU. The reference values (based on gold runs) for this metric are around 0.03%. Therefore, when faults are injected for more than 2 seconds, this metric hardly meets the requirement for TLS.

Figure 10 presents the impact of the injected faults on the Probability of False Track (PFT). This metric is only affected by the positioning error and, by

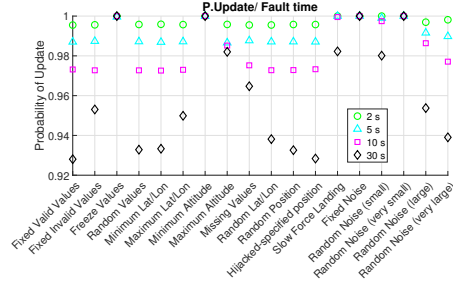


Fig. 8: Probability of Update

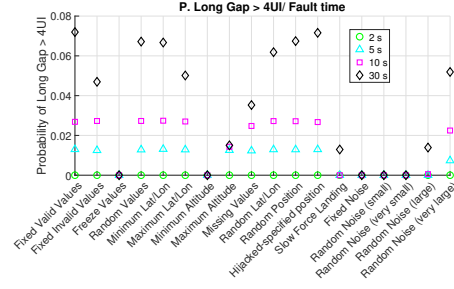


Fig. 9: Probability of Long Gap

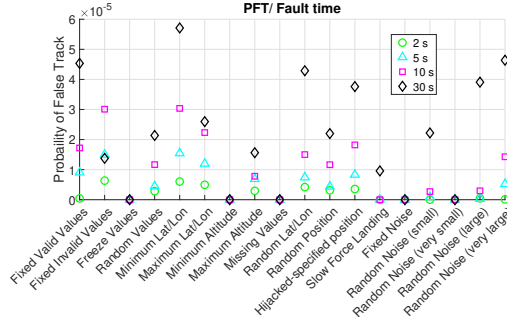


Fig. 10: Impact on Probability of False Tracks

definition, indicates how likely it is that there is a given time-correlated spatial error bias on a given sensor. Thus, small random values injected into latitude, longitude, and altitude do not affect PFT significantly. The impact becomes severe when large random values are injected. No negative effect is observed in the case of *Freeze Values*, *Minimum Altitude*, and *Fixed Noise* faults. This happened because no positioning error is injected with the freeze and missing values. In the other cases, the injected values are not high enough to observe any false telemetry report in the faulty runs.

4.3 Discussion

Considering that the faults analyzed in this study are worst-case scenarios, it is interesting to relate the results obtained in sections 4.1 and 4.2 to suggest threshold values or at least a range for the surveillance performance metrics. Values selected are the following: $PU = [0.995 - 0.987]$, $PLG = [0.001 - 0.0128]$, $PFT = [4.956E-6 - 1.197E-5]$. These ranges were selected, taking into account the worst case in the conflict assessment. The conflict metrics are defined for a defined level of safety (TLS). An increase of 5% in the frequency of conflict is considered unacceptable. This increase, for the worst cases, happens between the faults of duration between 2 and 5 seconds. Since the PU is the most sensitive metric (as being affected by various impairments), it is more likely that this metric is the factor that triggers an alert on the degradation of the surveillance

service and its possible effect on the overall safety of the airspace volume analyzed. This alert would lead to a conflict management action taken by the USSP and the pilots/operators in charge of the corresponding UAS. As is proposed in [6] the USSP would receive the surveillance degradation alert and consequently would update the separation minima between drones. Updating the minimum separation between drones would instantly increase conflicts between drones in operation and then reduce them by mechanism targeting the appropriate level of security. This is because when a drone has a conflict, the pilot/operator receives an alert from the USSP that must resolve by a deconflict decision through their own judgment or by an indication provided by the USSP.

5 Threats To Validity

Although the results of this study are very representative for realistic scenarios, there are some limitations that could be taken into consideration for future studies. To verify the results and compare them among each other, the study uses a single scenario. Despite the scenario being pragmatic, in future studies, more scenarios can be experimented with and can be compared with each other. This study also considers the same drone model in each experiment and uses a single flight mode; in future studies, diverse types of UAVs can be observed with multiple flight modes. This study only considers GPS faults, but it would be really insightful to include different communication cases (e.g., latency).

6 Conclusion

Aiming to assess conflict and surveillance performance of the U-space services through qualitative analysis, a fault-injection-based framework was developed to simulate drone flights and emulate 14 types of faulty/abnormal/security issues concerning the UAV GPS module in realistic scenarios. The results suggest that these conditions significantly impact both the conflict and surveillance performance metrics. Internal GPS failures such as maximum values and missing values (which forced the drone to land) tend to have a greater impact on conflict metrics. On the other hand, it has also been observed that a short duration of such faults/failures does not affect the metrics. However, a longer duration, such as 30 seconds or more, has a significant impact, especially on surveillance performance metrics. The framework developed and results give us indications on how the U-space services can collaborate to continuously monitor the communications and surveillance systems in order to manage conflicts to support safe UAV operations.

Acknowledgment: This work was partially funded by the European Union in the scope of the BUBBLES Project (SESAR JU, 2020), funded in the scope of the SESAR Joint Undertaking (SESAR JU), under the Horizon 2020 Research and Innovation Program (agreement number 893206).

References

1. COMMISSION DELEGATED REGULATION (EU) 2021/664 of 22 April 2021 on a regulatory framework for the U-space. OJ **64** (2019-04-23)
2. COMMISSION DELEGATED REGULATION (EU) 2021/665 of 22 April 2021 amending Implementing Regulation (EU) 2017/373. OJ **64** (2019-04-23)
3. COMMISSION DELEGATED REGULATION (EU) 2021/665 of 22 April 2021 amending Regulation (EU) No 923/2012 as regards requirements for manned aviation operating in U-space. OJ **64** (2019-04-23)
4. Barrado, C., et al.: U-space concept of operations: A key enabler for opening airspace to emerging low-altitude operations. *Aerospace* **7**(3), 24 (2020)
5. Bo, C., Benkuan, W., Yuntong, M., Yu, P.: A fault injection platform for multirotor uav phm. In: IEEE Int. Conf. Electronic Measurement Instruments (ICEMI) (2019)
6. BUBBLES: BUBBLES project, <https://bubbles-project.eu/>
7. CORUS project: U-space Concept of Operations. SESAR JU (2019)
8. Deligne, E.: Ardrone corruption. *Journal in Computer Virology* (2012)
9. EUROCAE: ED-129B. Technical Specification for a 1090 MHz Extended Squitter ADS-B Ground System. 2016
10. EUROCAE: ED-142A. Technical Specification for a Wide Area Multilateration GroundSystem with Composite Surveillance Functionality. 2019
11. EUROCAE: ED-261-1. Safety and Performance Requirements Standard for a Generic Surveillance System (GEN-SUR SPR), <https://www.eurocae.net/news/posts/2020/january/eurocae-open-consultation-ed-261-1/>
12. Gong, S., et al.: Hardware-in-the-loop simulation of uav for fault injection. In: 2019 Prognostics and System Health Management Conference (PHM-Qingdao) (2019)
13. Gordon, J., Kraj, V., Hwang, J.H., Raja, A.: A security assessment for consumer wifi drones. In: IEEE International Conference on Industrial Internet (ICII) (2019)
14. Hooper, M., Tian, Y., Zhou, R., Cao, B., Lauf, A.P., Watkins, L., Robinson, W.H., Alexis, W.: Securing commercial wifi-based uavs from common security attacks. In: MILCOM 2016-2016 IEEE Military Communications Conference. IEEE (2016)
15. International Civil Aviation Organisation: Unmanned Aircraft Systems Traffic Management (UTM) – A Common Framework with Core Principles for Global Harmonization. Edition 3. International Civil Aviation Organisation
16. International Civil Aviation Organisation: ICAO Doc. 9426 Air Traffic Services Planning Manual (1992)
17. International Civil Aviation Organisation: ICAO Doc. 9854 Global Air Traffic Management Operational Concept. International Civil Aviation Organisation (2005)
18. JARUS: JAR doc 06 sora, <http://jarus-rpas.org/content/jar-doc-06-sora-package>
19. Kumar Chandhrasekaran, V., Choi, E.: Fault tolerance system for uav using hardware in the loop simulation. In: 4th International Conference on New Trends in Information Science and Service Science (2010)
20. Mendes, D., Ivaki, N., Madeira, H.: Effects of gps spoofing on unmanned aerial vehicles. In: 2018 IEEE 23rd Pacific Rim International Symposium on Dependable Computing (PRDC). IEEE (2018)
21. Miles, T., Suarez, B., Kunzi, F., Jackson, R.: Sora application to large rpas flight plans. In: IEEE/AIAA 38th Digital Avionics Systems Conference (DASC) (2019)
22. Nighswander, T., Ledvina, B., Diamond, J., Brumley, R., Brumley, D.: Gps software attacks. In: Proceedings of the 2012 ACM Conference on Computer and Communications Security. Association for Computing Machinery (2012)
23. SESAR JU: U-space blueprint. SESAR JU (2017)

24. SESAR JU: Consolidated report on SESAR U-space research and innovation results. SESAR JU (2020)
25. Wen, J., Wang, H., Zhang, M., Li, D., Wu, J.: Design of a real-time uav fault injection simulation system. In: IEEE Int. Conf. Unmanned Systems (ICUS) (2019)
26. Yihunie, F.L., Singh, A.K., Bhatia, S.: Assessing and exploiting security vulnerabilities of unmanned aerial vehicles. In: Smart Systems and IoT: Innovations in Computing. Springer (2020)