SESAR JOINT UNDERTAKING

Final Collision Risk Model

Deliverable ID:	**D3.6**
Project acronym:	**U-AGREE**
Grant:	**101167187**
Call:	**HORIZON-SESAR-2023-DES-ER-02**
Topic:	**HORIZON-SESAR-2023-DES-ER2-WA1-4**
Consortium coordinator:	**UPV**
Edition date:	**20 May 2026**
Edition:	**01.00**
Status:	**Official**
Classification:	**PU**

Abstract

This document presents the final formulation of the collision risk model developed by the U-AGREE consortium. The model applies a Monte Carlo approach to estimate the average number of collisions over a large number of simulated flight hours and to derive the corresponding collision frequency. Realistic trajectories are generated using a PX4 Software-in-the-Loop (SITL) simulator to capture representative UAS flight behaviour. The proposed model has been benchmarked against the extended Endoh's model developed as part of the initial U-AGREE collision risk model. In addition, a sensitivity assessment has been conducted to identify the main sources of uncertainty that may affect the obtained results

Authoring & approval

Author(s) of the document

Organisation name	Date
EUROUSC	14-05-2026
UPV	14-05-2026

Reviewed by

Organisation name	Date
ENGage srl	14-05-2026
COLLINS	14-05-2026

Approved for submission to the SESAR 3 JU by

Organisation name	Date
UPV	14-05-2026
EUROCONTROL	20-05-2026
EUROUSC ITALIA	20-05-2026
ENGage srl	14-05-2026
DLR	20-05-2026
COLLINS	14-05-2026
ENAIRE	20-05-2026
CRIDA	20-05-2026
INECO	20-05-2026
CRA	20-05-2026

Rejected by[1]

Organisation name	Date

[1] Representatives of the beneficiaries involved in the project.

Document history

Edition	Date	Status	Organisation author	Justification
00.01	23rd April 2026	Draft	UPV	Initial draft
00.02	14th May 2026	Final draft	UPV	Final draft for partners review
01.00	20th May 2026	Official	UPV	Final document

Copyright statement

Disclaimer

U-AGREE

U-space Air and Ground Risk modEls Enhancement

U-AGREE

This document is part of a project that has received funding from the SESAR 3 Joint Undertaking under grant agreement No 101167187 under European Union's Horizon Europe research and innovation programme.

Table of contents

EUROPEAN PARTNERSHIP
Co-funded by the European Union

List of tables

List of figures

EUROPEAN PARTNERSHIP

Co-funded by the European Union

1 Executive summary

This deliverable presents the final collision risk model (CRM) develop by the U-AGREE project aimed to be integrated in the final formulation of the concept alongside the final air and ground risk models set out in U-AGREE deliverables D3.4 and D3.5, respectively, and the final models of the risk mitigation barriers reported in U-AGREE deliverable D4.2. This CRM applies a Monte Carlo approach to compute the number of collisions over a large number of simulated flight hours to derive the collision frequency. The model utilizes realistic UAS trajectories built using a software-in-the-loop (SITL) trajectory generator based on a PX4 flight simulator to capture the particular behaviour of UAS. The simulator runs on a Gazebo environment to account for environmental factors that might impact the flight execution uncertainty.

The collision risk model has been first benchmarked against the extended Endoh's model developed in the initial U-AGREE collision risk model reported in deliverable D3.3 using random linear trajectories. Then, the SITL trajectory generator has been used to generate a large data set containing realistic trajectories representative of actual UAS missions (delivery, trajectories between fixed points, scan and random) flown by UAS of different types. Then, the collision risk model randomly picks trajectories up from the data set according to a pre-defined traffic mix and runs thousands of simulated flight hours with different numbers of simultaneous UAS. In the short term, given the lack of data to obtain both the type of missions and the composition of the UAS fleet, there figures must be defined based on traffic outlooks. From the mid-term onwards, these values should be refined using recorded historical data. Interactions between UAS and manned traffic have been characterised using ADS-B data provided by the U-AGREE partner COLLINS. In the mid-term, this information should be expanded using other e-conspicuity data sources. In line with the results reported in deliverable D3.3, the collision frequency linearly depends on the number of simultaneous UAS (i.e., the UAS density), while the slope of the curve exclusively depends on the characteristics of the airspace (i.e., the airspace volume, the composition of the fleet, the type of trajectories and the environmental conditions affecting the flight execution uncertainty, expressed as the Total System Error – TSE). A sensitivity assessment has been conducted to identify the sources of uncertainty in the computed results, like the simulation time step, the number of simulated flight hours or the way the trajectories are picked up.

This deliverable also presents the 3D encounter model that U-AGREE will use in WP5 to estimate the mitigation effectiveness of tactical barriers. The encounter model generates a huge number of converging trajectories using the PX4 SITL trajectory generator and applies a Monte Carlo approach to estimate the frequency of tactical conflicts, loss of separation and near mid-air collisions from which different safety (i.e., risk ratios) and operational efficiency (i.e., alert rate and probability of false alarm) metrics can be estimated. The encounter model generates trajectories at different altitudes uniformly distributed within a predefined range and accounts for random turns and vertical manoeuvres according to specific probabilities. The encounter model also accounts for a variety of communication, navigation and surveillance (CNS) performances, as well as for different human-in-the-loop behaviour patterns.

2 Introduction

2.1 Purpose and scope of the document

This document describes the final collision risk model developed by the U-AGREE consortium to estimate the unmitigated collision frequency within an airspace volume where different mixes of manned and unmanned traffic coexist. The model will be integrated into the final U-AGREE risk model (v2). In addition, this document presents the encounter model that will be used within U-AGREE to estimate the mitigation effectiveness of tactical barriers in WP5.

This deliverable serves a twofold purpose. First, it aims to provide evidence of the validity of the collision frequency results by benchmarking them against the analytical formulations described in Deliverable D3.3 under equivalent conditions. Second, it supports the integration of both the collision risk model and the encounter model into the simulation environment that will be used in WP5 to validate the U-AGREE concept, including implementation details and a sensitivity assessment.

The scope of this document is limited to these two objectives. The examples presented herein are provided exclusively for validation purposes and do not carry operational relevance. Whenever possible, the simulations have been configured to obtain representative outcomes while limiting computational effort. Therefore, the presented results should not be extrapolated to the project operational environments, which are specifically addressed in Deliverable D5.2.

2.2 Intended readership

Given that this document is intended to support the integration of the U-AGREE risk model, the consortium itself constitutes one of its main target audiences, particularly the team involved in integrating the risk models into the simulator that will be used in WP5 for validation purposes.

Beyond the U-AGREE consortium, two additional key target audiences can be identified. First, the document provides valuable input for aviation authorities that need to estimate collision risk in U-space airspaces using reliable and trusted methods. State-of-the-art approaches traditionally applied in manned aviation are not fully capable of addressing the diversity of performances and missions envisaged in these operating environments. The method proposed by U-AGREE builds upon accepted approaches, such as the Endoh gas model, together with well-established statistical techniques, including the Monte Carlo method, to derive meaningful risk metrics capable of capturing the complexity of U-space operations. Standardisation bodies, particularly JARUS, also fall within this category of intended audience.

The second target audience is the academic community, which is actively working on risk assessment in complex operational scenarios. The work presented in this document represents a relevant contribution to the existing body of knowledge in the field and is expected to serve as a foundation for future research and further developments.

2.3 Background

The U-AGREE project builds upon prior SESAR exploratory research and external studies on collision risk modelling, including the BUBBLES project and the Reich and Endoh models. These models have been instrumental in structured and unstructured airspace safety assessments. The extended gas model proposed here addresses the limitations of earlier approaches by incorporating diverse UAS characteristics and enabling scenario-specific risk estimation. This work aligns with the SESAR Strategic Research and Innovation Agenda (SRIA) and contributes to the safe integration of drones into European airspace.

2.4 Structure of the document

The document is structured as follows:

- Chapter 3 reviews the state-of-the-art in collision risk modelling.
- Chapter 4 provides insight into the U-AGREE collision risk models and assesses the sensitivity of the numerical collision risk model against different parameters.
- Chapter 5 details the U-AGREE encounter model.

2.5 Glossary of terms

Term	Definition	Source of the definition
CRM	A mathematical or computational framework used to estimate the likelihood of mid-air collisions between aircraft. CRMs are essential for evaluating separation standards, safety barriers, and CNS performance in both manned and unmanned aviation.	JARUS SORA Guidelines
U-space	A set of services and procedures designed to support safe, efficient, and secure access to airspace for large numbers of drones. U-space enables complex operations, including beyond visual line of sight (BVLOS), particularly in urban environments.	EASA AMC/GM for U-space Regulation (EU) 2021/664
MAC	An event in which two aircraft come into physical contact while airborne. In safety assessments, MACs are used as a critical metric for evaluating airspace risk.	EASA Good Practice on Flight Data Monitoring
Unmitigated Risk	The level of risk assessed without the application of any mitigation measures or safety barriers. It serves as a baseline for evaluating the effectiveness of risk controls.	EASA Guidance on Hazard Identification and Risk Assessment
NMAC	An NMAC is a situation in which *two aircraft come within a given distance of each other while in flight, both horizontally and vertically. For UAS*	ASTM, F3442/F3442M-20 Standard Specification for Detect and Avoid System

EUROPEAN PARTNERSHIP
Co-funded by the European Union

	with less than 25 ft wingspan, the applicable NMAC distances are 30 ft vertically and 100 ft horizontally.	Performance Requirement, May 1st, 2020.

Table 1: Glossary of terms

2.6 List of acronyms

Term	Definition
ATM	Air traffic management
ATS	Air traffic services
CNS	Communication navigation surveillance
CRM	Collision Risk Model
DAA	Detect-and-avoid
DES	Digital European Sky
Eq.	Equation (used to reference formulas like Eq. 1, Eq. 2, etc.)
ERP	Exploratory research plan
FTE	Flight Technical Error
GA	Grant agreement
GDPR	General data protection regulation
HE	Horizon Europe
ID	Identifier
ION	Ionospheric delay modelling (ION)
MAC	Mid-Air Collision
NAS	National Airspace System
NMAC	Near Mid-Air Collision
NAT	North Atlantic
NM	Nautical Miles
NMAC	Near mid-air collision
NSE	Navigation System error
OSED	Operational service and environment description
SESAR	Single European sky ATM research
SESAR 3 JU	SESAR 3 Joint Undertaking
sNMAC	Small Near mid-air collision

SORA	Specific Operations Risk Assessment
SRIA	Strategic research and innovation agenda
TLS	Target Level of Safety
TMA	Terminal Manoeuvring Areas
TRL	Technology Readiness Level
TSE	Total System Error
U-space	European airspace designated for drone operations
UTM	Unmanned traffic management
VLL	Very Low Level

Table 2: List of acronyms

3 Collision modelling in aviation

3.1 Collision risk models in aviation

3.1.1 What Is a Collision Risk Model?

A Collision Risk Model (CRM) is a mathematical or computational framework designed to estimate the likelihood of mid-air collisions between aircraft. These models simulate aircraft trajectories and assess the probability of breaching defined separation thresholds. CRMs are essential for airspace safety analysis, air traffic management (ATM), and unmanned traffic management (UTM), particularly in evaluating separation standards and mitigation strategies [1] [2] [3].

3.1.2 Applications of Collision Risk Models in Aviation

Collision risk models serve several critical functions in aviation:

- Quantifying collision frequencies: CRMs estimate the expected number of mid-air collisions or near mid-air collisions (NMACs) per unit of time or flight hours, supporting safety assessments and regulatory compliance [1] [2].
- Determining mitigation effectiveness: CRMs help define the required performance of strategic and tactical mitigation layers (e.g., separation provision, detect-and-avoid systems) to meet target levels of safety (TLS) [3].
- Specifying CNS performance: CRMs inform the necessary performance of communication, navigation, and surveillance (CNS) systems to ensure reliable conflict detection and resolution [3].

3.1.3 Collision Risk Models vs. Encounter Models

While both CRMs and encounter models simulate aircraft interactions, they differ in scope:

- CRMs focus on estimating the probability of collision or NMACs using statistical representations of aircraft behaviour, separation standards, and system performance. They are often used in regulatory and safety assurance contexts [1] [2].
- Encounter models, such as the Lincoln Lab NAS model, generate realistic aircraft trajectories based on empirical radar data. These are typically used in simulation-based evaluations of detect-and-avoid (DAA) systems and offer detailed representations of aircraft dynamics and environmental variability [4].

Although both frameworks deal with airborne interactions, they answer different questions and therefore play complementary roles in airspace safety assessment. CRMs are statistical or analytical frameworks whose output is a collision frequency expressed in occurrences per flight-hour (e.g., MACs/FH or NMACs/FH). They are typically grounded in aggregated descriptors of the fleet — traffic mix, mean and variance of speed, nominal trajectories and aircraft density — and are best suited to produce benchmark figures for regulators, to derive Target Levels of Safety (TLS) and to compare them against the safety outcome of alternative operational concepts [1] [2] [5]. The extended gas model developed in Chapter 4 is a representative example: it yields a closed-form expression for the

Co-funded by the European Union

unmitigated collision rate given macroscopic airspace parameters but does not model individual encounter geometries nor the behaviour of any tactical mitigation service.

By contrast, encounter models are generative: they produce statistically representative pairs (or sets) of aircraft trajectories that emulate the kinematic and behavioural diversity observed in actual operations. The MIT Lincoln Laboratory Uncorrelated Encounter Model of the U.S. National Airspace System (NAS) [3] and its correlated counterpart use Dynamic Bayesian Networks (DBNs) trained on radar track data to reproduce the time-evolving distributions of altitude, airspeed, vertical rate and turn rate of aircraft. The Eurocontrol CRÈME extension adapts this paradigm to European traffic using more than 12 million flight-hours of radar data and introduces an aircraft-centric architecture with performance-class and controlled-status nodes. Weinert et al. [8] further proposed a quantitative NMAC analogue (sNMAC) with 50 ft × 15 ft volumes tailored to the physical dimensions and likely collision outcomes of small UAS, enabling robust statistical analysis when MAC events are too rare to be observed directly in simulation. Pastor et al. [9] introduced the Drone Encounter Generation (DEG) model for U-space, in which encounters are generated by stochastically sampling U-plan segments around a reference NMAC point and then propagating the resulting waypoint-driven trajectories through a UAS dynamics engine. Alvarez et al. [10] illustrate how such encounter sets are exploited for the decentralised evaluation of DAA solutions such as ACAS sXu.

The two approaches are thus not interchangeable but complementary. CRMs provide the absolute baseline collision rate against which mitigations can be benchmarked and scale well to macroscopic airspace questions (density, traffic mix, airspace sizing). They yield figures that can be compared with a TLS but reveal little about how a tactical mitigation service actually behaves in a specific encounter geometry. Encounter models, in turn, deliver realistic trajectories that exercise the tactical mitigation chain end-to-end — surveillance, tracking, conflict prediction, conflict resolution and pilot/autopilot response. They are pivotal for estimating conditional metrics such as the Risk Ratio for NMAC, given by $RR = P(\mathrm{NMAC} \mid \mathrm{TCR})/P\left(\mathrm{NMAC} \mid \overline{\mathrm{TCR}}\right)$, as well as the Loss-of-Separation Ratio LR, the false-alarm probability P_{fa} and the alert rate [7] [8]; but they do not, by themselves, indicate whether the absolute NMAC frequency is acceptable for a given U-space volume.

Within U-AGREE both artefacts are therefore developed in parallel. The extended gas model of Chapters 3 and 4 provides the unmitigated collision frequency for a given U-space volume, whereas the pairwise encounter model presented in Chapter 5 is used to characterise the residual risk once the tactical conflict management barriers defined in U-AGREE deliverable D4.1 are in place. The combination of both enables a full risk assessment in which the baseline collision rate produced by the CRM is weighted by the risk ratio produced by the encounter-based assessment, in line with the layered conflict management paradigm of ICAO's GATMOC and the current DAA standards [7].

3.2 Key Collision Risk Models in Aviation

3.2.1 Reich Model

The Reich model is one of the earliest analytical models for estimating mid-air collision risk. It was developed to assess the safety of reducing lateral separation standards between aircraft flying on parallel tracks, particularly over oceanic airspace like the North Atlantic (NAT) [1] [5].

The model treats aircraft as rectangular volumes and calculates the probability of overlap in three dimensions, longitudinal, lateral, and vertical, based on statistical deviations from nominal flight paths. The collision rate C is given by:

$$C = \lambda_x P_x(0) \cdot \lambda_y P_y(S_y) \cdot \lambda_z P_z(0) \quad \text{Eq. 1}$$

Where:

- $P_x(x), P_y(y), P_z(z)$: probability density functions of deviations in the longitudinal, lateral, and vertical directions, respectively.
- $\lambda_x, \lambda_y, \lambda_z$: closure rates between aircraft pairs in each dimension.
- S_y: nominal lateral separation between tracks.
- $P_y(S_y)$: probability that the lateral deviation brings the aircraft within the collision volume.

The model typically assumes Gaussian or Laplacian distributions for deviations, derived from empirical radar data.

3.2.1.1 Main features

- Assumes aircraft fly on parallel tracks at the same altitude.
- Uses statistical distributions for deviations.
- Conservative and analytically tractable.
- Requires empirical data for lateral and vertical deviations.

3.2.1.2 Typical use

The Reich model is particularly suited for structured, procedurally controlled airspace, such as the North Atlantic (NAT) oceanic region, where radar surveillance is limited and procedural separation is used [1] [5]. In this environment, aircraft follow parallel tracks at fixed altitudes, and radar surveillance is limited. The model estimates the probability of mid-air collisions by analysing deviations from nominal flight paths in the longitudinal, lateral, and vertical dimensions.

A historical application of the Reich model occurred during the evaluation of reducing lateral separation standards in the NAT region, from 120 nautical miles (NM) to 90 NM. The model treated aircraft as rectangular volumes and used empirical data to model deviations, typically assuming Laplacian or Gaussian distributions. The collision rate was calculated using the Eq. 1. The analysis revealed that reducing lateral separation to 90 NM increased the lateral collision risk by a factor of six compared to 120 NM. This led to the implementation of staggered separation schemes to mitigate the increased risk. This example illustrates how the Reich model supports regulatory decisions in structured airspace by quantifying the impact of changes in separation standards.

3.2.2 Gas Model (Endoh Model)

The Gas Model, introduced by Shinsuke Endoh, is inspired by kinetic gas theory. It models aircraft as particles moving randomly in a defined airspace volume and estimates collision frequency based on aircraft density, relative velocity, and protected volume [5].

In 2D, the expected number of collisions C per unit of time is:

$$C = \frac{N^2 g \cdot E[v_r]}{A} \quad \text{Eq. 2}$$

which in 3D becomes:

$$C = \frac{2N^2 ghE[v_r]}{AH} \quad \text{Eq. 3}$$

where:

- N: number of aircraft.
- A: horizontal area.
- H: vertical height.
- g: aircraft diameter.
- h: aircraft height.
- $E[\boldsymbol{v}_r]$ expected relative velocity.

Eq. 3 is valid provided $h \ll H$.

3.2.2.1 Main features

- Aircraft are modelled as randomly moving particles.
- Suitable for high-density, non-structured airspace.
- Can be extended to 3D, overtaking, and intersection scenarios.
- Allows estimation of both collision and conflict rates.

3.2.2.2 Typical use

The Gas Model, developed by Shinsuke Endoh, is well-suited for modelling unstructured, high-density airspace such as urban environments where unmanned aircraft operate. It treats aircraft as particles moving randomly in a defined volume and estimates collision frequency based on aircraft density, relative velocity, and protected volume. This model has been applied in several key contexts:

- Evaluating unmanned vs. manned aircraft interactions in Very Low Level (VLL) airspace, particularly in the SORA 2.0 framework [5].
- Estimating collision risk in Terminal Manoeuvring Areas (TMA) [6].

This model allows for the estimation of both collision and conflict rates and can be extended to include overtaking and intersection scenarios. If the calculated collision frequency exceeds the acceptable threshold (e.g., 1x10^{-7} collisions per flight hour), mitigation strategies such as altitude stratification, speed limits, or dynamic rerouting can be introduced. This example demonstrates how the Gas Model supports safety assessments and operational planning in dynamic, high-density environments like urban U-space corridors.

4 U-AGREE Collision risk models

The Endoh's collision risk model in section 3 illustrates the dependency of the collision risk on several key parameters:

- $C \propto N/AH$, i.e., the density of UAS in the airspace volume.
- $C \propto \pi g^2 h$, i.e., the UAS volume.
- $C \propto E[v_r]$, which in turn depends on the UAS speed and trajectories.

All those parameters are scenario dependent and so will be the collision risk. In [5], Endoh applied Eq. 3 to homogeneous encounters between Instrumental Flight Rules (IFR) flights in en-route operations and to combinations IFR and Visual Flight Rules (VFR) flights in a Terminal Manoeuvring Area (TMA).

However, the direct application of Eq. 3 to the fleet of UAS envisaged in U-space airspaces addressed by U-AGREE is not straightforward. Therefore, we will elaborate on the Endoh's approach in [5] to derive an extended gas model formulation that can be applied to compute analytically the collision risk for a fleet of UAS. This model will provide a first rough estimate of the collision risk, while more accurate figures would require numerical models that we will introduce later on.

4.1 Initial Collision risk model

4.1.1 Theoretical derivation

To extend Eq. 3 to address the diversity in UAS performance, size, and missions, we assume that UAS operating within any given volume of airspace can be categorized into M distinct classes, each characterized by specific attributes, i.e., aircraft in category i are represented by a cylinder of diameter g_i and height h_i, centred at the nominal aircraft position, as illustrated in Figure 1. UAS are randomly distributed in height following a uniform distribution and fly horizontal trajectories at a speed v_i, which can be modelled as a random variable uniformly distributed within the interval $[v_{i,min}, v_{i,max}]$ and random bearings (uniformly distributed within the interval $[0,2\pi]$ radians). UAS are not permitted to perform vertical displacements or collision avoidance manoeuvres. We also assume that there are N_i aircraft in each class such that $N = \sum_{i=1}^{M} N_i$, being N the total number of UAS flying within the volume, which has dimensions $A \times H$, where A is the area of the surface projected by the volume on the Earth's ellipsoid and H is its height (assumed constant).

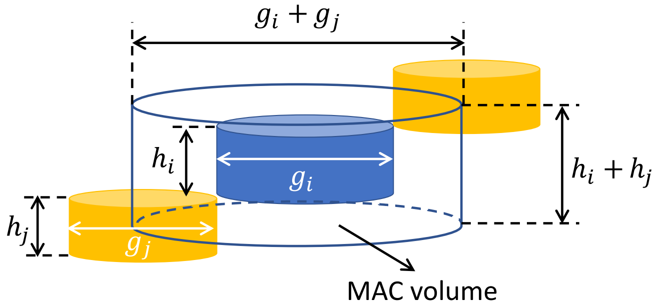


Figure 1. MAC volume between UAS of category i with UAS of category j.

EUROPEAN PARTNERSHIP
Co-funded by the European Union

In Figure 1, we depict a collision between one UAS of category i (represented by the blue cylinder) and UAS of category j (represented by orange cylinders). A collision between UAS of category i and j occurs when they overlap in both horizontal and vertical dimensions, forming a MAC volume around them.

Following the approach in [5], we express the collision rate as:

$$C = \sum_{i=1}^{M} \sum_{j=i}^{M} \mathcal{F}_{H_{i,j}} \cdot P_{V_{i,j}} \quad \textbf{Eq. 4}$$

where C stands for the rate of collisions per unit of time, $\mathcal{F}_{H_{i,j}}$ is the rate of horizontal overlaps per unit of time between category i and j UAS, and $P_{V_{i,j}}$ is the probability that a category i UAS vertically overlaps with a UAS of category j. Notice that the lower limit of the inner sum has been limited to i to avoid accounting for symmetric collisions.

The overlap condition is $|z_i - z_j| \leq \frac{h_i+h_j}{2}$, and the probability of overlap is given by:

$$P_{V_{i,j}} = \int_{z_0}^{z_0+H} f_z(z_i) \int_{-u_{i,j}}^{u_{i,j}} f_u(u) du \, dz \quad \textbf{Eq. 5}$$

where $u = |z_i - z_j|$ and $u_{i,j} = \frac{h_i+h_j}{2}$

Considering that the UAS are uniformly distributed between z_0 and $z_0 + H$:

$$f_z(z) = \begin{cases} \frac{1}{H} & \text{for } z_0 < z < z_0 + H \\ 0 & \text{otherwise.} \end{cases} \quad \textbf{Eq. 6}$$

while

$$f_u(u) = \frac{H - |u|}{H^2} \text{ for } -H < u < H \quad \textbf{Eq. 7}$$

Thus, we obtain:

$$P_{V_{i,j}} = \frac{h_i + h_j}{H} - \frac{(h_i + h_j)^2}{4H^2} \quad \textbf{Eq. 8}$$

assuming $h_i + h_j \leq 2H$. This expression can be further reduced to:

$$P_{V_{i,j}} = \frac{h_i + h_j}{H}$$

if $h_i + h_j \ll H$. While this approximation is generally applicable to manned aircraft, it is not universally applicable to UAS operations, especially when assessing collision risk below the VLL. Therefore, the full expression of $P_{V_{i,j}}$ in Eq. 8 will be retained in the remaining of the document and the applicability of the approximation will be assessed case by case.

To obtain $\mathcal{F}_{H_{i,j}}$ we compute the number of horizontal overlaps between category i and category j UAS in a very short time increment dt as:

$$A_{s_{i,j}} \cdot \rho_j \quad \textbf{Eq. 9}$$

being $A_{s_{i,j}}$ the shadowed area in Figure 2, which can be approximated by:

$$A_{s_{i,j}} = \frac{(g_i + g_j)}{2} E[v_{i,j}]dt \qquad \text{Eq. 10}$$

where $E[v_{i,j}]$ is the expected value of the relative velocities of category i and category j UAS and ρ_j is the horizontal density of category j UAS, given by:

$$\rho_j = \frac{N_j}{A} \qquad \text{Eq. 11}$$

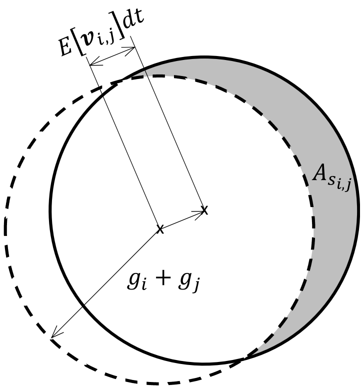


Figure 2. Horizontal overlap between a category i UAS and a category j UAS in dt.

Finally, considering that there are N_i UAS in category i, we compute the number of horizontal overlaps of category i UAS with category j ones in dt as:

$$\mathcal{F}_{H_{i,j}} = \frac{N_i}{2A}(g_i + g_j)E[v_{i,j}]N_j \qquad \text{Eq. 12}$$

However, if $i = j$, Eq. 12 needs to be modified to avoid counting the collision of one UAS of category i with itself:

$$\mathcal{F}_{H_{i,j}} = \frac{N_i(N_i - 1)}{A} g_i E[v_{i,i}] \qquad \text{Eq. 13}$$

By introducing Eq. 8 and Eq. 13 into Eq. 4 we get the overall collision rate per unit of time as:

$$C = \sum_{i=1}^{M}\sum_{j=i}^{M} \frac{N_i(N_j - \delta_{i,j})}{2A}(g_i + g_j)\left[\frac{h_i + h_j}{H} - \frac{(h_i + h_j)^2}{4H^2}\right] E[v_{i,j}] \qquad \text{Eq. 14}$$

In simpler terms, Eq. 14 estimates the number of collisions occurring per unit of time, i.e., C is typically expressed in hours collisions per hour. However, target levels of safety and their proxies are usually expressed in terms of occurrences per flight-hour. Hence, we need to manipulate Eq. 14 so that the result is expressed as collisions per flight hour. To this purpose, we compute the total number of hours flown (FH) in the time interval between t_1 and t_2 as:

$$FH = \int_{t_1}^{t_2} N(t)\, dt \qquad \text{Eq. 15}$$

where $N(t)$ is a piecewise distribution of integer numbers representing the number of UAS that are flying at time t ($t_1 \le t \le t_2$), so that $\Delta t = t_2 - t_1$. Considering a period in which N is constant, the number of flight hours is directly proportional to the elapsed time:

$$FH = N \cdot \Delta t \quad \textbf{Eq. 16}$$

where Δt is given in hours.

Hence, we compute the frequency of mid-air collisions in terms of flight hours experienced by each aircraft as:

$$\lambda_{UAS_{MAC}}[1/FH] = \frac{C[1/\Delta t]}{N} \quad \textbf{Eq. 17}$$

Therefore, we can express the unmitigated frequency of MAC as a function of the number of simultaneously operating aircraft as:

$$\lambda^0_{UAS_{MAC}}(N) = \frac{1}{2AN}\sum_{i=1}^{M}\sum_{j=i}^{M} N_i(N_j - \delta_{i,j})(g_i + g_j)\left[\frac{h_i + h_j}{H} - \frac{(h_i + h_j)^2}{4H^2}\right]E[v_{i,j}] \quad \textbf{Eq. 18}$$

where we use the super-index 0 simply to remark that the collision frequency given by Eq. 18 does not take into consideration any of the mitigation barrier set out in U-AGREE deliverable D.4.1 Initial barriers characterisation, either strategic or tactical (i.e., it is the unmitigated frequency of collision). It should be noted that the collision frequencies provided by Eq. 19 are expressed per unit time (i.e., per flight second). Therefore, they must be multiplied by 3,600 to obtain values per flight hour.

The expected relative speed can be further developed in light of the assumptions from this section. To this end, we first compute the relative speed $v_{i,j}$ from the dot product between the speed vectors of the UAS in i-th and j-th classes (use italic, bold fonts to represent vectors and italic fonts to represent scalar magnitudes):

$$v_{i,j} = \left[v_i^2 + v_j^2 - 2\cos\theta_{i,j}\right]^{1/2} \quad \textbf{Eq. 19}$$

where v_i, v_j and $\theta_{i,j}$ are uniformly distributed random variables with probability density functions given by (according to the assumptions in this section):

$$f_{v_k}(v_k) = \frac{1}{\Delta v_k}; \quad f_\theta(\theta) = \frac{1}{2\pi}; \quad \textbf{Eq. 20}$$

where $\Delta v_k = v_{k,max} - v_{k,mim}$. Therefore, the expected value of $v_{i,j}$ is given by:

$$E[v_{i,j}] = \int_{\Delta v_i}\int_{\Delta v_j}\int_{\mathbf{0}}^{2\pi}\left[v_i^2 + v_j^2 - 2\cos\theta_{i,j}\right]^{1/2} f_{v_i}(v_i) f_{v_j}(v_j) f_\theta(\theta)\, dv_i dv_j d\theta \quad \textbf{Eq. 21}$$

which, taking into consideration Eq. 20, leads to:

$$E[v_{i,j}] = \frac{1}{\Delta v_i}\frac{1}{\Delta v_j}\frac{1}{2\pi}\int_{v_{i,min}}^{v_{i,max}}\int_{v_{j,min}}^{v_{j,max}}\int_0^{2\pi}\left[v_i^2 + v_j^2 - 2\cos\theta_{i,j}\right]^{1/2} dv_i dv_j d\theta \quad \textbf{Eq. 22}$$

The integral in Eq. 22 has no closed solution, but it can be solved using numerical integration.

Although Eq. 18 does not take into consideration the uncertainty of the flight execution, usually represented by the Total System Error (TSE) jointly accounting for the System Navigation Error (NSE) and the Flight Technical Error (FTE), it provides rough analytical estimates of the expected collision rate in a volume of airspace depending on the specific traffic therein and projected demand. Furthermore, it provides a sound benchmark for the Monte Carlo based numerical implementation of the gas model, as detailed in section 4.2.

Notice that for geometrically homogenous fleets (i.e., $g_i = g_j, h_i = h_j \ \forall i, j$), Eq. 15 is equivalent to Eq. 3 provided $h_i, h_j \ll H$, since $E[v_r] = \frac{1}{N^2}\sum_{i=1}^{M}\sum_{j=i}^{M} N_i\left(N_j - \delta_{i,j}\right)E\left[v_{i,j}\right]$.

4.2 Final Collision risk model

4.2.1 Numerical implementation.

To estimate the collision frequency using the expanded gas model described in section 4.1.1 taking into consideration the flight uncertainty we have developed a simulation tool consisting of a 3D encounter generator. The generator runs random trajectories in a pre-defined volume and computes the unmitigated frequency of relevant accident precursors occurring amongst involved aircraft, according to the U-AGREE Accident-Incident Model, namely (in a scale of increasing proximity to the accident), encounter, near mid-air collisions (NMAC) and mid-air collisions (MAC). Trajectories are generated according to different policies and include errors that account for the flight execution uncertainty modelled using a combination of stochastic processes. The simulator uses the Monte Carlo method to obtain statistically relevant results.

The following data are required to configure each simulation run:

- Simulation volume scenario: a rectangular prism defined by the base's length (L) and width (W), both in kilometres, by minimum and maximum heights above the ground level, in meters and the number of layers.
- Position data: Horizontal and vertical Total System Error, accounting for both navigation systems and flight technical errors.
- The number of simultaneous flights.
- Total duration of the simulation, in seconds.
- The number of threads that will be used by the program.
- The timestep desired.
- The Protection Buffer size for strategic deconfliction.
- Selection of the trajectory simulation model.
- The recovery angle for simulations with conformance monitoring.
- The probability of failure for tactical (conformance monitoring) and strategic deconfliction mechanisms.
- The number of Monte Carlo trials.

To compute MACs, we consider cylinders centred on the aircraft's positions, representing their approximate size, according to the assumptions in section 4.1.1. We just consider a collision if the horizontal distance at any instant through the simulation is smaller than the sum of the sizes of the aircraft (horizontal radius) and the difference in their flight height is smaller than their vertical dimension. For near mid-air, we consider an sNMAC distance of 50 ft horizontally and 15 ft vertically for aircraft with less than 25 ft of maximum size [8] and an NMAC distance of 500 ft horizontally and 100 ft vertically [7] otherwise. In line with the U-AGREE encounter model, the encounter distance is 1 NM for UAS and 2 NM for manned aircraft.

The process of detecting MAC (Mid-Air Collision) is computationally intensive, as it requires running simulations at high traffic densities to ensure that the number of MAC occurrences is sufficient to guarantee the statistical validity of the results. To support large-scale simulations—such as those involving numerous simultaneous aircraft and/or extended flight durations—without demanding excessive computer memory, the toolbox avoids generating or storing non-essential data.

The tool also provides additional statistical indicators, among which there are simulation times and MAC, NMAC and encounter counts considering aircraft with passengers involved.

The final numerical implementation is composed of several interacting modules.

The configuration file is parsed by SimulationConfiguration. The simulator expects the following top-level structure:

- **identifier:** global name used to generate the final results file.
- **simulation_parameters:** contains the following mandatory fields:
    - simulation_time,
    - max_drones,
    - n_hilos,
    - n_sim,
    - min_dt,
    - max_dt,
    - protection_buffer
    - mode,

  and the following optional fields:
    - project,
    - dynamic_model, and
    - mix_mode.
- **strategic_deconfliction:** contains:
    - strategic_deconfliction_failure, and
    - max_strategic_deconfliction_attempts.
- **conformance_monitoring:** contains:
    - probability_failure_tse,
    - reaction_delay_k,
    - reaction_delay_theta,
    - reaction_delay_min,
    - communication_gamma_k,
    - communication_gamma_theta,
    - max_excursion_time, and
    - probability_failure_pbuffer.
- **categories:** a dictionary of aircraft classes. For each class the simulator reads:
    - mix,
    - tse_h,
    - tse_v,
    - excursion_frequency,
    - passengers,
    - performance.min_horizontal_speed,
    - performance.max_horizontal_speed, and

  - MAC, NMAC and encounter sizes.
- **linear_configuration:** required when the selected mode is not realistic. It defines:
  - area_length,
  - area_width,
  - h_min,
  - and h_max.
- **ornstein_uhlenbeck:** required when dynamic_model is OrnsteinUhlenbeck. It defines
  - theta_h, and
  - theta_v.

At the top level, each Monte Carlo simulation is controlled by UTC. This class owns the active fleet of drones, advances the global clock, inserts new aircraft, calls the conflict detector, removes finished flights, applies conformance monitoring, and finally uploads all statistics to ResultManager. The active simulation state is therefore not stored in a monolithic object but distributed across a small number of focused classes.

Main classes and responsibilities

- **SimulationConfiguration** parses the JSON file and exposes the simulation parameters, the category table, the conformance-monitoring parameters, the strategic-deconfliction parameters, the airspace geometry, and the selected dynamic model.
- **TrafficMix** selects the category of each new aircraft according to the declared mix. In insertion mode, the probability of category c_i is proportional to its mix value. In flight-hours mode, the selector compensates categories that are underrepresented by comparing the expected share with the accumulated flight time per category.
- **TrajectoriesGenerator** is an abstract interface. The code provides three implementations: LinearTrajectory, LocalRandomGenerator, and RealisticDBTrajectoryGenerator.
- Drone stores the complete state of one aircraft: its trajectory, current position, nominal position, velocity, dynamic model, integrity state, passengers, waypoint queue, excursion timers, and autopilot flag.
- **UTC is the simulation loop.** It runs one Monte Carlo trial, handles insertion and deletion, advances all active drones, and triggers the detectors and monitors.
- **UniformInserter** fills the airspace up to the maximum concurrent drone count. Before accepting a new trajectory, it applies strategic deconfliction against all currently active trajectories.
- **ConflictDetector** performs the online conflict check between all active aircraft at every time step. It classifies each pair into one of five severity levels: MAC, NMAC, and encounter.
- **Logger** optionally writes position and encounter logs. ResultManager aggregates the counters from all simulations and writes the final JSON file.

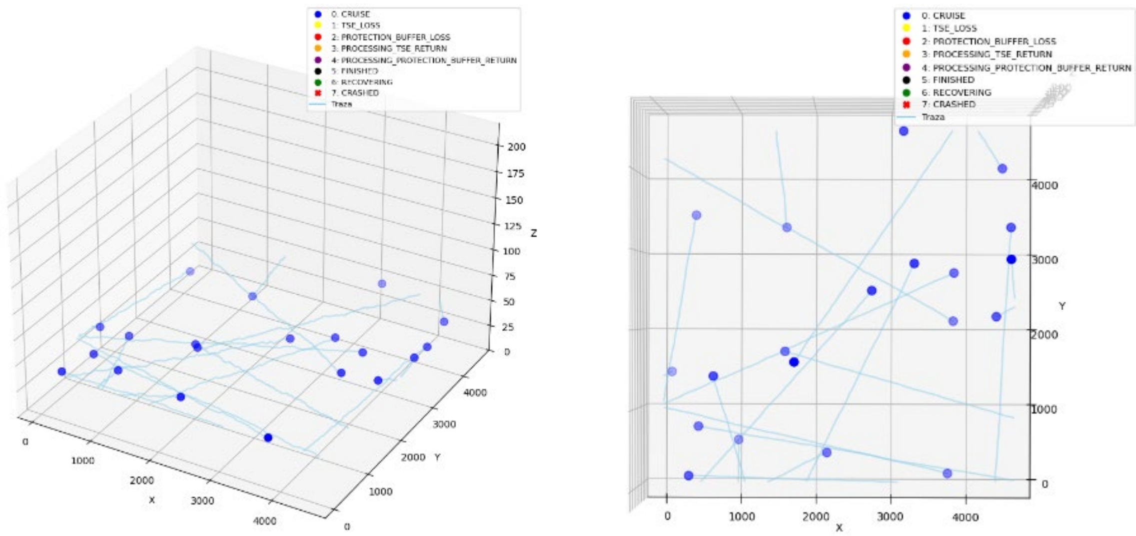
Trajectory generation

The simulator currently implements four trajectory generators.

1. **LinearTrajectory.** This generator produces a straight horizontal crossing inside a rectangular prism. A random entry side is selected, a random start point is drawn on that side, and the exit point is drawn on a different side. The aircraft flies at constant altitude, so the trajectory contains only two waypoints. The horizontal speed is sampled uniformly between the

minimum and maximum speed declared for the selected category. If $\boldsymbol{p}_0$ and $\boldsymbol{p}_1$ are the two waypoints, the nominal motion is piecewise linear, and the end time is computed as:

$$t_1 = t_0 + \frac{\|\boldsymbol{p}_1 - \boldsymbol{p}_0\|}{v_h} \quad \text{Eq. 23}$$



(a) Isometric view.

(b) Top view.

Figure 3. Random linear trajectories.

2. **LocalRandomGenerator.** This generator creates a more complex synthetic flight profile. It first generates a random take-off point at ground level, then a climb to a random altitude, then between 1 and 10 random cruise waypoints inside the area, and finally a descent back to the ground. Each segment is assigned a random horizontal speed within the category bounds. This generator is useful when one wants a richer local traffic pattern than a single crossing line.

EUROPEAN PARTNERSHIP
Co-funded by the European Union

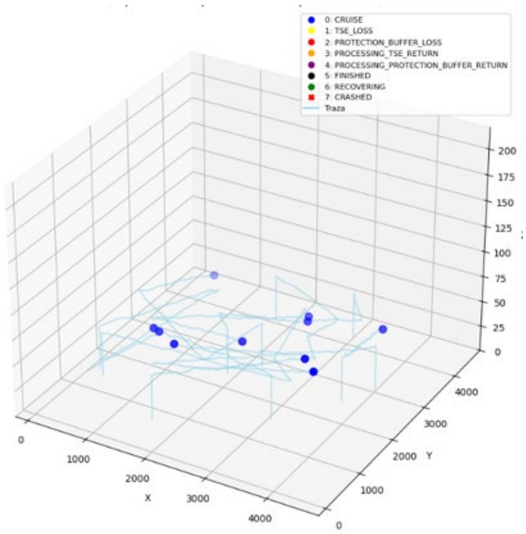


Figure 4. Local random trajectories (isometric view).

3. **RealisticDBTrajectoryGenerator.** This generator loads trajectories from an external JSON database. The trajectories are generated using a PX4/Gazebo simulation framework [11], providing a high level of fidelity relative to real UAS trajectories in terms of flight dynamics and control behaviour. Each trajectory record contains an id, a class, an aircraft type, and a list of geographic waypoints with time, latitude, longitude, and altitude (φ, λ, h). The coordinates are converted to a local tangent plane using a small-area approximation:

$$x = R\,(\lambda - \lambda_0)\,cos(\varphi_0) \quad \textbf{Eq. 24}$$

$$y = R\,(\varphi - \varphi_0) \quad \textbf{Eq. 25}$$

where R is the Earth radius, (φ_0, λ_0) are the coordinates of the projection centre, and all angles are expressed in radians. The generator then selects a trajectory at random from the category chosen by TrafficMix.

The nominal trajectory stored in each drone is always a piecewise-linear interpolation of the selected waypoints. For a time t inside the segment [tk, tk+1], the nominal position is:

$$\boldsymbol{p}_{nom}(t) = \boldsymbol{p}_k + \left(\frac{t - t_k}{t_{k+1} - t_k}\right)(\boldsymbol{p}_{k-1} - \boldsymbol{p}_k) \quad \textbf{Eq. 26}$$

and the nominal velocity is the constant segment slope.

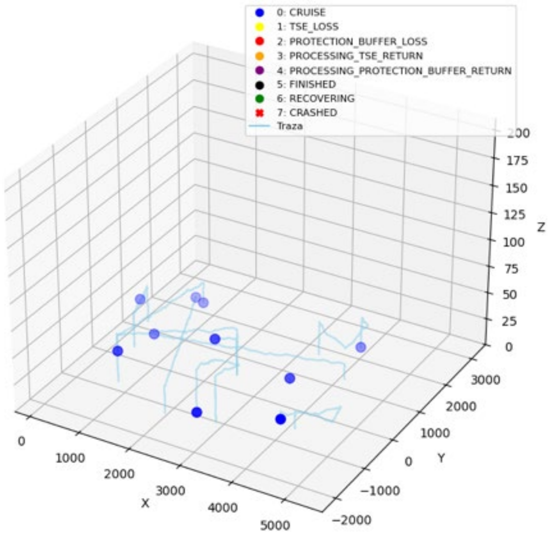


Figure 5. Realistic trajectories (isometric view).

4. **Excursion generator:** A drone in excursion (PROTECTION_BUFFER_LOSS) is represented in red. This type of trajectories is not corrected back into their nominal path and are free to evolve following a random walk process.

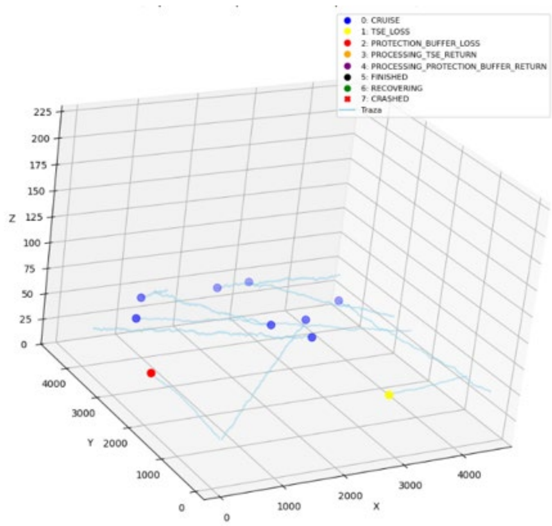


Figure 6. Excursions outside the 4D volume bounded by the TSE (in yellow) and outside the protection buffer (isometric view).

Traffic mix and insertion policy

TrafficMix preserves the declared category distribution on average. If the mix is a_i, then the nominal selection probability of category i is:

$$P(c_i) = \frac{a_i}{\sum_j a_j} \quad \textbf{Eq. 27}$$

Co-funded by the European Union

The simulator utilises two distinct methods to manage the traffic mix, differing in how they prioritise aircraft introduction.

- **Absolute Insertion** (Insertion): Aircraft are inserted based on the numerical proportion defined in the mix. This maintains a fixed ratio of total airframes but ignores differences in mission duration.
- **Flight-Hour Adjusted** (fh): Aircraft are inserted to match the proportion of total flight time per category. This method compensates for categories with shorter missions by inserting them more frequently, ensuring the simulated exposure time aligns with the target mix.
- **UniformInserter** (Inserter) is responsible for creating new drones. At each simulation step it tries to fill the active fleet up to max_drones. For each candidate trajectory, the code first samples a category through TrafficMix, then asks the chosen trajectory generator for a trajectory starting at the current time.

When a drone is accepted, the inserter creates its dynamic model according to simulation_parameters.dynamic_model. Two options are currently used in the main executable:

- **Gaussian:** a memoryless position error model.
- **Ornstein-Uhlenbeck:** a correlated, mean-reverting position error model.

The inserter also schedules excursion events probabilistically through the category field excursion_frequency. When the expected number of excursions, computed from the accumulated flight hours, exceeds the number already assigned to the category, a future excursion start time is programmed for the drone.

Drone state and stored data

Each Drone object stores both static and dynamic information. The static part includes the drone identifier, the aircraft category, the selected trajectory, and the passenger count. The dynamic part includes:

- the deque of remaining waypoints;
- the nominal position at the current simulation time;
- the real position produced by the dynamic model;
- the current velocity vector;
- the selected dynamic model;
- the current integrity state;
- the pilot delay used by conformance monitoring;
- the programmed excursion start time;
- the time at which the integrity loss started;
- flags indicating whether the drone has exceeded the protection buffer, has been rerouted, or has autopilot enabled.

The class provides methods to advance the aircraft, query the current position, query the remaining waypoints, read the trajectory, mark the drone as finished, and switch the autopilot on or off. In other words, Drone is the local state machine of one aircraft.

Dynamic models

Co-funded by the European Union

The simulator implements two operational dynamic models in the main executable.

- **Gaussian model.** This model assumes independent zero-mean Gaussian errors in x, y, and z. The standard deviation is derived from the total system error TSE assuming a 95% containment:

$$\sigma_h = TSE_h / 1.96 \quad \textbf{Eq. 28-a}$$

$$\sigma_v = TSE_v / 1.96 \quad \textbf{Eq. 28-b}$$

The real position is then:

$$\boldsymbol{p}_{real}(t) = \boldsymbol{p}_{nom}(t) + \boldsymbol{e}(t) \quad \textbf{Eq. 29}$$

with e sampled independently at each step. This model is simple, memoryless, and appropriate when the goal is to estimate purely instantaneous spread around the planned path.

- **Ornstein-Uhlenbeck** model. This is the default correlated model when a non-Gaussian dynamic behaviour is required. The model evolves a zero-mean error process $\boldsymbol{e}(t)$ with mean reversion toward the nominal path. In discrete time, for a time step dt, the implementation uses:

$$\alpha_h = e^{-\theta_h dt} \quad \textbf{Eq. 30-a}$$

$$\alpha_v = e^{-\theta_v dt} \quad \textbf{Eq. 30-b}$$

$$e_{x(k+1)} = \alpha_h e_{x(k)} + \sigma_h \sqrt{1 - \alpha_h^2}\eta_x \quad \textbf{Eq. 31-a}$$

$$e_{y(k+1)} = \alpha_h e_{y(k)} + \sigma_h \sqrt{1 - \alpha_h^2}\eta_y \quad \textbf{Eq. 31-b}$$

$$e_{z(k+1)} = \alpha_v e_{z(k)} + \sigma_v \sqrt{1 - \alpha_v^2}\eta_z \quad \textbf{Eq. 31-c}$$

where η_x, η_y, η_z are independent standard normal variables. The parameters θ_h and θ_v control how quickly the error is pulled back toward zero: larger theta means stronger mean reversion and shorter correlation time. The simulator uses the same TSE-to-sigma conversion as in the Gaussian case so that both models can be compared on a common scale.

During recovery, the code reduces the effective alpha factor by a factor of 2, which makes the Ornstein-Uhlenbeck correction stronger and accelerates convergence back to the nominal path.

Excursion handling is implemented as a supervisory state transition rather than as a separate production trajectory generator. The current executable does not instantiate the dedicated ExcursionModel in the main loop; instead, the conformance-monitoring logic disables the autopilot when the programmed excursion time is reached and keeps tracking the aircraft through the selected stochastic error model. This is the behaviour that should be considered when interpreting the baseline collision-risk figures.

Accident precursors

The conflict detector evaluates all unordered aircraft pairs at every step, so the cost grows quadratically with the number of active drones. For a pair i, j, the horizontal and vertical separations are

$$d_h = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad \textbf{Eq. 32-a}$$

$$d_v = |z_i - z_j| \quad \textbf{Eq. 32-b}$$

Each category stores a set of horizontal and vertical thresholds for the five severity levels. The detector first applies a permissive outer filter using the encounter thresholds. If the pair is farther than the sum of the encounter radii of the two categories, no finer test is needed.

The severity ladder is ordered so that smaller values mean more severe conflicts:

- $0 = MAC$
- $1 = NMAC$
- $2 = encounter$
- $3 = no\ conflict$

For MAC, the detector uses a symmetric combined threshold built from both aircraft categories. For the remaining symmetric levels, it uses the maximum category threshold. This means that MAC and NMAC are symmetric events at pair level: if one aircraft is inside the severity region, the pair is considered in that state. By contrast, encounter is directional. Aircraft i may consider j to be in encounter state even if j does not yet consider i to be inside its own encounter bubble. This is why the code stores directional counts for encounters.

The pairwise severity matrix is updated whenever the pair moves to a more severe state. If the severity reaches MAC, both drones are marked as finished and the crash positions are logged. If one of the aircraft is in excursion, the MAC is counted separately in the excursion statistics.

How drones are removed

A drone leaves the active set in three situations.

1. It reaches the end of its waypoint list. In that case Drone::next_waypoint marks it as FINISHED.
2. It is involved in a MAC. The conflict detector marks both aircraft as finished and logs the crash.
3. Conformance monitoring forces termination. This happens if the aircraft exceeds the protection-buffer limits and remains beyond the recovery window for longer than max_excursion_time, or if the excursion state cannot be recovered in time.

The physical removal from the active vector happens in UTC::removeFinishedFlights, which also updates the per-category flight-time counters. This separation between logical termination and physical removal keeps the simulation loop simple and avoids invalidating iterators during the main advancement step.

Co-funded by the European Union

Logged data and final results

If the -l option is enabled, the simulator writes a CSV file in position_logs with one record per time step and per aircraft. Each record stores timestamp, aircraft id, x, y, z, and the current integrity code. If the -e option is enabled, the simulator also writes binary encounter files in encounter_logs. For each encounter, the logger stores the encounter id, the timestamp, the ownship and intruder states, their positions and velocities, and the remaining waypoints of both aircraft.

The final JSON result file contains, for each simulation, the total flight count, the rejected insertions, the flight hours per category, the number of excursions per category, the number of excursions beyond the protection buffer, the rejection counts, the total MAC/NMAC/encounter counts, and the directional conflict breakdown by aircraft type.

For the baseline figures in this section, the relevant quantity is the empirical frequency of MAC/NMAC normalized by the total simulated flight hours:

$$f_{MAC} = \frac{N_{MAC}}{FH_{tot}} \quad \text{Eq. 33-a}$$

$$f_{NMAC} = \frac{N_{NMAC}}{FH_{tot}} \quad \text{Eq. 33-b}$$

$$f_{enc} = \frac{N_{enc}}{FH_{tot}} \quad \text{Eq. 33-c}$$

where N_{MAC}, N_{NMAC} and N_{enc} are the total counts accumulated over all Monte Carlo trials and H_{tot} is the total amount of simulated flight hours.

Limitations of the model

The collision-risk estimates should be interpreted as geometric and stochastic estimates rather than as a complete operational safety assessment. The dominant source of uncertainty is the dynamic model assigned to each category: if the TSE, the OU correlation times, or the traffic pattern are changed, the resulting MAC and NMAC frequencies can change substantially. The risk output also depends strongly on the chosen trajectory generator, because linear, synthetic random, and realistic database trajectories produce very different encounter structures.

In addition, the conflict detector uses cylindrical protection envelopes and piecewise-linear motion. It does not model wake turbulence, aircraft attitude, detailed aerodynamic constraints, or reactive pilot behaviour beyond the conformance-monitoring state machine. The strategic conflict resolution mechanism is conservative but only acts before insertion, so it cannot resolve conflicts that arise later because of stochastic deviations. Finally, the online pairwise conflict check is $O(N^2)$, in the number of active aircraft, which makes dense scenarios computationally expensive.

4.2.2 Approach to the computation of the unmitigated collision frequency.

We designed and implemented the tool described in section 4.2.1 to efficiently estimate collision risk using representative UAS trajectories. Despite its parallel implementation, the computational cost increases with the number of simultaneous UAS, leading to unacceptably high computation times.

However, it is possible to keep the number of simultaneous UAS relatively low while still obtaining statistically valid results. Indeed, a detailed inspection of Eq. 18 shows that the numerator is $O(N^2)$, while the denominator is $O(N)$. Therefore, it is possible to rearrange Eq. 18 as:

$$\lambda^0_{UAS_{MAC}}(N) = N \cdot P_{MAC} \quad \textbf{Eq. 34}$$

where:

$$P_{MAC} = \frac{1}{2AN^2} \sum_{i=1}^{M} \sum_{j=i}^{M} N_i (N_j - \delta_{i,j})(g_i + g_j) \left[\frac{h_i + h_j}{H} - \frac{(h_i + h_j)^2}{4H^2} \right] E[v_{i,j}] \quad \textbf{Eq. 35}$$

Note that now both numerator and denominator in Eq. 35 are $O(N^2)$, thus P_{MAC} will not depend on the number of aircraft operating in the airspace. On the other hand, P_{MAC} will depend on the relative size of the aircraft operating in the volume (namely, g_i, h_i) and their relative speeds (i.e., $v_{i,j}$), which, in turn, depend on the traffic mix (i.e., N_i/N). Therefore, assuming that A and H are constant, the traffic mix is fixed, and the traffic flows are stable (so that $E[v_{i,\boldsymbol{j}}]$ is constant), we can identify P_{MAC} as a scenario-related parameter, namely, the component of the unmitigated probability of MAC that depends on the traffic mix operating in the UTM airspace.

Therefore, to determine the scenario-dependent parameter P_{MAC} for a given airspace with a specific size and the traffic mix, we can conduct simulations and record the total number of mid-air collisions during the simulation time to compute $\lambda^0_{UAS_{MAC}}$. Using Eq. 34 with the number of UAS used in that scenario, we can obtain:

$$P_{MAC} = \frac{\lambda^0_{UAS_{MAC}}}{N} \quad \textbf{Eq. 36}$$

Once P_{MAC} is calculated for a particular scenario, we use Eq. 34 to compute $\lambda^0_{UAS_{MAC}}(N)$ for any value of operating aircraft (N) in that scenario.

However, when the collision frequency is obtained through Monte Carlo experiments, Eq. 34 needs to be rewritten as:

$$\lambda^0_{UAS_{MAC}}(N) = (N-1) \cdot P_{MAC} \quad \textbf{Eq. 37}$$

because in that case we are estimating the collision frequency from the number of pairwise collisions (see [5] for details).

Expressions similar to Eq. 34 and Eq. 37 can be used to compute NMAC and encounter frequencies both analytically or numerically.

4.3 Simulation results and sensitivity assessment

4.3.1 Analytical results

To validate the extended Endoh model, different versions of the original Endoh model were compared to corroborate that both models obtain the same results.

For these validations, the collision volume selected represents the MAC, with a distance of 30 m. This same value was applied to both horizontal (i.e. g) and vertical (i.e. h) distances, in accordance with the definitions provided in Figure 1, Section 4.1.1.

The size of the area selected is 4.61x4.61 kms.

Extended Endoh vs Endoh – 2D

This initial validation assumes that the model operates in a two-dimensional (2D) scenario (Eq. 2). To recreate this scenario, it was assumed that there is only one category of UAS, with both a minimum and maximum speed of 10 m/s. The direction of the velocity is uniformly distributed over 0 and 2π rad.

To simulate a 2D scenario, h (MAC vertical distance) equals H (height of the U-space); this way, it is assumed that the UAS would be equally distributed in height. Applying all these assumptions, the results obtained are the following:

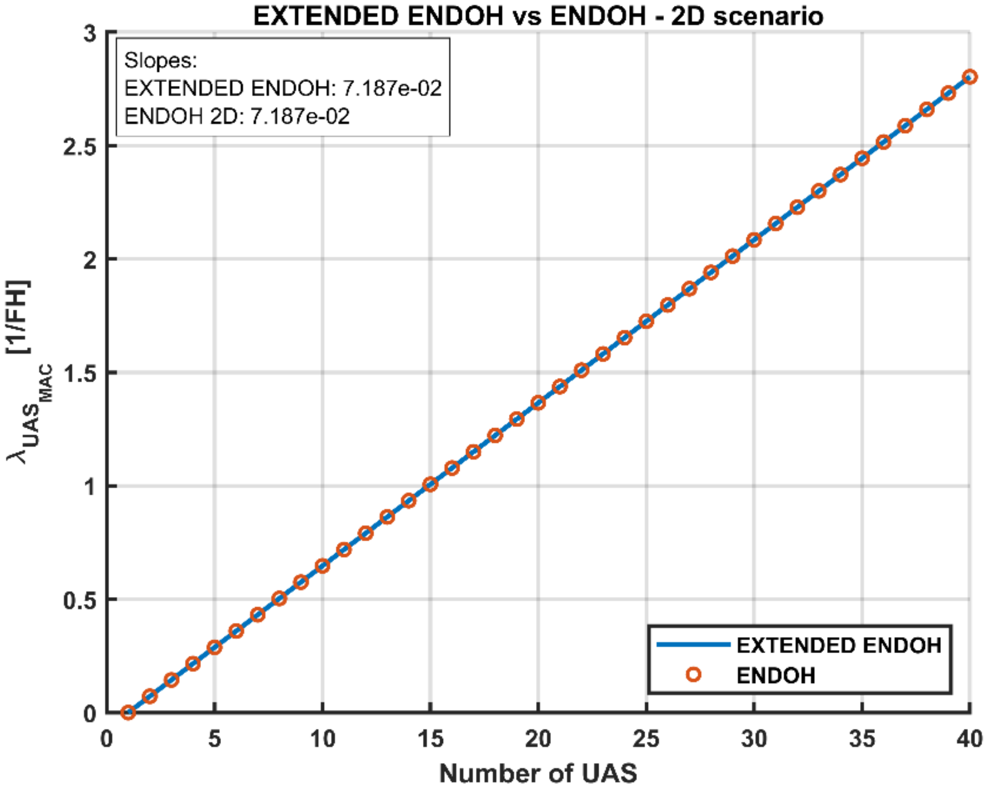


Figure 7. Extended Endoh vs Endoh - 2D scenario.

Figure 7 shows that both the extended model and the Endoh model produce consistent results when applied under the same assumptions.

Extended Endoh vs Endoh – 3D

Following a similar approach to the previous validation, only one UAS category was considered, with a minimum and maximum speed of 10 m/s. The direction of the velocity is also uniformly distributed over 0 and 2π rad.

Contrary to the previous case, this scenario represents a 3D scenario (Eq. 3 of Endoh) in which the H (height of the U-space) is bigger than h (size of the UAS). In this case, the selected H was 5000 m. The obtained results are presented in Figure 8.

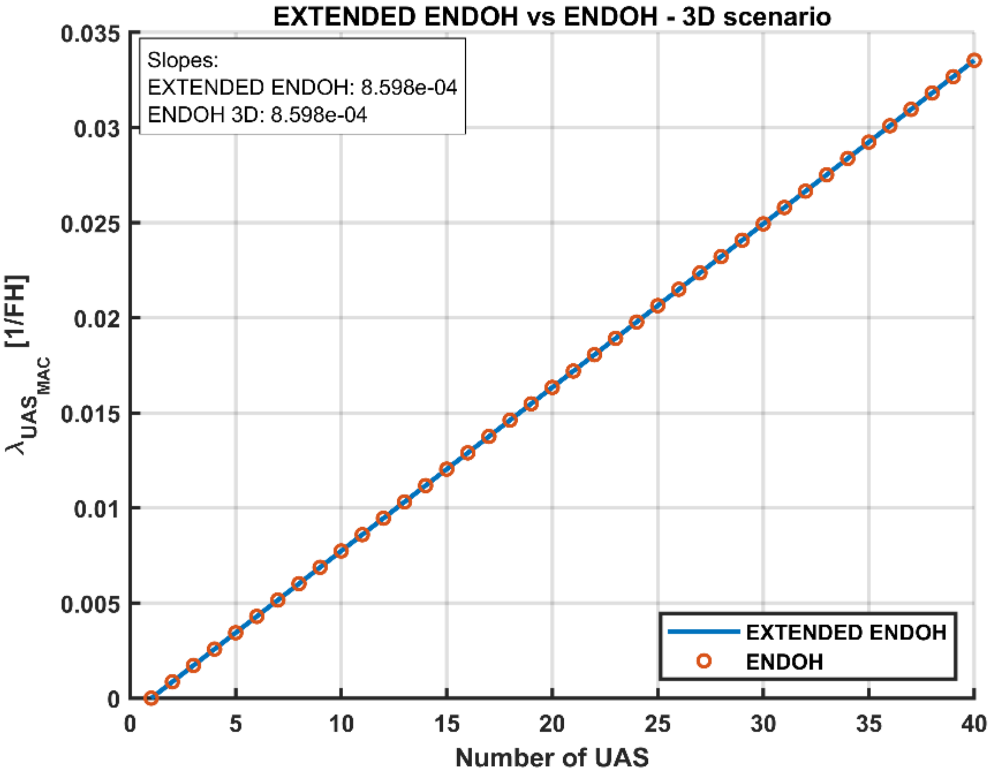


Figure 8. Extended Endoh vs Endoh - 3D scenario.

Figure 8 shows a similar tendency to the previous analysis; both models obtain the same results under the same assumptions.

Extended Endoh vs Endoh – 3D varying velocity

In this validation the selected value of H was 5000 m. Since H>>h, both Endoh models were simplified accordingly. This example considers two categories of UAS with the same size and different velocity ranges. One category has speeds between 10 m/s and 20 m/s, while the other ranges from 25 m/s to 35 m/s, with velocity being also uniformly distributed over 0 and 2π rad. Each of the categories represent 50% of the traffic.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

The results obtained for this case are the following:

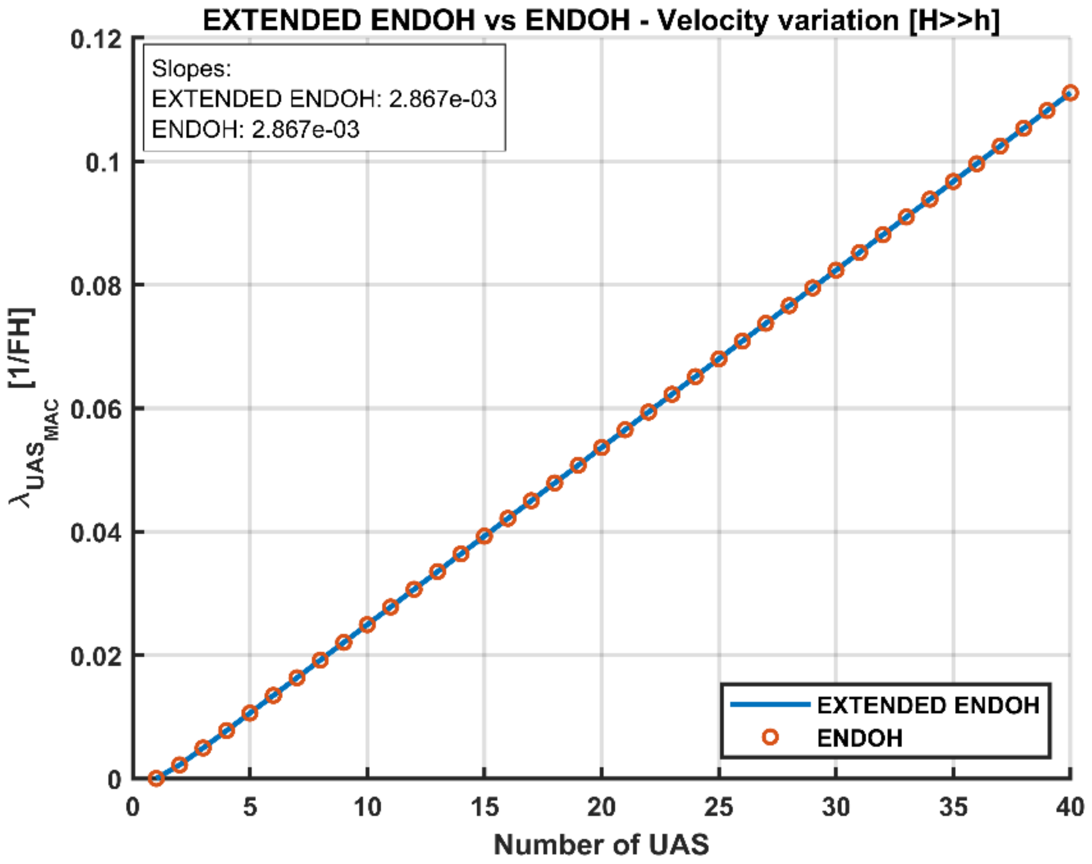


Figure 9. Extended Endoh vs Endoh - Scenario varying velocity.

As demonstrated in both previous cases, Figure 9 indicates that both models yield identical results; therefore, the Extended model has been validated against the Endoh model.

4.3.2 Simulated results.

4.3.2.1 Simulated results sensitivity analysis

Time step variation:

For the sake of simplification of the simulations, the sensitivity analysis for the time step was performed under some general assumptions:

- The collision volume selected represents the MAC, with a distance of 30 m. This same value was applied to both horizontal (i.e. g) and vertical (i.e. h) distances, in accordance with the definitions provided in Figure 1, Section 4.1.1.
- There is only one category of UAS, with both a minimum and maximum speed of 10 m/s. The direction of the velocity is uniformly distributed over 0 and 2π rad.
- The size of the area selected is 4.61x4.61 kms and H (height of the scenario) is 500 m.
- Flight hours are constant per category and simulation.

- Trajectories are generated with the **LinearTrajectory** generator explained in the previous section.

The simulations were performed under those assumptions, and the time steps (DT) selected were 2 s, 1 s, 0.5 s, 0.05 s and 0.01 s. The results obtained are presented in Figure 10.

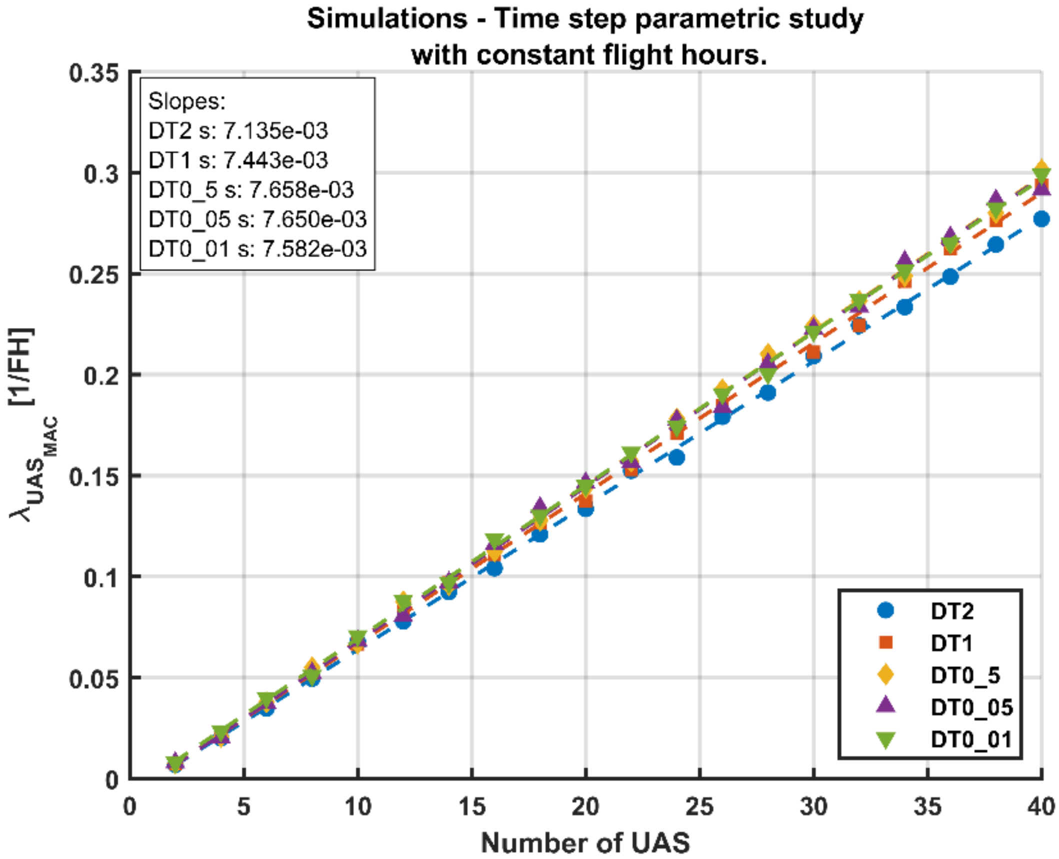


Figure 10. Simulation results - Time step parametric study.

The results indicate that when starting at large time step (e.g. DT 2 s) compared to the speed to the UAS results in a lower MAC frequency is lower than using smaller time steps (e.g. DT <=0.5 s)

Flight hours variation:

Following a similar approach to the previous parametric step, this sensitivity analysis for the flight hours per each UAS was performed under some general assumptions:

- The collision volume selected represents the MAC, with a distance of 30 m. This same value was applied to both horizontal (i.e. g) and vertical (i.e. h) distances, in accordance with the definitions provided in Figure 1, Section 4.1.1.
- There is only one category of UAS, with both a minimum and maximum speed of 10 m/s. The direction of the velocity is uniformly distributed over 0 and 2π rad.
- The size of the area selected is 4.61x4.61 kms and H (height of the scenario) is 500 m.
- Time step is constant set at DT 0.5 s.

- Trajectories are generated with the **LinearTrajectory** generator explained in the previous section.

The variation of flight hours per UAS is 100 fh, 500 fh and 2000 fh. The results obtained are the following:

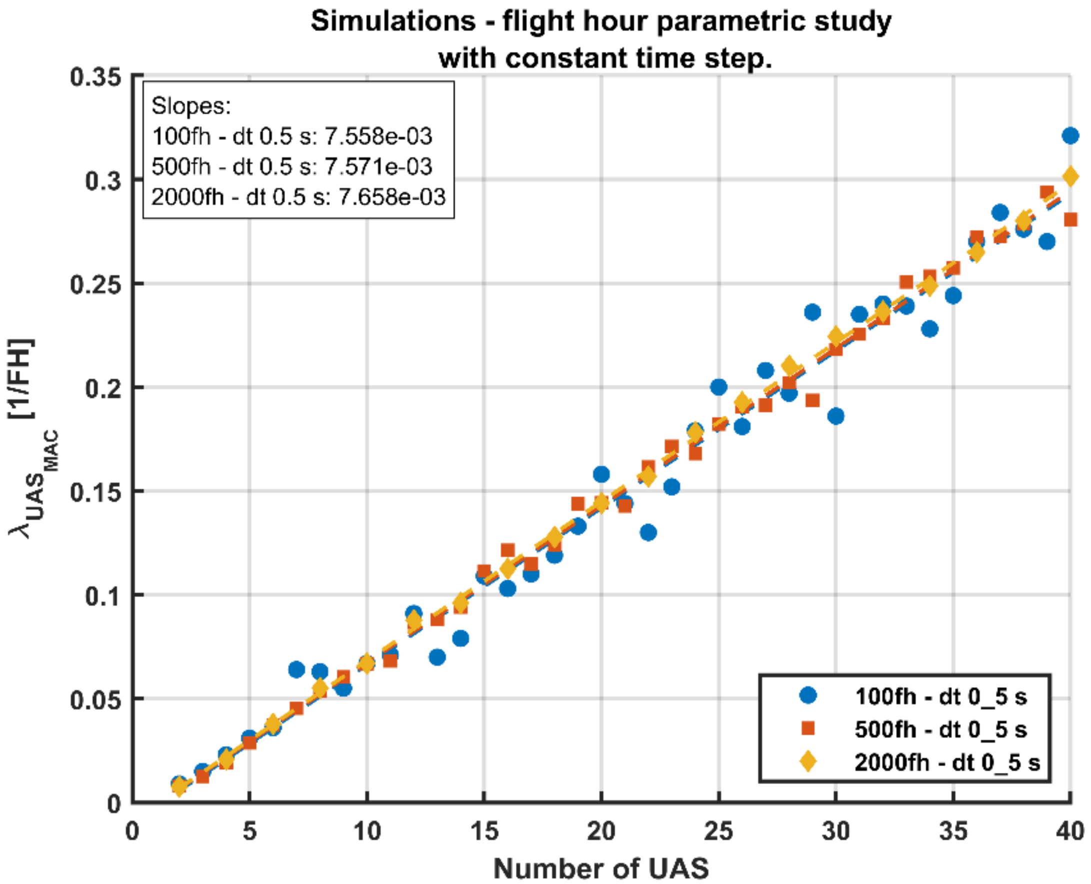


Figure 11. Simulation results - Flight hours parametric study.

Figure 11 shows that applying more flight hours per UAS results in points being more similar to the linear fit of the results.

Endoh extended vs simulation comparison

To compare the results from the simulations with the extended Endoh model for U-space, the following traffic mix was selected:

Category	Percentage (%)	g_i (m)	h_i (m)	v_min (m/s)	v_max (m/s)
Certified_Unknown	14.69	15.24	4.57	30	90
OpenA1_MR	2.91	15.24	4.57	3	13

OpenA2_MR	4.85	15.24	4.57	5	20
OpenA3_MR	24.23	15.24	4.57	10	21
PDRA/STS_MR	19.39	15.24	4.57	15	23
SpecificSAILI-II_MR	33.93	15.24	4.57	10	19

Table 3: Traffic mix for a given scenario.

The results obtained by simulation are:

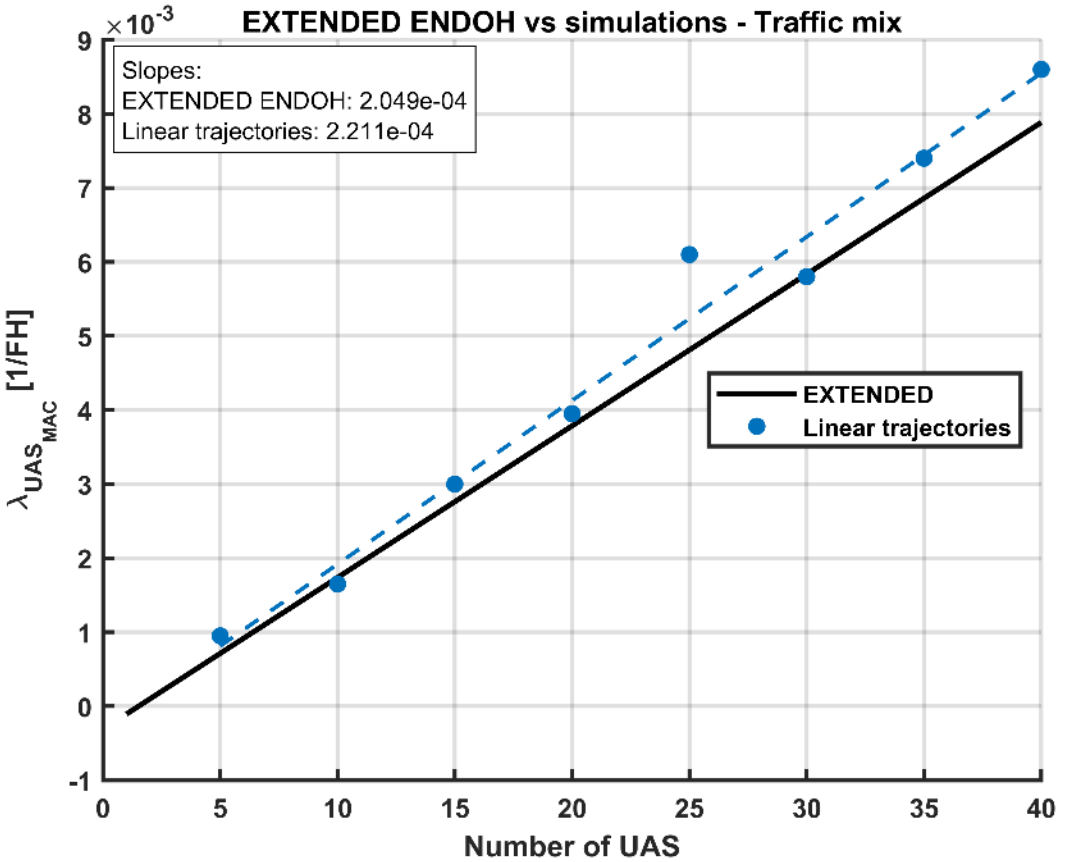


Figure 12. Extended Endoh vs simulation results with a given traffic mix.

Figure 12 shows the results obtained by simulation are similar to the extended Endoh model.

Dynamic models: Gaussian vs Ornstein-Uhlenbeck model

Another possible parameter that could affect the results is the dynamic models used to generate the trajectories.

This test assumed a given traffic mix of UAS:

Category	Percentage (%)	g_i (m)	h_i (m)	v_min (m/s)	v_max (m/s)
Certified_Unknown	8.16	15.24	4.57	30	90
OpenA1	26.11	15.24	4.57	3	13
OpenA2	15.3	15.24	4.57	5	20
OpenA3	15.3	15.24	4.57	10	21
PDRA/STS	20.41	15.24	4.57	15	23
SpecificSAILI-II	10.2	15.24	4.57	10	19

Table 4: Traffic mix for a dynamic models study.

The results obtained by simulation are:

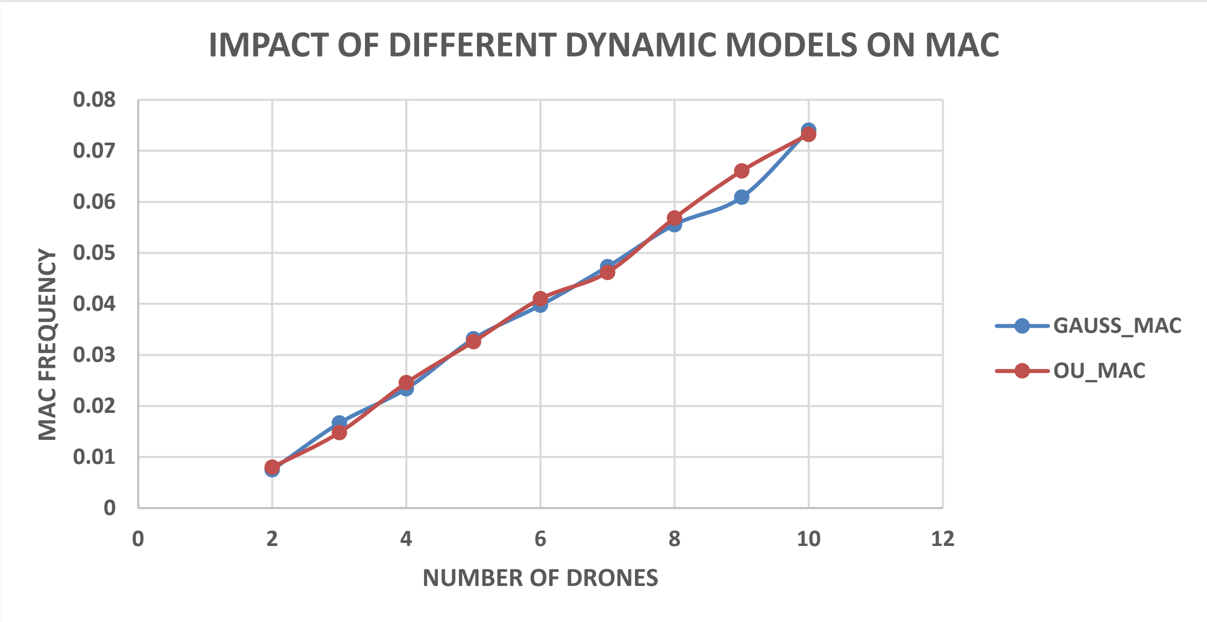


Figure 13. Impact of different dynamic models – MAC.

Co-funded by the European Union

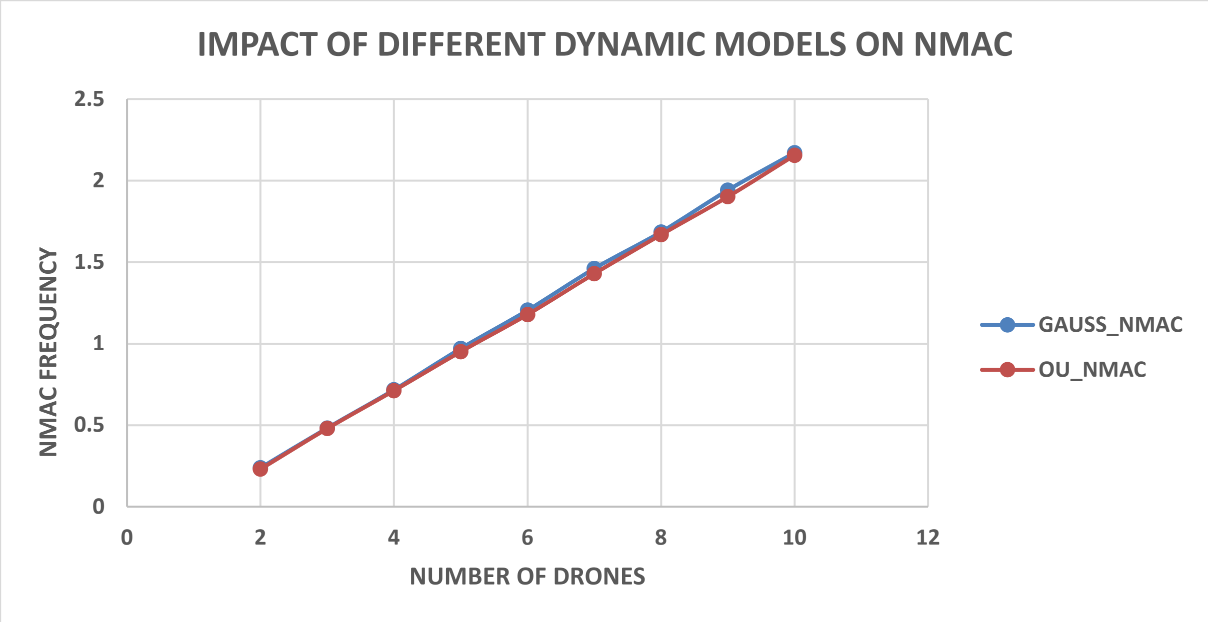


Figure 14. Impact of different dynamic models - NMAC.

Trajectory insertion mix: absolute number vs flight hour monitoring

The discrepancy in detected collisions between methods is driven by the operational profile of each category. In the Absolute method, categories with longer trajectories remain in the airspace longer, disproportionately increasing their "presence" and the resulting conflict frequency. The Flight-Hour method corrects this by ensuring the airspace occupancy reflects the intended temporal distribution.

In a small scenario, the higher conflict count in the Flight-Hour mode suggests that risk-prone categories were underrepresented in time under the Absolute method.

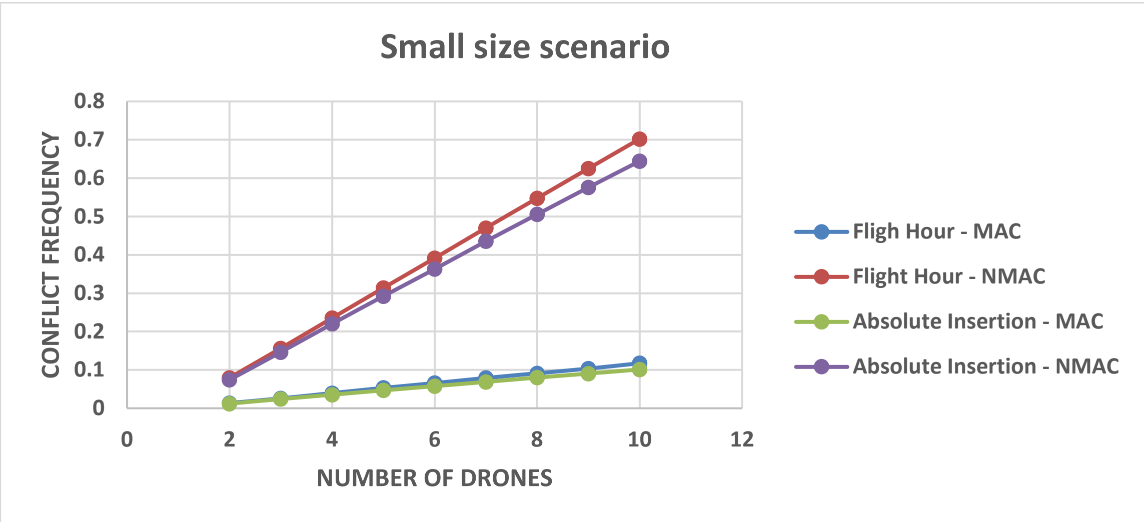


Figure 15. Impact on the trajectory insertion mode - Small size scenario.

In a bigger size scenario, the negligible difference indicates that mission durations are relatively uniform across categories with relevant presence in the mix, making the two insertion logic types functionally equivalent.

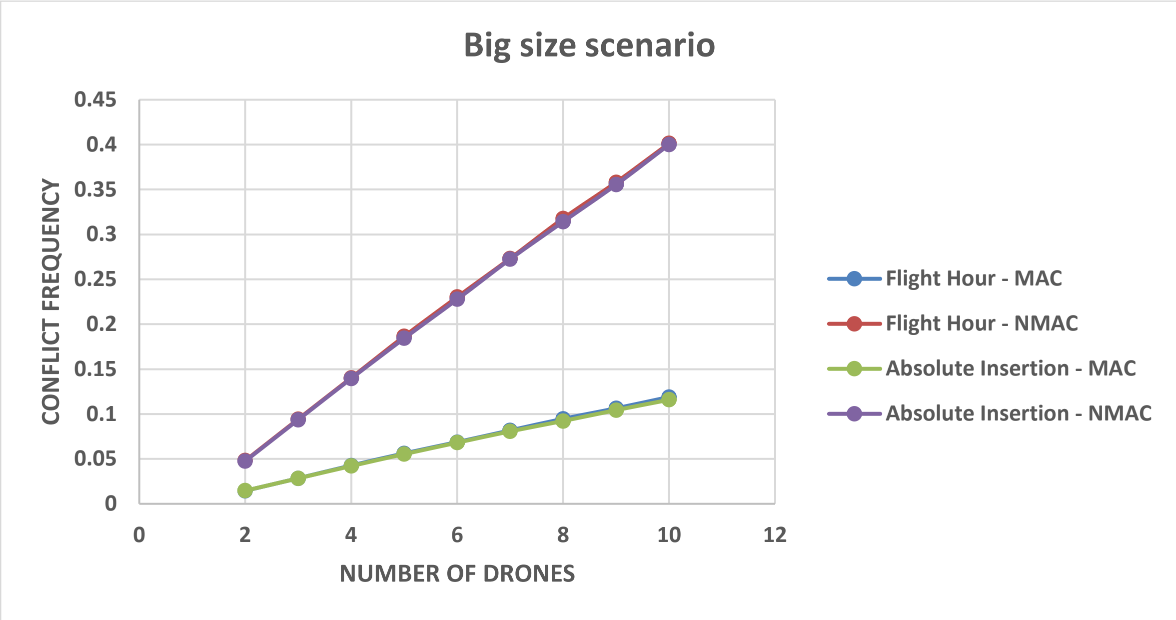


Figure 16. Impact on the trajectory insertion mode - Big size scenario.

Trajectory type: Realistic vs linear

The simulator can generate linear trajectories (uniformly distributed as defined in Endoh model) or inject realistic traffic from a previously generated trajectory. In this sensitivity analysis, the impact on trajectory type is analysed.

The traffic mix used for this analysis is the shown in Table 3 .

Co-funded by the European Union

The results obtained are the following:

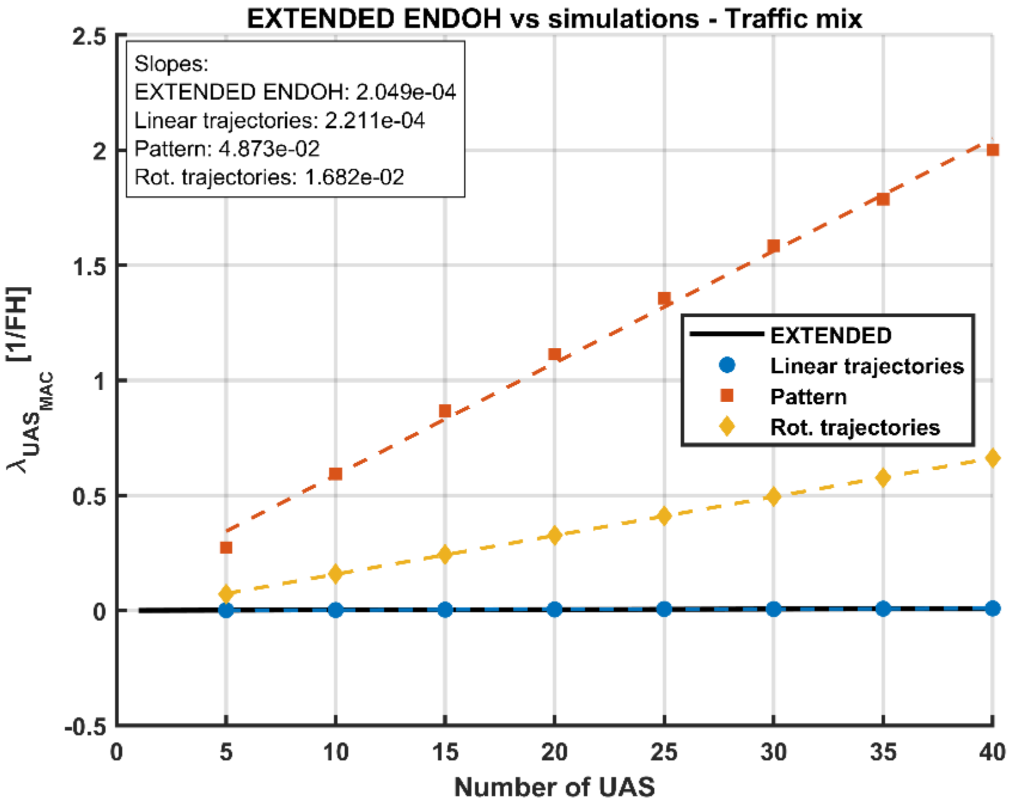


Figure 17. Extended endoh vs simulations - Trajectory type.

Given a dataset of realistic trajectories for a specific scenario, the difference between having the real pattern and rotating the trajectories is presented in the following figures:

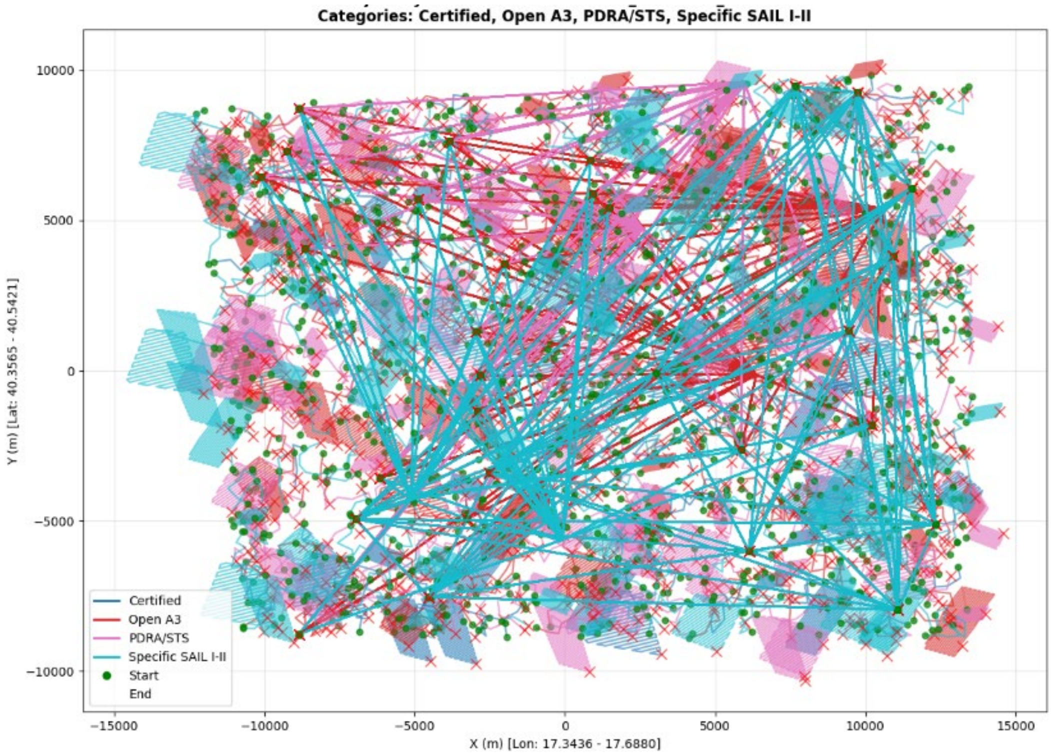


Figure 18. Trajectory pattern in a given scenario.

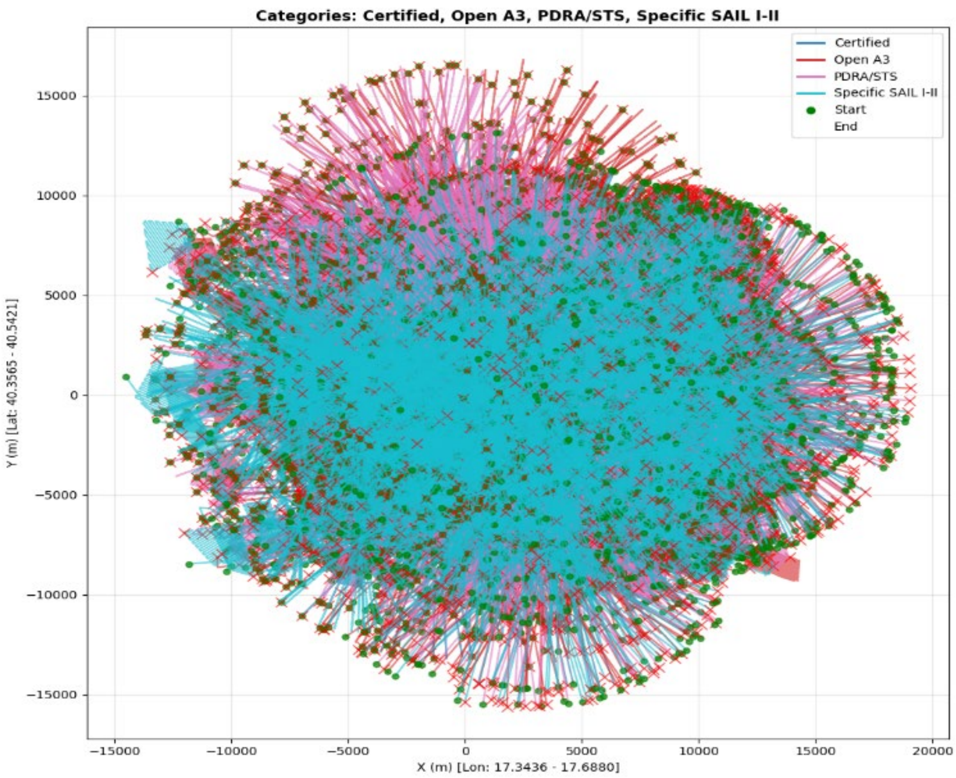


Figure 19. Trajectory pattern rotated in a given scenario.

4.3.3 Conclusions

The slight divergence between the analytical model and the Monte Carlo simulations at higher traffic densities in Figure 12 (~8%) can be explained by the assumptions underlying the Endoh formulation. The extended Endoh model is based on a gas-kinetic analogy, in which aircraft are assumed to be uniformly distributed and to interact through statistically independent pairwise encounters [5]. Under these assumptions, the collision rate is proportional to the product of aircraft densities, consistent with classical collision theory, where interaction frequencies scale with the square of the number density [12]. However, previous studies on air traffic collision risk modelling have highlighted that, particularly at higher traffic densities, interactions between aircraft are not strictly independent, and system-level effects emerge due to the complex interaction of multiple agents [12]. Furthermore, encounter modelling approaches based on recorded or simulated trajectories have shown that higher-order encounter configurations and correlations between aircraft can significantly influence the effective collision rate [13]. In Monte Carlo simulations, the estimation is implicitly conditioned on realised encounters, which introduces a bias towards geometries with higher relative velocities or longer interaction times, as observed in trajectory-based collision risk studies [14], [15]. These second-order effects, which are not captured in the analytical formulation, explain the slight but systematic underestimation of collision frequency by the extended Endoh model as the number of UAS increases.

Figure 10 illustrates the effect of the time step on the collision frequency. The value of this parameter should be fixed pursuing a balance between the computation time and the accuracy of the results. Considering the UAS speed of 10 m/s and a MAC size of 30 m, we can establish a relationship where the MAC frequency is independent of the simulation time step. For example, if we use a time step of 1 second (DT = 1 s), each UAS would advance 10 m with each iteration. This could result in missing a MAC event between two UAS during simulations, as the MAC distance is applied with a radius of 15 m. In contrast, if we use a time step of 0.5 seconds (DT = 0.5 s), each UAS would advance only 5 m, minimising the chance of missing a potential MAC event. Thus, we conclude that the relation is DT*speed < MAC_size. For this particular example, reducing the timestep below 0.5 seconds yields no relevant improvement while the computation time can be increased more than 10 times. The exact value will depend on a trade-off between the size and the speed of the aircraft in a particular environment: the time step must be small enough to avoid missing MACs between time samples.

Another parameter that has an apparent impact on the quality of the results is the simulation time, as demonstrated in Figure 11. If the simulation time is low (e.g., 100 flight-hours per point), when the number of UAS increases, the time flown by each of them can result significantly reduced, resulting in a highly noisy curve. However, increasing the simulation time to 2,000 flight hours apparently leads to more precise solutions. Nonetheless, the impact on the interpolated results is almost negligible (less than 1.5%).

Figure 13 and Figure 14 illustrate the effect of both the way the flight uncertainty is modelled and (in the case of sampled trajectories), the approach to inject new UAS into the simulations. It can be observed that the effect is not very apparent. Only in rather small urban scenarios the way trajectories are randomly sampled from the trajectory pool seems to have an influence on the NMCA frequency.

The main factor impacting the simulation results is the traffic pattern. Figure 17 compares the NMAC frequency obtained with the Endoh model and with linear trajectories with the NMAC frequency in the case UAS do not fly randomly but according to a given pattern, like the one displayed in Figure 18. It can be observed that the spatial correlation of trajectories has a dramatic effect on the NMAC frequency. The effect of reducing the spatial correlation is illustrated by the orange line in Figure 17, that corresponds to the same trajectories as in Figure 18 but randomly rotated and shifted, yielding the traffic pattern in Figure 19, much less correlated than the original in Figure 18.

5 U-AGREE Encounter model

5.1 Encounter model description

The evaluation of detect-and-avoid (DAA) and tactical conflict management services under realistic operational conditions cannot be addressed using purely analytical CRMs. Likewise, the numerical, Monte Carlo-based CRM described in section 4.2 is not suitable for assessing the performance of these systems and services, as the simulation time required to generate a sufficient number of encounters to ensure statistically robust results would be computationally prohibitive. Therefore, U-AGREE complements the extended gas model of Chapter 4 with a dedicated pairwise encounter model. The model is implemented as a fast-time, multi-thread Monte-Carlo simulator that:

1. generates pairwise conflicting trajectories,
2. drives them through the complete tactical conflict prediction and resolution (TCP&R) pipeline assumed for U-space,
3. injects realistic surveillance and communication degradation and
4. computes the performance metrics prescribed by ASTM F3442/F3442M-23 [7].

The approach is methodologically aligned with the Drone Encounter Generation (DEG) model of Pastor et al. [9] and with the Lincoln Laboratory encounter family [3], [8], but extended to account for the multi-USSP nature of U-space by modelling GNSS errors, communication latencies and packet losses on a per-aircraft basis. The simulator populates a 36 × 36 interaction matrix in which every cell stores the empirical estimates of the relevant probabilities for the corresponding pair of drone classes; postprocessing then aggregates the cell values into the global metrics for any desired traffic mix.

Encounter geometry and flight-plan generation

Each encounter is built on a circular convergence geometry. Two aircraft are initialised on radial paths heading towards a common centre point and separated by a crossing angle $\theta \sim \mathcal{U}\{5°, 6°, \dots, 355°\}$. The initial radial distance is set to $R = 2\, d_{\mathrm{enc},k}$, where $d_{\mathrm{enc},k}$ denotes the class-dependent encounter distance, equal to 1 NM and 2NM for the Small/Medium and Big speed classes, respectively. This geometry was selected because it (i) guarantees that virtually every generated pair leads to a nominal conflict, thereby focussing the simulation budget on actually-conflicting encounters and avoiding the inefficiency of free-flight scenarios, and (ii) provides a uniform angular coverage of the relative-bearing subspace, which is essential for the subsequent risk-ratio aggregation. The flight-plan associated with each radial leg is rendered into a 1 Hz sequence of position, velocity and attitude samples by a PX4 software-in-the-loop (SITL) instance, so that the resulting trajectories respect the realistic flight dynamics of the underlying airframe — turn-rate limits, climb/descent envelopes and altitude-hold behaviour — rather than being purely kinematic.

Traffic mix and aircraft types

Aircraft types are sampled at random from the 12-type EASA classification of UAS adopted by U-AGREE: Open A1/A2/A3 (multirotor and fixed-wing), PDRA (multirotor and fixed-wing), Specific SAIL (multirotor and fixed-wing), Certified and Manned VFR — using the per-scenario type-share weights defined in the U-AGREE operational scenarios. Each type comes with its own velocity range ($v_{\min}$,

Co-funded by the European Union

$v_{\max}$), turn rate, vertical rate and protection-volume size from which separation minima are derived; the trajectory generator therefore samples a cruise speed from the type-dependent interval ($v_{\min}$, $v_{\max}$) instead of using a single fixed value, so that the encounter set covers a continuum of flight regimes per type. The two largest types (Certified and Manned VFR) use the conventional ASTM F3442/F3442M-23 NMAC volume (500 ft horizontal, 100 m vertical) rather than the small-NMAC (sNMAC) one (50 ft horizontal, 30 ft vertical), in line with their dimensions and likely collision outcomes. The simulator stores the empirical estimates for every interaction (i, j) of the 12 × 12 = 144 cells of the resulting matrix is densely populated with statistically meaningful samples, and any operational traffic mix can be applied as a weighting function during postprocessing. Each cell stores the empirical estimates of $P_{ij}(NMAC|TCR)$, $P_{ij}(NMAC|\overline{TCR})$, $P_{ij}(LoS|TCR)$, $P_{ij}(LoS|\overline{TCR})$, the alert probability $P_{ij}(\mathrm{ALERT})$, computed from the number of events registered over the simulations assigned to that cell.

Vertical conflict generation and trajectory perturbations

To populate vertical conflict geometries, an independent random altitude offset uniformly distributed in $[30,200]\ \mathrm{m\,AGL}$ is applied to each trajectory, so that pairs of aircraft can encounter each other at any altitude inside the very-low-level (VLL) airspace. In addition, two trajectory-perturbation controls are exposed at the configuration level, defined independently for fixed-wing and multirotor platforms in order to reflect their different dynamic behaviour:

- Random Turn probability $RT_{\mathrm{FW}}, RT_{\mathrm{MR}} \in [0,1]$: at every 1 Hz step of an aircraft's trajectory, a Bernoulli trial of probability RT decides whether a single random heading deviation is triggered. When triggered, a turn angle is drawn from $\Delta\chi \sim \mathcal{U}[0°, 90°]$ and a sign is drawn from $\{-1, +1\}$ with equal probability; the manoeuvre is then executed at the class-dependent maximum turn rate ω_{max}. This control captures the sporadic, low-probability deviations from the nominal cruise heading that occur when remote pilots correct course or react to unmodelled disturbances.
- Climb Probability $CP_{\mathrm{FW}}, CP_{\mathrm{MR}} \in [0,1]$: analogous Bernoulli control governing whether a vertical excursion is triggered. When fired, a climb angle is drawn from $\gamma \sim \mathcal{U}[\gamma_{min}, \gamma_{max}]$ class-specific limits ($[1°, 8°]$ for FW and $[10°, 90°]$ for MR), and the aircraft executes a vertical leg at its rate-of-climb $v_z = \pm\,\mathrm{RoC/RoD}$ until either the upper/lower altitude limit is reached or the leg duration expires. CP therefore exercises the vertical separation logic of the TCP&R pipeline.

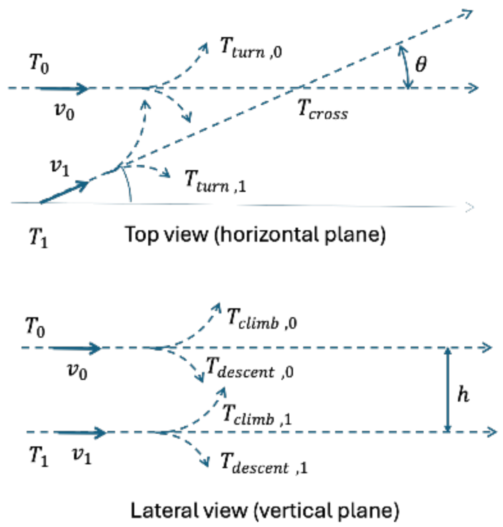


Figure 20. Encounter trajectories used for the validation.

Surveillance and tracking pipeline

The 1 Hz reported positions of each UAS are processed by a six-state Interacting Multiple Model (IMM) filter, which runs three parallel Kalman filters — Constant Velocity (CV), Constant Acceleration (CA) and Coordinated Turn (CT) implemented as a Singer-type model with short correlation time to account for the high manoeuvrability of UAS — and combines them through adaptive mode probabilities. At each update the filter delivers the position and velocity estimates $\boldsymbol{p}_i^k$ and $\boldsymbol{v}_i^k$ together with their covariance, which feed the tactical conflict prediction algorithm. The IMM scheme has been retained because it smoothly hands over among the three motion models when an aircraft transitions from cruise to manoeuvre, which is essential to keep the false-alarm probability bounded under perturbed trajectories.

Quadratic conflict detection

Under the constant-velocity assumption, the position of aircraft i at future time $t \geq t_k$ is given by:

$$\boldsymbol{p}_i(t) = \boldsymbol{p}_i^k + \boldsymbol{v}_i^k\,(t - t_k) \qquad \textbf{Eq. 38}$$

Defining the relative horizontal position and velocity at t_k as $\boldsymbol{p}_{\mathrm{rlxy}}^k = \boldsymbol{p}_{j,xy}^k - \boldsymbol{p}_{i,xy}^k$ and $\boldsymbol{v}_{\mathrm{rlxy}}^k = \boldsymbol{v}_{j,xy}^k - \boldsymbol{v}_{i,xy}^k$, the squared horizontal distance between aircraft i and j evolves quadratically with the elapsed time $\tau = t - t_k$:

$$d^2(\tau) = \left\|\boldsymbol{v}_{\mathrm{rlxy}}^k\right\|^2 \tau^2 + 2\left(\boldsymbol{p}_{\mathrm{rlxy}}^k \cdot \boldsymbol{v}_{\mathrm{rlxy}}^k\right)\tau + \left\|\boldsymbol{p}_{\mathrm{rlxy}}^k\right\|^2 \qquad \textbf{Eq. 39}$$

Each aircraft is enveloped by a cylindrical protection volume with horizontal radius $d_{\mathrm{SL},i}$ and vertical half-height $h_{\mathrm{SL},i}$. The pairwise separation minima are therefore $d_{\mathrm{SL},ij} = d_{\mathrm{SL},i} + d_{\mathrm{SL},j}$ horizontally and $h_{\mathrm{SL},ij} = h_{\mathrm{SL},i} + h_{\mathrm{SL},j}$ vertically. Setting $d^2(\tau) = d_{\mathrm{SL},ij}^2$ produces the standard quadratic equation $A\tau^2 + B\tau + C = 0$ with coefficients:

$$A = \left\|\boldsymbol{v}^k_{\mathrm{rlxy}}\right\|^2, \qquad B = 2\left(\boldsymbol{p}^k_{\mathrm{rlxy}} \cdot \boldsymbol{v}^k_{\mathrm{rlxy}}\right), \qquad C = \left\|\boldsymbol{p}^k_{\mathrm{rlxy}}\right\|^2 - d^2_{\mathrm{SL},ij}. \qquad \textbf{Eq. 40}$$

If the discriminant $\Delta = B^2 - 4AC \leq 0$ or $A = 0$, no horizontal conflict is predicted; otherwise the entry and exit times are:

$$\tau_{\mathrm{entry}} = \frac{-B - \sqrt{\Delta}}{2A}, \qquad \tau_{\mathrm{exit}} = \frac{-B + \sqrt{\Delta}}{2A}. \qquad \textbf{Eq. 41}$$

The vertical distance evolves linearly as:

$$h(\tau) = \left(p^k_{i,z} - p^k_{j,z}\right) + \left(v^k_{i,z} - v^k_{j,z}\right)\tau, \qquad \textbf{Eq. 42}$$

and a tactical conflict is declared when $0 \leq \tau_{\mathrm{entry}} \leq t_{\mathrm{LA}}$ and $\left|h\left(\tau_{\mathrm{entry}}\right)\right| < h_{\mathrm{SL},ij}$, i.e. when both separation minima are projected to be breached simultaneously within the look-ahead horizon t_{LA}. To remove transient detections, the avoidance manoeuvre is triggered only after three consecutive 1 Hz samples in conflict with strictly decreasing τ_{entry}, which filters out spurious alerts caused by tracker noise.

GNSS error model

A significant extension introduced in U-AGREE with respect to prior work is the individual modelling of surveillance degradations for each aircraft, which is necessary to capture the heterogeneous performance of the multi-USSP U-space ecosystem. The GNSS error applied to the reported position of aircraft i at every simulation step is modelled as a zero-mean Gaussian process with independent horizontal and vertical components,

$$\varepsilon_{h,i} \sim \mathcal{N}\left(0, \sigma^2_{h,i} I_2\right), \qquad \varepsilon_{v,i} \sim \mathcal{N}\left(0, \sigma^2_{v,i}\right), \qquad \textbf{Eq. 43}$$

where the configuration parameters are exposed as the 95 % two-sided percentile of the error (i.e. the 2σ values $\sigma_{h,2} = 2\sigma_{h,i}$ and $\sigma_{v,2} = 2\sigma_{v,i}$) and the relationship between the configured σ and the standard deviation actually used by the simulator is $\sigma = \sigma_2/2$. The baseline NSE 95 % configuration assumes $\sigma_{h,2} = 1.87\ \mathrm{m}$ and $\sigma_{v,2} = 3.16\ \mathrm{m}$, consistent with open-sky UAS-grade GNSS receivers. The perturbed Cartesian position fed to the IMM filter is:

$$\tilde{x}_i = x_i + \varepsilon_{x,i}, \qquad \tilde{y}_i = y_i + \varepsilon_{y,i}, \qquad \tilde{z}_i = z_i + \varepsilon_{v,i}, \qquad \textbf{Eq. 44}$$

while the ground-truth positions are retained for NMAC and loss-of-separation scoring (see below). Crucially, the GNSS error parameters are defined independently for each aircraft, $\left(\sigma_{h,1}, \sigma_{v,1}\right) and \left(\sigma_{h,2}, \sigma_{v,2}\right)$, which is what allows the simulator to represent multi-USSP scenarios in which UAS subscribed to different USSPs are tracked through equipment of different quality.

Communication latency model

The finite delay between position sampling at the aircraft and reception at the TCP&R service is modelled as a per-aircraft latency $\tau_{\mathrm{lat},i}$ drawn once per encounter from a half-normal distribution,

$$\tau_{\mathrm{lat},i} = |Z_i|, \qquad Z_i \sim \mathcal{N}\left(0, (L_i/2)^2\right), \qquad \textbf{Eq. 45}$$

where L_i is the 2σ communication latency declared for aircraft i (with the baseline values $L_1 = 1$ s and $L_2 = 2$ s). The half-normal form ensures that the resulting delay is non-negative while preserving an exponential-like tail. The effect of $\tau_{\mathrm{lat},i}$ is applied by delaying the position report fed to the IMM filter by $\lfloor \tau_{\mathrm{lat,i}}/\Delta \mathrm{t} \rfloor$ steps (with $\Delta t = 1$ s), so that the conflict-detection algorithm operates on a stale state vector while the ground truth continues to evolve normally. Treating each aircraft independently is essential because in multi-USSP environments one aircraft may be tracked by its own USSP through a low-latency local link while the other is made available via a higher-latency USSP-to-USSP exchange — a condition that cannot be reproduced with a single global latency value.

Packet-loss model

A Bernoulli packet-loss process is applied independently to each aircraft's telemetry stream. At every 1 Hz update, the telemetry packet from aircraft i is dropped with probability $p_{\mathrm{loss},i} \in [0,1]$ (baseline $p_{\mathrm{loss},1} = 0.02$, $p_{\mathrm{loss},2} = 0.05$) whenever an independent uniform draw $u_i \sim \mathcal{U}[0,1]$ satisfies $u_i < p_{\mathrm{loss},i}$. When a packet is lost, the IMM filter does not perform a measurement update; only the time-propagation (predict-only) step is applied, with the consequence that the covariance of the state estimate grows as $\Sigma_{\mathrm{k+1|k}} = \mathrm{F} \cdot \Sigma_{\mathrm{k|k}} \cdot \mathrm{F}^T + \mathrm{Q} \cdot \Delta$ until the next valid measurement. As a result, sustained or bursty packet losses induce a progressive degradation of the tracking quality that is propagated downstream to the quadratic conflict detection step. Providing two independent p_{loss} values — one per aircraft — again preserves the ability to model multi-USSP heterogeneity in communication quality.

Pilot reaction time model

In addition to the surveillance and communication degradations, the simulator includes a configurable model of the time that elapses between the moment a conflict-management instruction is available at the cockpit and the moment the avoidance manoeuvre actually starts to materialise on the trajectory. This *pilot reaction time* is sampled once per encounter, independently for each aircraft, and is enforced as a deterministic delay applied on top of the IMM tracker output and the communication-latency model: the avoidance manoeuvre triggered by the conflict-resolution logic does not begin at the alert instant, but at the alert instant plus a per-aircraft realisation of the reaction-time random variable. The model is therefore active only on the *mitigated* execution of each encounter, since the unmitigated baseline assumes that no manoeuvre is attempted.

The reaction time of aircraft $i$is modelled as a (possibly shifted) gamma random variable,

$$T_{r,i} = X_i + t_{min,i}, X_i \sim \mathrm{Gamma}(k_i, \theta_i), \qquad \textbf{Eq. 46}$$

whose probability density function is:

$$f(t; k, \theta, t_{min}) = \frac{(t - t_{min})^{k-1}\, e^{-(t-t_{min})/\theta}}{\theta^k\, \Gamma(k)}, t \geq t_{min}, \qquad \textbf{Eq. 47}$$

where $k > 0$is the shape parameter, $\theta > 0$is the scale parameter, $t_{min} \geq 0$is a deterministic shift that captures the minimum physiological processing time of a remote pilot, and $\Gamma(\cdot)$is the gamma function. The summary statistics of the resulting distribution are $\mu = k\theta + t_{min}$, mode $= (k-1)\theta + t_{min}$, $\sigma = \theta\sqrt{k}$and skewness $2/\sqrt{k}$. The shifted gamma family was chosen because it allows the reaction time to take a strictly positive lower bound t_{min} while preserving an asymmetric, exponential-like upper tail, which is consistent with empirical observations of human reaction times under operational stress.

The simulator exposes the triple $(k_i, \theta_i, t_{min,i})$ as a per-aircraft configuration parameter, in the same spirit as the per-aircraft GNSS, latency and packet-loss models. This allows the user to represent multi-USSP scenarios in which the two aircraft involved in a given encounter are operated by remote pilots with different training levels or different cockpit-interface ergonomics, simply by feeding two different parameter triples into the simulator. Two reference parameterisations are supported out of the box and can be selected through the simulator configuration:

- A short-reaction parameterisation, intended to model pilots that receive a fully specified avoidance manoeuvre from a Tactical Conflict Resolution Service (TCRS) and only need to acknowledge and mechanically execute it. The corresponding triple is $(k, \theta, t_{min}) = (1.584,\ 1.541,\ 6.0)$s, which yields a mean reaction time of approximately 8.4 s and a 99-th percentile of approximately 17 s once added on top of the surveillance/tracking latency $T_1 \sim \text{Gamma}(k_1 = 2.513, \theta_1 = 0.661)$ that the simulator applies before the alert reaches the cockpit. The total end-to-end response time $T_{tot} = T_1 + T_{r,i}$ is therefore the convolution

$$f_{T_{tot}}(t) = \int_0^{t-t_{min}} f_{T_1}(\tau)\, f_{T_{r,i}}(t-\tau)\, d\tau, \qquad \textbf{Eq. 48}$$

  which is evaluated numerically by the postprocessing pipeline whenever the 99-th percentile of the total response time is required as a reference for the minimum effective look-ahead time.

- A long-reaction parameterisation, intended to model pilots that receive only a generic conflict alert (i.e. no specific avoidance manoeuvre is pre-computed) and have to design and initiate the avoidance manoeuvre on their own. To capture the dispersion in pilot performance expected in real operations, this parameterisation is instantiated at three different training levels, each of them with a different scale parameter and a common $(k, t_{min}) = (2.0,\ 5.0)$s: high (expert) with $\theta = 7.5$s, medium (intermediate) with $\theta = 17.5$s and low (novice) with $\theta = 27.5$s, yielding mean reaction times of 20 s, 40 s and 60 s and 99-th percentiles of approximately 55, 121 and 187 s, respectively.

Sampling $T_{r,i}$ from a different parameterisation between the two aircraft of a pair, or between consecutive encounters, allows the simulator to expose the sensitivity of the residual risk ratio to the assumed pilot performance without re-executing the rest of the Monte-Carlo machinery. As is the case for the other surveillance and communication degradations, the per-aircraft definition of $T_{r,i}$ is what makes the U-AGREE encounter model representative of the heterogeneous multi-USSP environments envisaged for U-space.

Event scoring

At the end of each simulated encounter, four classes of events are extracted from the ground-truth (un-perturbed) position streams, in parallel, both for the unmitigated execution (no TCP&R) and for the mitigated one (TCP&R active): (i) the horizontal and vertical minimum distance between the two aircraft along the simulation; (ii) the NMAC flag, set to 1 when, simultaneously at any time step, the horizontal distance is below the applicable NMAC threshold and the vertical distance is below the applicable vertical NMAC threshold; (iii) the loss-of-separation (LoS) flag, set to 1 when the cylindrical protection volume defined by $(d_{\mathrm{SL},i}, h_{\mathrm{SL},i})$ is breached at any time step; and (iv) the ALERT flag, set to 1 if the TCP&R service issues at least one conflict detection during the encounter. The NMAC volumes follow the criteria of Weinert et al. [8]: the small-NMAC (sNMAC) volume $H = 50$ ft, $V = 15$ ft is used for encounters in which both aircraft are Small or Medium UAS, while the conventional NMAC volume $H = 500\,\mathrm{m}$, $V = 100\,\mathrm{m}$ prescribed by ASTM F3442/F3442M-23 [7] is used whenever a Big/manned aircraft is involved.

Performance metrics

The performance of the TCP&R service is characterised through the four metrics defined in ASTM F3442/F3442M-23 [7]. The primary safety metric is the Risk Ratio:

$$RR = \frac{P(NMAC|TCR)}{P(NMAC|\overline{TCR})}, \quad \text{Eq. 49}$$

which estimates the factor by which the TCP&R service reduces the likelihood of an NMAC observed in the unmitigated execution of the same encounter. The Loss-of-Separation Ratio is defined analogously,

$$LR = \frac{P(LoS|TCR)}{P(LoS|\overline{TCR})}, \quad \text{Eq. 50}$$

and the false-alarm probability is:

$$LP_{fa} = 1 - \frac{P(NMAC|\overline{TCR})}{P(ALERT)}, \quad \text{Eq. 51}$$

with the alert rate being computed as the fraction of simulated encounters that trigger at least one conflict detection. These four metrics jointly capture the safety/operational-burden trade-off that characterises any DAA-like system: RR and LR measure how effective the service is at preventing NMACs and losses of separation, while P_{fa} and the alert rate measure the price paid in terms of unnecessary manoeuvres and pilot workload.

Postprocessing and traffic-mix weighting

Because each of the 144 cells of the 12 × 12 interaction matrix is populated with a sufficiently large number of samples, the global metrics for any desired fleet composition can be obtained in postprocessing without re-executing the Monte-Carlo simulation. Let p_i be the traffic share of UAS type i (with $\sum_{i=1}^{36} p_i = 1$); the pair-interaction weight is:

$$w_{ij} = \frac{p_i\, p_j}{\sum_{k=1}^{36} \sum_{\mathrm{l}=1}^{36} p_k\ p_{\mathrm{l}}} = p_i\, p_j. \quad \text{Eq. 52}$$

The weighted probabilities are then:

$$P_w(NMAC|TCR) = \sum_{i=1}^{12} \sum_{j=1}^{12} w_{ij} P_{ij}(NMAC|TCR), \quad \text{Eq. 53-a}$$

$$P_w(NMAC|\overline{TCR}) = \sum_{i=1}^{12}\sum_{j=1}^{12} w_{ij} P_{ij}(NMAC|\overline{TCR}),$$ **Eq. 53-b**

with analogous definitions for $P_w(LoS|\ \cdot)$ and $P_w(ALERT)$. The aggregated risk ratio is finally obtained as a ratio of weighted means rather than as a weighted mean of per-cell ratios,

$$RR = \frac{P_w(NMAC|TCR)}{P_w(NMAC|\overline{TCR})}, LR = \frac{P_w(LoS|TCR)}{P_w(LoS|\overline{TCR})}, P_{fa} = 1 - \frac{P_w(NMAC|\overline{TCR})}{P_w(ALERT)}.$$ **Eq. 54**

This formulation has the advantage of remaining well-defined for cells with zero unmitigated NMAC events (which would make a per-cell ratio undefined) and is the form actually applied in the U-AGREE postprocessing pipeline.

6 References

[1] Hamissi, A., Dhraief, A., & Sliman, L. (2025). A Comprehensive Survey on Conflict Detection and Resolution in Unmanned Aircraft System Traffic Management. IEEE Transactions on Intelligent Transportation Systems, 26(2), 1395–1406.

[2] Hoekstra, J. M., Ellerbroek, J., & Ribeiro, M. J. (2020). Review of Conflict Resolution Methods for Manned and Unmanned Aviation. Aerospace, 7(6), 79. https://doi.org/10.3390/aerospace7060079.

[3] Kochenderfer, M. J., Weinert, A. J., Harkleroad, E. P., Griffith, J. D., & Edwards, M. W. (2013). Uncorrelated Encounter Model of the National Airspace System Version 2.0. Lincoln Laboratory, MIT. Project Report ATC-404.

[4] SESAR Joint Undertaking. (2022). BUBBLES Project – Concept Formulation (Deliverable D2.1, Edition 04.00.00). Grant Agreement No. 893206.

[5] Endoh, S. (1982). Aircraft Collision Models. Flight Transportation Laboratory Report R82-2, Massachusetts Institute of Technology.

[6] Krauth, T., Figuet, B., Olive, X., & Morio, J. (2023). Collision Risk Assessment in Terminal Manoeuvring Areas based on Trajectory Generation Methods.

[7] ASTM, F3442/F3442M-20 Standard Specification for Detect and Avoid System Performance Requirement, May 1st, 2020.

[8] Weinert, A., Alvarez, L., Owen, M., & Zintak, B. (2022). Near Midair Collision Analog for Drones Based on Unmitigated Collision Risk. Journal of Air Transportation, 30(2), 37–48. https://doi.org/10.2514/1.D0260.

[9] Pastor, E., Del Hierro, S., & Barrado, C. (2024). A Drone Encounter Model for Detect and Avoid Evaluation in U-space. SESAR Innovation Days 2024, Rome, Italy.

[10] Alvarez, L. E., Jessen, I., Owen, M. P., Silbermann, J., & Wood, P. (2019). ACAS sXu: Robust Decentralized Detect and Avoid for Small Unmanned Aircraft Systems. 2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC).

[11] Sanchis-Marín, Álex, Esparza-Becerra, Silvia Rocío, Balbastre Tejedor, Juan Vicente. (2025) Realistic Trajectory Generation in Simulated Environments for U-space Systems Assessment. 25th Integrated Communications, Navigation, and Surveillance Conference (ICNS 2025).

[12] H. A. P. Blom, B. Bakker, M. Everdij, and M. van der Park, "Collision risk modelling of air traffic," Proc. European Control Conference (ECC), 2003.

[13] M. J. Kochenderfer, M. W. M. Edwards, L. P. Espindle, J. K. Kuchar, and J. D. Griffith, "Airspace encounter models for estimating collision risk," Journal of Guidance, Control, and Dynamics, vol. 33, no. 2, pp. 487–499, 2010.

[14] V. Kallinen, S. Barry, and A. McFadyen, "Collision risk quantification for pairs of recorded aircraft trajectories," Journal of Navigation, vol. 76, 2023.

[15] C.-H. J. Wang, C. Deng, and K. H. Low, "Parametric study of structured UTM separation recommendations with physics-based Monte Carlo distribution for collision risk model," Drones, vol. 7, no. 6, 2023.