ASSURE A71 – Conduct Safety Risk Management Analysis on Unmanned Aircraft Detect and Avoid Systems

Task 3 Issue Report

January 21, 2026

v 1.0

NOTICE

This document is disseminated under the sponsorship of the U.S. Department of Transportation in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof. The U.S. Government does not endorse products or manufacturers. Trade or manufacturers' names appear herein solely because they are considered essential to the objective of this report. The findings and conclusions in this report are those of the author(s) and do not necessarily represent the views of the funding agency. This document does not constitute FAA policy. Consult the FAA sponsoring organization listed on the Technical Documentation page as to its use.

LEGAL DISCLAIMER

The information provided herein may include content supplied by third parties. Although the data and information contained herein has been produced or processed from sources believed to be reliable, the Federal Aviation Administration makes no warranty, expressed or implied, regarding the accuracy, adequacy, completeness, legality, reliability, or usefulness of any information, conclusions or recommendations provided herein. Distribution of the information contained herein does not constitute an endorsement or warranty of the data or information provided herein by the Federal Aviation Administration or the U.S. Department of Transportation. Neither the Federal Aviation Administration nor the U.S. Department of Transportation shall be held liable for any improper or incorrect use of the information contained herein and assumes no responsibility for anyone's use of the information. The Federal Aviation Administration and U.S. Department of Transportation shall not be liable for any claim for any loss, harm, or other damages arising from access to or use of data or information, including without limitation any direct, indirect, incidental, exemplary, special, or consequential damages, even if advised of the possibility of such damages. The Federal Aviation Administration shall not be liable to anyone for any decision made or action taken, or not taken, in reliance on the information contained herein.

TECHNICAL REPORT DOCUMENTATION PAGE

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
A11L.UAS.120_A71		

4. Title and Subtitle	5. Report Date
ASSURE A71 – Conduct Safety Risk Management Analysis on Unmanned System Detect and Avoid Systems: Issue Paper	January 21, 2026
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Tom Haritos, PhD., https://orcid.org/0000-0001-6546-383X Tim Bruner, https://orcid.org/0000-0002-7591-8823 Katie Silas, https://orcid.org/0000-0003-0647-4592 Kurt Carraway, https://orcid.org/0000-0002-1362-8177 Paul Snyder, https://orcid.org/0000-0003-2417-6388 Mark Askelson, https://orcid.org/0000-0002-8521-7158 James Cooley, Ph.D., https://orcid.org/0009-0004-2842-5892 Hever Moncayo, Ph.D., https://orcid.org/0000-0001-5088-9715 Richard Prazenica, Ph.D., https://orcid.org/0000-0001-6093-8891 Steven Weber Ph.D., https://orcid.org/0000-0002-9235-6922 Lifeng Zhou, Ph.D., https://orcid.org/0000-0001-7927-8504 Amirhosein Chahe, https://orcid.org/0009-0002-6883-2041 Sadia Afrin Ananna, https://orcid.org/0009-0003-6376-3275 Jorge Estupiñán, https://orcid.org/0009-0005-4922-3461 Rocio Jado-Puente, https://orcid.org/0000-0002-7567-1613	

9. Performing Organization Name and Address	10. Work Unit No.
Kansas State University University of North Dakota Embry Riddle Aeronautical University Drexel University	**11. Contract or Grant No.** A11L.UAS.120_A71

12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
Federal Aviation Administration UAS COE PM: Hector Rea, ANG-C2	Issue Report
	14. Sponsoring Agency Code 5401

15. Supplementary Notes

Conducted in cooperation with the U.S. Department of Transportation, Federal Highway Administration.

16. Abstract

This issue paper explores DAA system functions and operations against the backdrop of the Safety Risk Management (SRM) process to identify issues and gaps that pose challenges to assessing risks associated with DAA systems. Framing this issue paper in terms of the SRM process – describing the system, identifying hazards, assessing risk, analyzing risk, and controlling risk – provides a rational way to look for issues and gaps that challenge effective SRM for DAA systems in each process step. This approach identifies issues and gaps that may be considered and addressed during the development of a DAA risk assessment framework in future research tasks.

17. Key Words	18. Distribution Statement
Safety Management Systems, Safety Risk Management, Detect and Avoid, Beyond Visual-Line-of-Sight,	No restrictions.

19. Security Classification (of this report)	20. Security Classification (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	113	N/A

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

TABLE OF ACRONYMS

ADS-B	Automatic Dependent Surveillance-Broadcast
AGL	Above Ground Level
ASTM	American Society for Testing and Materials
CPA	Closest Point of Approach
DAA	Detect And Avoid
ECEF	Earth-Centered, Earth-Fixed
EKF	Extended Kalman Filter
FGA	Fast Geometric Avoidance
GNSS	Global Navigation Satellite System
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
LiDAR	Light Detection And Ranging
LOS	Line of Sight
MIT	Massachusetts Institute of Technology
PRA	Probabilistic Risk Assessment
ROS	Robot Operating System
RTP	Research Task Plan
SRM	Safety Risk Management
TCCA	Transport Canada Civil Aviation
UAV	Unmanned Aerial Vehicle
URDF	Unified Robot Description Format
YOLO	You Only Look Once

EXECUTIVE SUMMARY

This report satisfies the requirements of Task 3 of the ASSURE project "A71: Conduct Safety Risk Management Analysis on Unmanned Aircraft Detect and Avoid Systems." The report is aligned with the two subtasks defined in the Research Task Plan (RTP):

- *Task 3-1:* Simulation and system requirements to support sensitivity and DAA Performance (Section 3)
- *Task 3-2:* Safety risk management analysis document (Section **Error! Reference source not found.**).

Task 3-1 developed a modular simulation environment to enable systematic evaluation of Detect And Avoid (DAA) performance under diverse encounter geometries, system configurations, and operational conditions. Because publicly available datasets and performance data for vision-based DAA systems remain limited, the simulation architecture integrates multiple high-fidelity tools to support sensitivity studies and Probabilistic Risk Assessment (PRA). In particular, the environment enables assessment of DAA performance using the parameterized, machine-learning-based probabilistic model developed in Task 2.1 (with model development details documented in the Task 2 Issue Report).

Task 3-2 applied the PRA framework to estimate the instantaneous detection probability as a function of the instantaneous separation distance between two aircraft. Using realistic encounter trajectories from Massachusetts Institute of Technology (MIT) Lincoln Laboratory, Monte Carlo simulations were executed to quantify detection performance across a range of system characteristics (e.g., Casia X–inspired camera resolution and camera-out failures) and operating conditions (e.g., fog levels and visual clutter). The resulting distributions support estimation of instantaneous detection probability under specific configurations and environments. In addition, the report includes preliminary results characterizing the performance of an Automatic Dependent Surveillance-Broadcast (ADS-B)–based DAA system and demonstrating the application of the same probabilistic risk assessment approach to ADS-B-derived state estimates under different operating conditions (e.g., Global Navigation Satellite System (GNSS) signal degradation).

Summary of findings:

- The combination of high-fidelity simulation, machine-learning-based probabilistic modeling, and Monte Carlo analysis provides strong support for risk assessment of DAA systems (e.g., Casia X–inspired vision DAA and ADS-B DAA), particularly when limited empirical data constrain performance characterization across operating conditions.
- Sensor degradations and operating conditions complexity can be dominant drivers of detection risk: camera outages and high visual clutter (and, separately, GNSS dropouts that degrade ADS-B state estimates) can substantially reduce detection rates and effective detection range, highlighting the need to explicitly model both system reliability and operational conditions when assessing DAA performance and safety margins.

1 INTRODUCTION AND BACKGROUND

1.1 Background and Context

This document is the third report for the ASSURE project entitled "A71: Conduct Safety Risk Management Analysis on Small Unmanned Aircraft Detect and Avoid Systems." It is a sequel to the first and second reports, pursuant to the first and second tasks.

From the RTP, the task exit criteria are the two subtasks named Subtask 3-1 and Subtask 3-2:

- *Subtask 3-1: Simulation and system requirements to support sensitivity and DAA performance.*
- Subtask *3-2: Safety Risk Management Analysis Document.*

1.2 Summary of Subtask 3-1: Simulation and system requirements to support sensitivity and DAA performance.

The first subtask has the following description in the RTP:

> *"In this task, ERAU will lead the development of a high-fidelity simulation environment to support the design, testing, and evaluation of assessment methods developed in Task 2. A modular structure will be adopted for maximum portability, flexibility, and extension capability through the addition of new models and modules to comply with standards. A representative model (e.g., ACAS classes for unmanned aircraft, ADS-B, or TCAS) along with the operating environment will be implemented and will be integrated into the simulation environment to allow the application of a sensitivity analysis in support of Task 2 of this A71 project."*

To address this subtask 3-1, the presented report is organized into two main parts. First, it summarizes guidance from Transport Canada Civil Aviation (TCCA) to the American Society for Testing and Materials (ASTM) WK62669 working group on how modeling-and-simulation activities should be tied to empirical flight-test evidence when evaluating DAA performance. This section describes the stated purposes of flight testing (sampling encounter space and stressing the DAA performance envelope), discusses that TCCA does not prescribe a fixed "enough data" quantity, and includes an example radar-supported DAA case study and discussion of how position errors should be carried through the broader DAA chain (not just detection probability).

Second, the report describes the simulation environment developed using a modular Robot Operating System (ROS)2–Gazebo architecture. It lays out the overall structure and the main components: fixed-wing Unmanned Aerial Vehicle (UAV) dynamic models for ownship and intruders, sensor plugins (Inertial Measurement Unit (IMU), cameras, GNSS), a modeled ADS-B system, visual intruder detection, predicted trajectories, a trajectory follower, and a Fast Geometric Avoidance (FGA) controller, all operating within configurable 3D cluttered environments (including fog). It also details simulation configuration options for different ownship sensor packages (single camera, a CasiaX-inspired multi-camera layout, and random camera availability).

Within that setup, the report explains how navigation and surveillance signals are generated and used. It describes GNSS-plugin modifications that enable simulated navigation degradation (including a dropout flag that reduces visible satellites and worsens the least-squares position solution) and additional modeling, such as Line of Sight (LOS) obstruction/multipath handling. For ADS-B, it describes intruders broadcasting estimated state via an Extended Kalman Filter that fuses IMU and GNSS data (in an Earth-centered reference frame) for position/velocity estimation at a set update rate. The vision-based DAA module outlines a CasiaX-like 360° concept using five cameras around the airframe, including the camera placement, overlap, and the image-stitching preprocessing step. Because the proprietary CasiaX classifier is not available, the report implements an open-source vision-based detector, explains why different vision-based detection system variants were compared, and presents a table of ideal-condition detection distances and processing times; based on that comparison, a specific configuration is selected as the detection model used in the simulations.

1.3 Summary of Subtask 3-2: Safety Risk Management Analysis Document.

The second subtask has the following description in the RTP:

> *"For this task, ERAU will lead multiple and extensive simulation tests, a comparison analysis will be performed between traditional risk assessment methods and the assessment approach proposed in Task 2.0. Results from Task 3 will be compiled within a document that will include hazard identification along with risk assessment and applicable to ASTM and/or RTCA standards."*

To address this subtask 3-2, the third part of this report presents Monte Carlo simulation studies and example avoidance demonstrations. It describes replaying 1,000 MIT encounter trajectories near the Closest Point of Approach (CPA), running each case for at least ~75 seconds, and varying both environment type and visibility conditions. Clutter conditions are emphasized, with exponential fog applied at four levels, and the report shows how detections and confidence distributions change as fog density and background clutter change. It also summarizes detection range distributions (including reported mean ranges in the reference environment decreasing with increasing fog), detection-success rates as a function of range/fog, and the presence of false positives in the cluttered environment (reported via a table of false-positive percentages above a 0.7 confidence threshold).

A second set of Monte Carlo configuration include cases with partial camera functionality of the DAA CasiaX system under a cluttered environment, while randomly limiting the ownship. The report shows how this DAA degradation in performance changes detection density/coverage.

Finally, the report includes preliminary ADS-B DAA simulations demonstrating the FGA algorithm in several single-intruder encounter geometries and then repeats them with GNSS dropouts to illustrate how degraded ADS-B position estimates affect intruder localization and avoidance-trajectory generation; it also shows representative two-intruder cases, including situations where avoidance zones overlap and are treated as a combined region.

2 INTEGRATION OF FLIGHT TEST EVIDENCE WITH MODELING AND SIMULATION: INSIGHTS FROM TCCA ASTM WK62669 (MCKAY, 2023).

TCCA provided a detailed perspective to the ASTM WK62669 working group on how Modeling and Simulation activities should interface with empirical flight test data when evaluating DAA system performance (McKay, 2023). Because Task 3 of A71 aims to build a credible, simulation-driven framework that can support safety-based mitigations, these insights directly inform how flight-test evidence may be used to validate and anchor Monte Carlo–based simulation results (RPAS TF ENG – ASTM WK62669 TCCA Presentation).

2.1 Purpose and Role of Flight Test Data

TCCA emphasizes two primary objectives for DAA-related flight testing:

1. **Coverage of the encounter space.**

   Flight tests should sample critical portions of the geometric and kinematic encounter space to ensure the modeled encounters are representative of operationally relevant conditions.
2. **Coverage of the DAA system performance envelope.**

   Tests must challenge the system across its operational limits, including challenging geometries, low-detectability conditions, and maneuvers that stress sensor performance, tracking, and alerting logic.

TCCA stresses that the greatest validation value comes from testing challenging cases at the edges of the performance envelope, rather than nominal scenarios located in the "heart of the envelope" (ASTM WK62669). Data collected in nominal regions may support test setup and procedural validation, but contribute comparatively less to model validation credibility.

2.2 How Much Data Is "Enough"?

TCCA explicitly avoids prescribing a fixed quantity of flight test data. Instead, sufficiency depends on:

- The complexity of the DAA system
- The intended operational environment
- Expected intruder behaviors
- Environmental effects such as terrain masking and clutter

As illustrated in the ASTM WK62669 presentation, a flight test campaign that adequately samples the challenging portions of the envelope is generally sufficient to validate modeling assumptions. Agreement between modeled and observed performance in these regions substantially increases confidence in simulation-based extrapolation to untested encounter conditions.

2.3 Example Case: Ground-Based Radar DAA Support

The TCCA presentation includes a representative case study involving a ground-based radar detection function supporting procedural, human-in-the-loop DAA operations. Key characteristics relevant to A71 Task 3 include:

• A 2D radar sensor without altitude resolution, requiring a conservative assumption that horizontally approaching tracks are co-altitude.

• Operations conducted below 400 ft Above Ground Level (AGL), consistent with low-altitude beyond visual line of sight concepts of operation.

• Physics-based modeling of detection probability as a function of range and altitude.

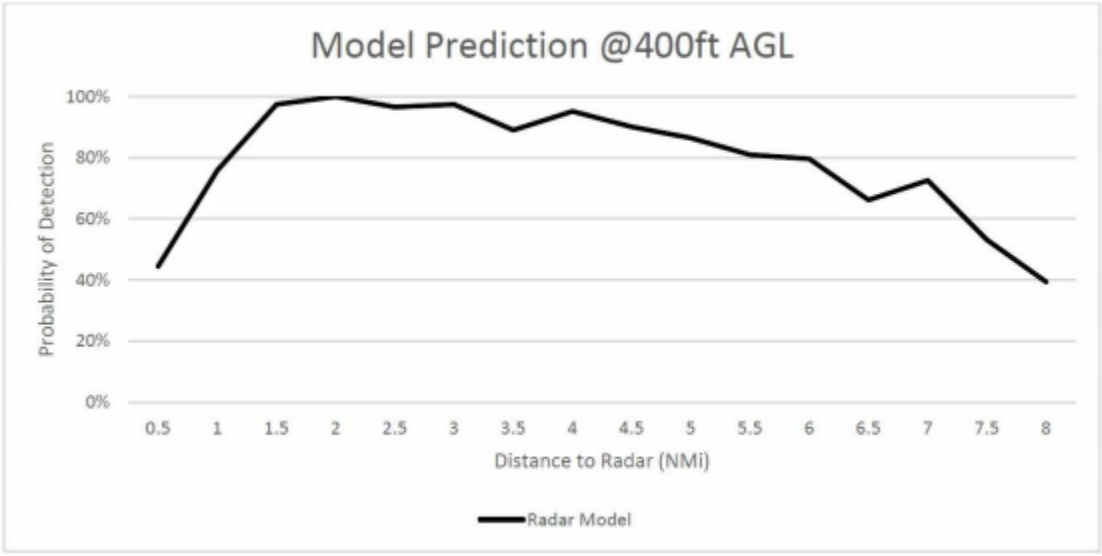


Figure 1. Modelled Probability of Detection at 400ft AGL.

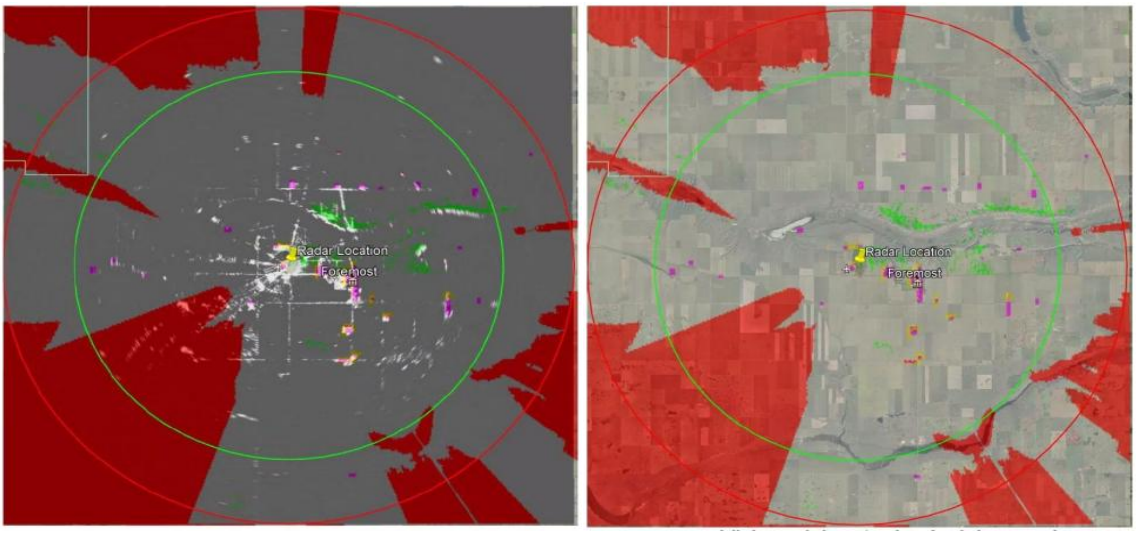


Figure 2. Modelled Ground Clutter/Masking for Flight 1 @400ft AGL.

Figure 3. Ground Clutter from Radar overlaid with Model @400ft AGL.

Figure 1 illustrates modeled probability of detection at 400 ft AGL as a function of radar range, while Figures 2 and 3 demonstrate how terrain and ground clutter masking were incorporated into the detection model (RPAS TF ENG–ASTM WK62669). These modeling results explicitly account for environmental effects that degrade detection performance at specific azimuths and ranges.

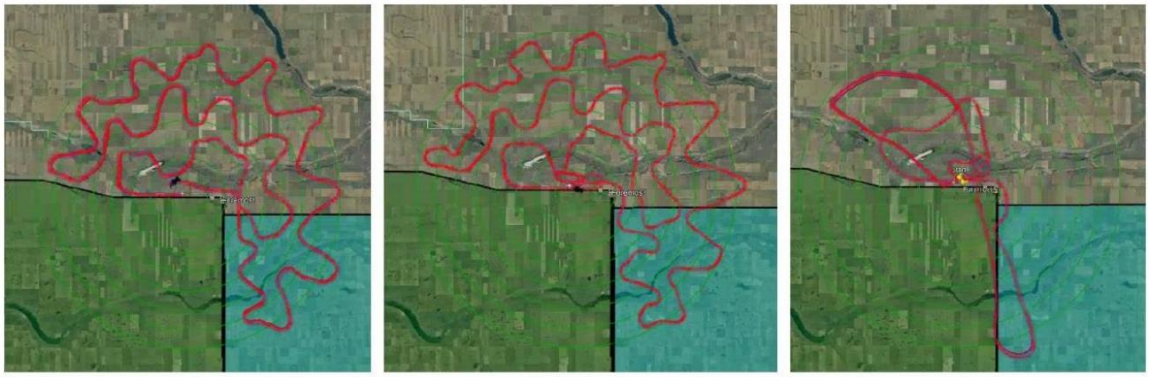
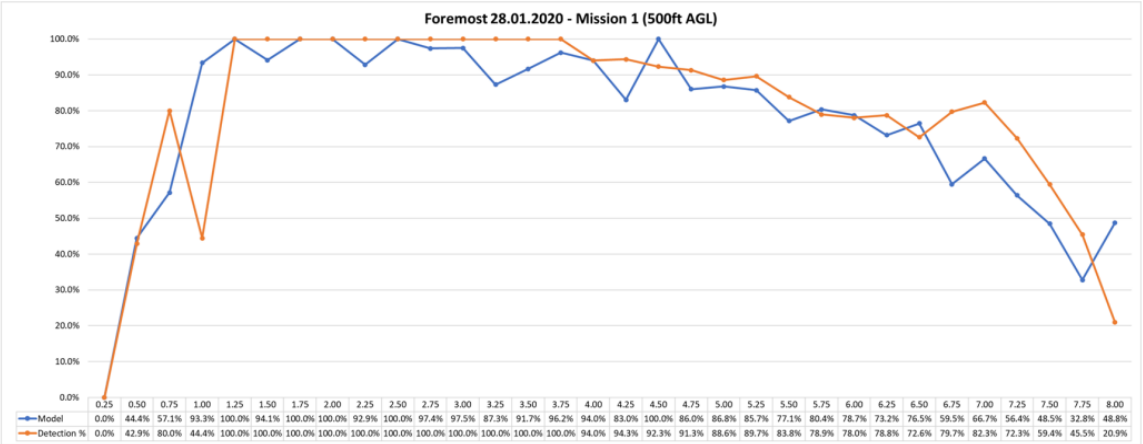


Figure 4. Cessna Flight Paths: 400ft AGL, 1400ft AGL, 5000ft AGL.

Flight test data were collected using cooperative intruder aircraft (Cessna 172 and RV-10) flown along structured radial and weaving trajectories to sample a wide range of encounter geometries and clutter conditions. Representative intruder flight paths at multiple altitudes are shown in Figure 4.



	0.25	0.50	0.75	1.00	1.25	1.50	1.75	2.00	2.25	2.50	2.75	3.00	3.25	3.50	3.75	4.00	4.25	4.50	4.75	5.00	5.25	5.50	5.75	6.00	6.25	6.50	6.75	7.00	7.25	7.50	7.75	8.00
Model	0.0%	44.4%	57.1%	93.3%	100.0%	94.1%	100.0%	100.0%	92.9%	100.0%	97.4%	97.5%	87.3%	91.7%	96.2%	94.0%	83.0%	100.0%	86.0%	86.8%	85.7%	77.1%	80.4%	78.7%	73.2%	76.5%	59.5%	66.7%	56.4%	48.5%	32.8%	48.8%
Detection %	0.0%	42.9%	80.0%	44.4%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	94.0%	94.3%	92.3%	91.3%	88.6%	89.7%	83.8%	78.9%	78.0%	78.8%	72.6%	79.7%	82.3%	72.3%	59.4%	45.5%	20.9%

Figure 5. Comparison of Modeled and Measured Detection Performance for Mission 1 at 500 ft AGL (Foremost, 28 January 2020).

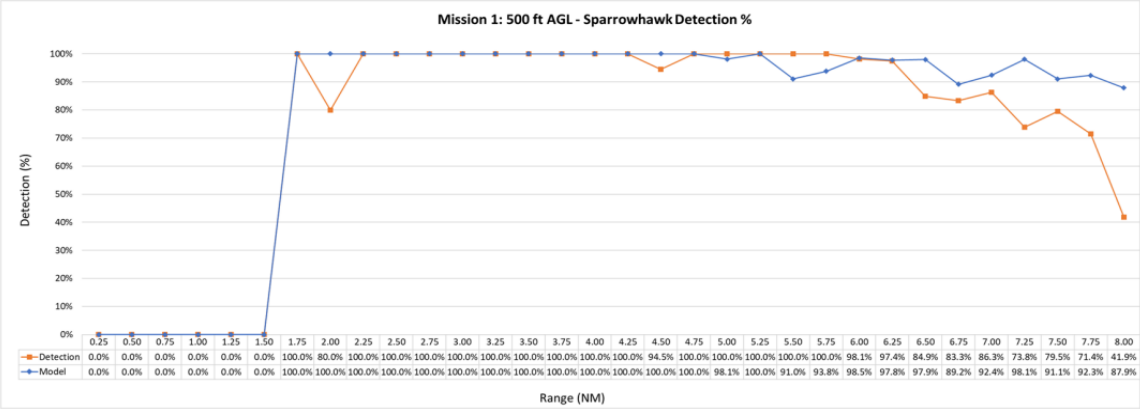


Figure 6. Mission 1 Sparrowhawk Detection Performance at 500 ft AGL.

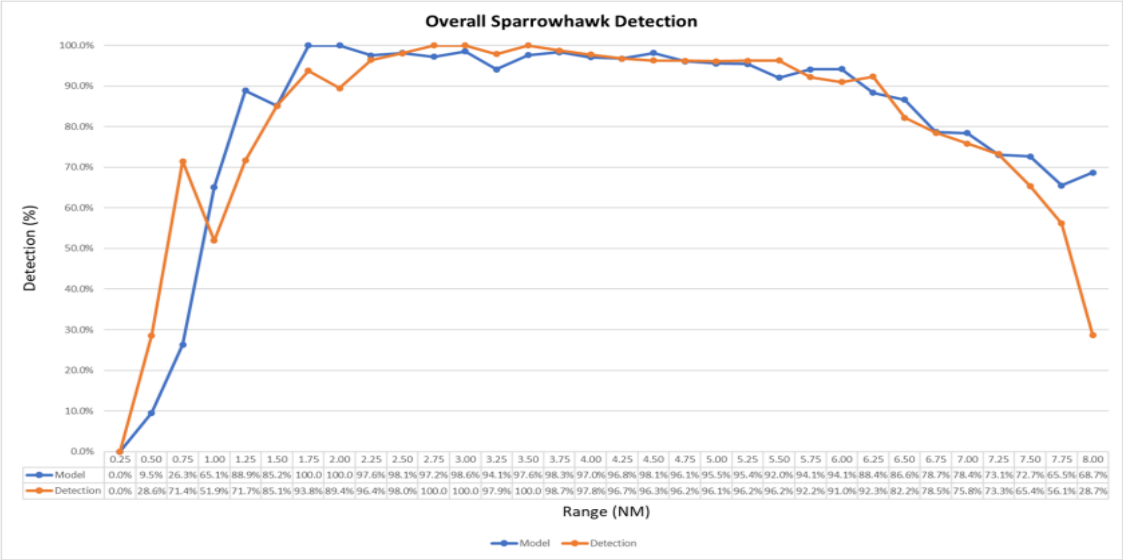


Figure 7. Overall Sparrowhawk Detection Performance Across All Missions.

Comparisons between modeled and measured detection performance (Figures 5, 6, and 7) show generally strong agreement across much of the operational range, with performance degradation emerging near the inner and outer boundaries of the detection envelope. These results demonstrate how targeted flight testing can validate model fidelity, specifically at envelope edges, where safety margins are most sensitive.

2.4 Position Error Analysis and Propagation Through the DAA Chain

Beyond detection probability alone, TCCA highlights the importance of propagating sensor position errors through the remainder of the operational DAA chain. As shown in Figure 8, horizontal positional errors on the order of 300–400 m increase with range and must be considered in downstream conflict assessment and avoidance decision logic.

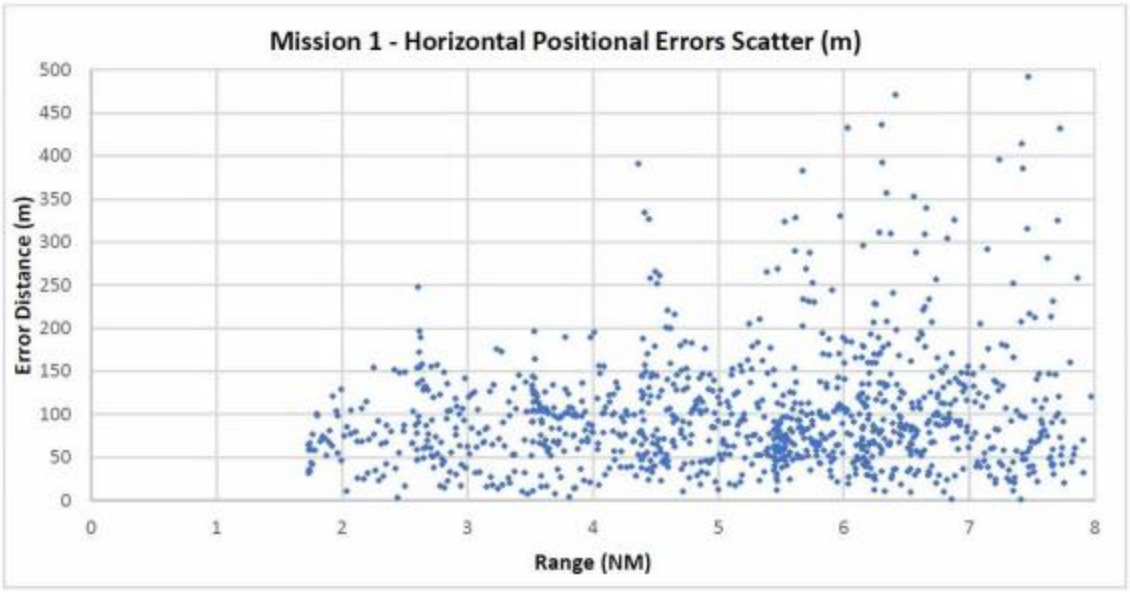


Figure 8. Mission 1- Horizontal Positional Errors Scatter.

This observation directly aligns with Task 3's emphasis on modeling detection timing uncertainty, sensor degradation, and environmental effects as contributors to overall DAA system risk, rather than treating detection performance as an isolated metric.

2.5 Implications of TCCA Guidance for A71 Task 3 Simulation Methodology

The TCCA guidance reinforces several foundational assumptions used in the A71 Task 3 simulation methodology:

TCCA Insight	Implication for A71 Task 3 Simulation
Focus on challenging edge cases, not only nominal operations	Monte Carlo encounter sets should emphasize early and late detection limits, clutter-heavy environments, and edge-of-range performance.
Model–test agreement is most important at extremes	Threshold optimization and cost-aware detection modeling must be evaluated under degraded visibility, long-range detection, and difficult intruder trajectories.
Terrain and environmental impacts must be modeled	A71 simulations include fog, lighting, clutter, and channel uncertainty, consistent with TCCA's recommendations.
Position errors affect the entire DAA chain	A71 risk models propagate detection uncertainty into conflict prediction, alerting, and missed-detection risk metrics.
No fixed flight-test quantity; validation is performance-driven	Simulation results can be anchored using selected representative validation flights rather than requiring exhaustive flight-test campaigns.

3 SIMULATION SETUP

As described preliminarily in the Task 2 report, the simulation environment was developed using a modular ROS2–Gazebo architecture in which DAA algorithms can be implemented or integrated. The environment can spawn the ownship in various cluttered scenarios, follow predefined trajectories, and support DAA operations with one or multiple intruders. The main modules of the simulation architecture are illustrated in Figure 9. Within this architecture, Gazebo provides high-fidelity, realistic sensor models, while ROS2 distributes data generated by the Gazebo models to other executables via topics, services, and actions.

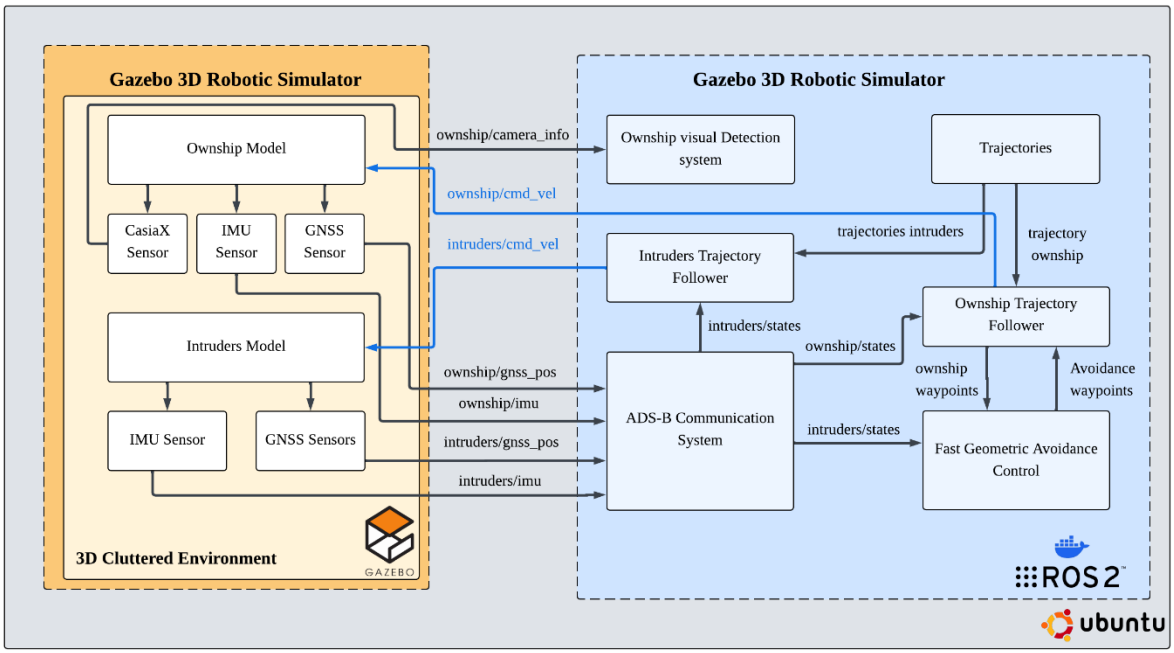


Figure 9. Simulation Structure.

The simulation environment is composed of the following elements:

- UAV dynamic models, including the ownship and intruders.
- Sensor plugins (IMU, cameras, and GNSS).
- ADS-B communication system.
- Visual intruder detection systems.
- Predicted Trajectories.
- UAV Trajectory Follower logic.
- Ownship Fast Geometric Avoidance Control.
- 3D Gazebo Cluttered Environment

In addition, the simulation currently runs in both Gazebo classic and modern Gazebo (Harmonic), which are suitable for the implementation of more environmental clutter conditions (fog) at high-rate simulations.

3.1 Unmanned Aerial Vehicle model

The UAV models in the simulation are spawned using a small fixed-wing Unified Robot Description Format (URDF) model divided into three subsystems. These subsystems define the UAV's 3D geometry, sensor configuration, plugin locations, and 3D meshes. For the simulation, four UAV configurations are available: (1) an ownship with a single camera, (2) an ownship with a 360° Casia X camera, (3) an ownship with a random number of cameras, and (4) an intruder. These four configurations are depicted in Figure 10.

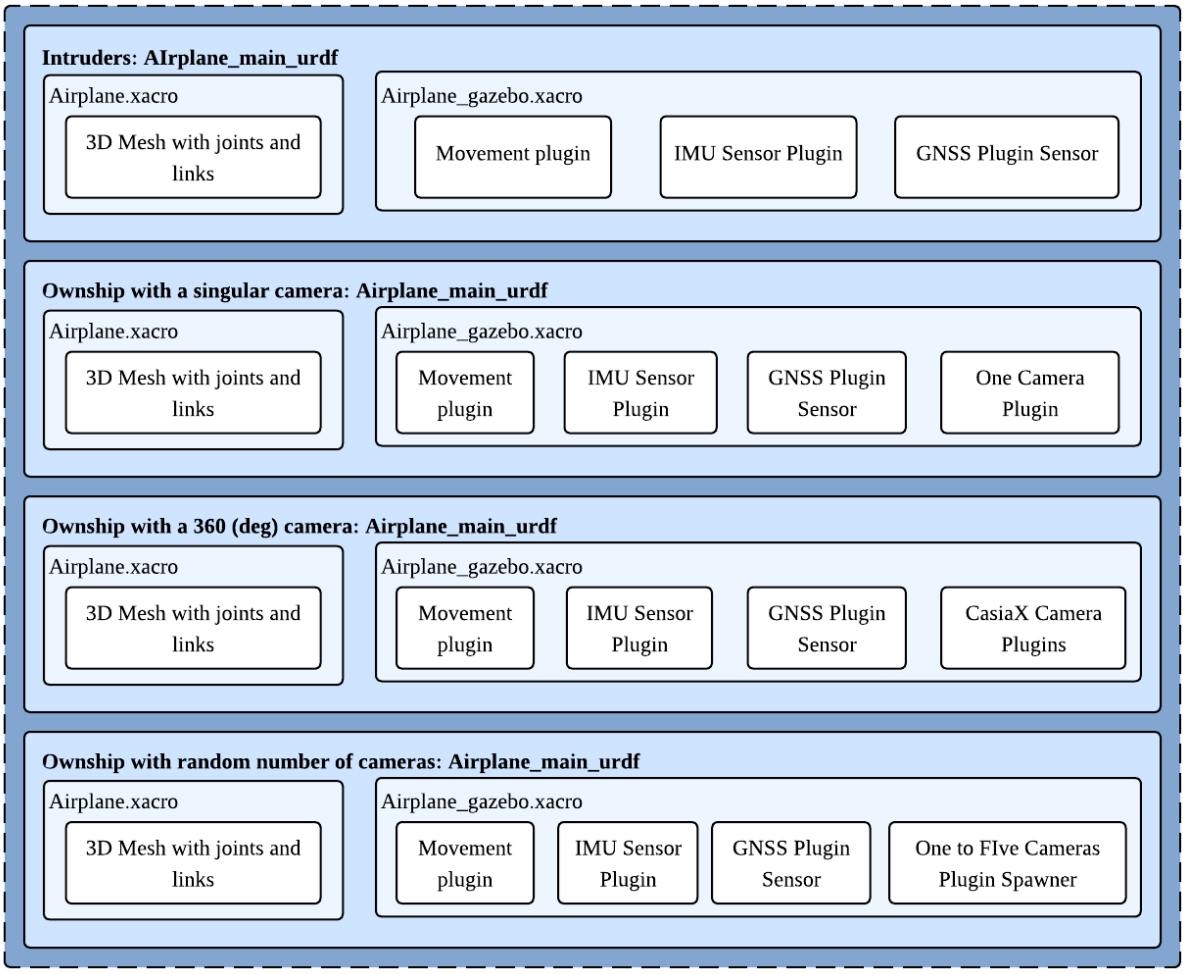


Figure 10. URDF UAV configurations in the Gazebo simulation.

Each UAV configuration is defined using a URDF/Xacro file that specifies the 3D model structure, sensor plugins (IMU, GNSS, and camera models), and the flight dynamics of a small fixed-wing UAV plugin. Each module is explained in detail in the following sections.

3.1.1 UAV 3D Model

As illustrated in Figure 11, the 3D model of the fixed-wing UAV is based on the PX4 Cessna 172 model, where all meshes are used to define the body of the UAV for both the intruders and the ownship [PX4, 2025].

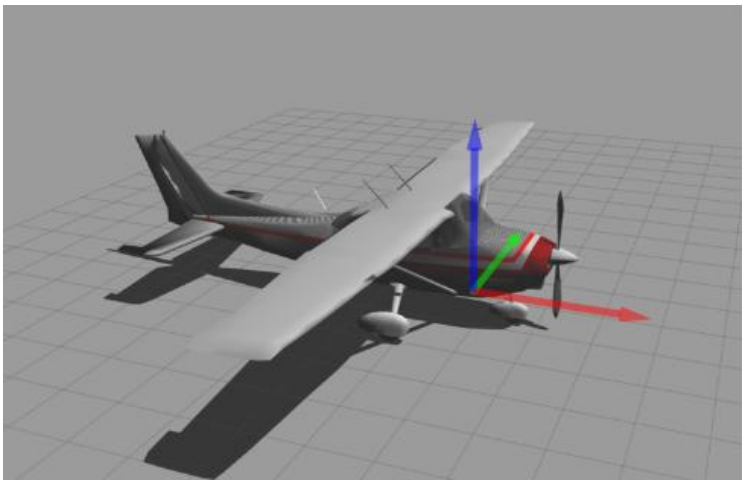


Figure 11. Gazebo Cessna 172 3D model.

3.1.2 UAV Kinematics

The UAVs use a movement plugin that publishes a *Twist* message to apply linear and angular velocity commands in the UAV model's body reference frame. At this stage, the UAV is represented using an autopilot-based proportional–derivative kinematic guidance framework, which generates the commanded velocities (V_{cmd}) from a commanded roll angle (ϕ_c), commanded airspeed (V_{ac}), and commanded altitude (h_c) provided by the trajectory follower. This kinematic approach neglects force and moment calculations and focuses only on the aircraft's kinematic behavior. In addition, the model assumes a coordinated turn with zero sideslip. The state vector is defined as ground speed (V_g), roll angle (ϕ), roll rate ($\dot{\phi}$), flight-path angle (γ), course angle (χ), airspeed (V_{ac}), north position (p_n), east position (p_e), and altitude (h) [Beard, 2012].

$$\dot{\vec{x}} = \begin{bmatrix} \dot{p_n} \\ \dot{p_e} \\ \dot{h} \\ \dot{\chi} \\ \dot{\gamma} \\ \dot{V_a} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} V_g \cos\chi \cos\gamma + \omega_n \\ V_g \sin\chi \cos\gamma + \omega_e \\ V_g \sin\gamma - \omega_d \\ \frac{g\cos(\chi-\psi)}{V_g} \tan\phi \\ k_\gamma(\gamma_c - \gamma) \\ k_{V_a}(V_{a_c} - V_a) \\ k_{p_\phi}(\phi_c - \phi) \\ k_{p_\phi}(\phi_c - \phi) - k_{d_\phi}(-\dot{\phi}) \end{bmatrix} \tag{1}$$

The commanded airspeed and commanded altitude are directly obtained from the UAV trajectory follower. However, the commanded roll is obtained with the roll limits and the course-roll relationship.

$$\chi_c = \text{atan}\frac{\Delta p_e}{\Delta p_n} \tag{2}$$

$$\phi_c = \text{atan}\frac{k_{p_\psi}(\chi_c - \psi)V_g}{g} \tag{3}$$

To reduce the risk of induced roll oscillations, the maximum absolute value of the commanded roll is limited to 25°.

3.1.3 Global Navigation Satellite System Plugin

The UAV positions are determined in Gazebo using a GNSS model based on the open-source multipath signal emulation presented in [Pant, 2022]. The plugin requires satellite Two-Line Element data to generate a custom satellite constellation and to simulate real-time satellite motion using the libpredict C++ library. In addition, the plugin estimates multipath signal reflections in urban environments using two Light Detection And Ranging (LiDAR)-based sensor rays emitted from the UAV: one ray connects the UAV to the satellite to detect surface reflections, while the other is cast perpendicular to the satellite LOS to estimate the multipath offset (m). This is illustrated in Figure 12.

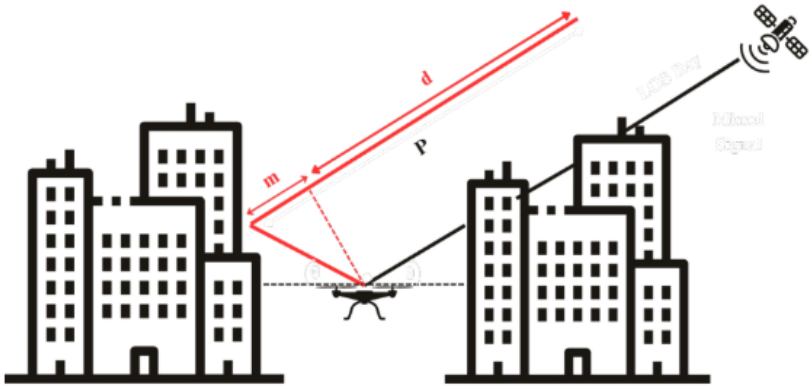


Figure 12. Multipath reflection of the LiDAR-sensor rays.

Within the plugin, when the ray toward a satellite detects a LOS obstruction, two cases are considered. If the estimated multipath offset is below a threshold (i.e., <100 m), it is added to the pseudorange P during position estimation. Otherwise, the satellite is treated as out of range and excluded from the detection process. The simulation considers only single-bounce reflections in m, since additional reflections have a negligible effect on the multipath offset [Ray, 1999]. For each visible satellite, the pseudorange P_i is computed using the true satellite range (ρ_i), the estimated multipath offset (m_i), the receiver clock bias ($c_0\Delta t$), and additive white measurement noise (e_i) with a standard deviation of 10 m [Pant, 2022].

$$P_i = \rho_i + m_i + c_o\Delta t + e_i \tag{4}$$

The pseudorange values of n visible satellites are equal to the distance between the satellite position $\overrightarrow{x_{sat}} = [x_i, y_i, z_i]$ and the receiver position $\vec{x} = [x, y, z, c_0\Delta t]$ with an estimated receiver time offset.

$$\begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ \vdots \\ P_n \end{bmatrix} = \begin{bmatrix} \sqrt{(x_1 - x)^2 + (y_1 - y)^2 + (z_1 - z)^2} + c_o \Delta t \\ \sqrt{(x_2 - x)^2 + (y_2 - y)^2 + (z_2 - z)^2} + c_o \Delta t \\ \sqrt{(x_3 - x)^2 + (y_3 - y)^2 + (z_3 - z)^2} + c_o \Delta t \\ \vdots \\ \sqrt{(x_n - x)^2 + (y_n - y)^2 + (z_n - z)^2} + c_o \Delta t \end{bmatrix} \tag{5}$$

Since the number of pseudorange measurements exceeds the four unknowns (three components of the receiver position and the clock bias), the receiver position is estimated by solving an overdetermined least-squares problem.

$$\Delta \vec{P} = H \Delta \vec{x} + \vec{e} \tag{6}$$

Where H is the negative Jacobian obtained from the pseudorange equations. The objective is to reduce the pseudoranges residual ($\Delta \vec{P}$), by iteratively updating the receiver state with $\Delta \vec{x}$ until its norm converges to a prescribed threshold or it reaches a maximum number of iterations that by default are 10.

$$H = \begin{bmatrix} \frac{x_1 - x}{\rho_1} & \frac{y_1 - y}{\rho_1} & \frac{z_1 - z}{\rho_1} & -1 \\ \frac{x_2 - x}{\rho_2} & \frac{y_2 - y}{\rho_2} & \frac{z_2 - z}{\rho_2} & -1 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{x_n - x}{\rho_n} & \frac{y_n - y}{\rho_n} & \frac{z_n - z}{\rho_n} & -1 \end{bmatrix} \tag{7}$$

The calculation of $\Delta \vec{x}$ uses a simple least squares algorithm:

$$\Delta \vec{x} = (H^T H)^{-1} H^T \Delta \vec{P} \tag{8}$$

In addition, the baseline GNSS plugin was modified to increase the number of DAA scenarios and to simulate failures or performance degradation in the navigation system during UAV flight. These modifications allow the user to attach the GNSS plugin to any model, specify a maximum number of visible satellites, and enable a dropout flag that simulates a GNSS outage in flight. When the dropout flag is active, the number of visible satellites is limited to three, which degrades the estimation accuracy; in particular, the least-squares position solution becomes ill-conditioned (and, in practice, may be underdetermined).

Finally, as part of the simulation environment, the GNSS plugin was updated to use LiDAR sensors to detect LOS obstructions and compute the multipath offset.

3.1.4 Automatic Dependent Surveillance–Broadcast Communication System (ADS-B) Model

Within the simulation, the ownship receives each intruder's current state through a modeled ADS-B system. Each intruder carries a virtual ADS-B transmitter that publishes an ADS-B Data message containing vehicle information such as identification, course, position, and linear velocity

in the Earth-Centered, Earth-Fixed (ECEF) reference frame. These transmitted states are estimated using an Extended Kalman Filter (EKF) that fuses IMU data with the GNSS position estimates to compute the UAV's position, orientation, and linear velocity at a specified update rate. This ADS-B architecture model is illustrated in Figure 13.

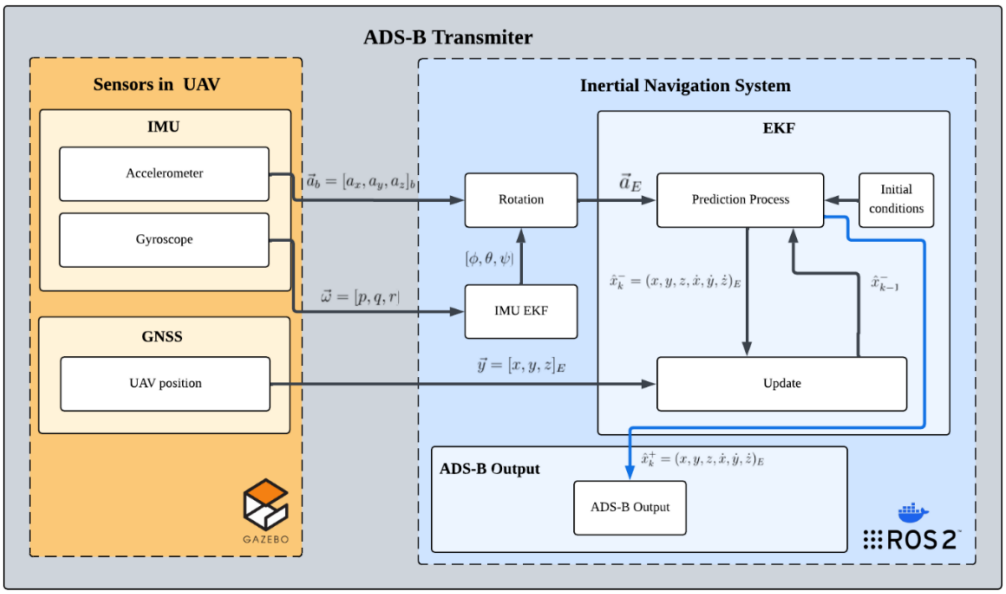


Figure 13. ADS-B UAV transmitter structure.

The estimation process begins with the UAV's attitude estimation, performed by an integrated EKF in the IMU sensor that uses body-frame accelerations and angular rates to estimate its orientation. That attitude estimation is then used to rotate the accelerometer measurements into the Earth frame and remove the gravity component, obtaining translational acceleration.

$$\begin{bmatrix} a_{x,E} \\ a_{y,E} \\ a_{z,E} \end{bmatrix} = \begin{bmatrix} \cos\psi\cos\theta & -\sin\psi\cos\phi + \cos\psi\sin\theta\sin\phi & \sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi \\ \sin\psi\cos\theta & \cos\psi\cos\phi + \sin\psi\sin\theta\sin\phi & -\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi \\ -\sin\theta & \sin\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix} \begin{bmatrix} a_{x,b} \\ a_{y,b} \\ a_{z,b} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad (9)$$

With acceleration now in Earth coordinates, the Inertial Navigation System (INS) EKF estimates position and velocity using the first-order discretization.

$$f(\vec{x},\vec{u}) = \vec{x}_{k+1} = \begin{bmatrix} x_{k+1} \\ y_{k+1} \\ z_{k+1} \\ \dot{x}_{k+1} \\ \dot{y}_{k+1} \\ \dot{z}_{k+1} \end{bmatrix} = \begin{bmatrix} x_k + \dot{x}_k\Delta t \\ y_k + y_k\Delta t \\ z_k + z_k\Delta t \\ \dot{x}_k + a_{x,E}\Delta t \\ \dot{y}_k + a_{y,E}\Delta t \\ \dot{z}_k + a_{z,E}\Delta t \end{bmatrix} \quad (10)$$

The estimation process of the EKF is divided into two main parts: a propagation of the estimation, and the measurement update. In the propagation step, the EKF uses the system model and its Jacobian to determine a priori estimates of the states and the error covariance matrix (P).

$$\vec{\hat{x}}_{k+1}^{-} = \begin{bmatrix} \hat{x}_{k+1}^{-} \\ \hat{y}_{k+1}^{-} \\ \hat{z}_{k+1}^{-} \\ \hat{\dot{x}}_{k+1}^{-} \\ \hat{\dot{y}}_{k+1}^{-} \\ \hat{\dot{z}}_{k+1}^{-} \end{bmatrix} = \begin{bmatrix} \hat{x}_{k}^{+} + \hat{\dot{x}}_{k}^{+}\Delta t \\ \hat{y}_{k}^{+} + \hat{\dot{y}}_{k}^{+}\Delta t \\ \hat{z}_{k}^{+} + \hat{\dot{z}}_{k}^{+}\Delta t \\ \hat{\dot{x}}_{k}^{+} + a_{x,E}\Delta t \\ \hat{\dot{y}}_{k}^{+} + a_{y,E}\Delta t \\ \hat{\dot{z}}_{k}^{+} + a_{z,E}\Delta t \end{bmatrix} \tag{11}$$

$$P_{k+1}^{-} = A_k P_k^{+} A_k^{T} + Q_k^{-} \tag{12}$$

Where (A_k) refers to the Jacobian of the discrete system, $(\hat{x})$ is the propagated estimated state, (P_k) is the propagated state error covariance matrix and Q_k is the process covariance matrix.

$$A_k \triangleq \frac{\delta \vec{\hat{x}}_{k+1}^{-}}{\delta \vec{\hat{x}}_{k}^{+}} = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{13}$$

In the update step, GNSS position measurements are selected to update the EKF estimated states, relying on its high precision but low update rate. Therefore, the measurement model is a linear system that directly depends on the estimated states.

$$h\left(\vec{\hat{x}}_{k+1}^{-}\right) = C\vec{\hat{x}}_{k+1}^{-} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{14}$$

Finally, EKF updates both the states and the error covariance matrix using the Kalman gain (K_k) and the measurements.

$$K_k = P_{k+1}^{-} C^T (C P_k^{-} C^T + R)^{-1} \tag{15}$$

$$\vec{\hat{x}}_{k+1}^{+} = \vec{\hat{x}}_{k+1}^{-} + K_k(\tilde{y}_{GNSS} - C\vec{\hat{x}}_{k+1}^{-}) \tag{16}$$

$$P_{k+1}^{+} = (I - K_k C) P_{k+1}^{-} \tag{17}$$

Where R is the measurement covariance matrix and $\tilde{y}_{GNSS}$ represents the GNSS position estimates in the ECEF. During the simulations, the INS constantly feeds the estimated states to the ADS-B, so the ownship DAA logic tracks the intruders at every time step.

3.1.5 DAA CasiaX Model

The Gazebo camera sensor plugin allows the user to simulate a wide range of camera specifications, including image geometry (pixel width and height), pixel format (RGB, L8, BGR, etc.), horizontal and vertical fields of view, update rate, mounting position on the UAV, distortion

coefficients (k1, k2, k3, p1 and p2), white pixel noise (Gaussian noise applied to each pixel), lens intrinsics, lens models, field-of-view scaling, maximum visual range, and camera type (RGB or depth) [Open Robotics, 2025].

Within the simulation environment, the ownship's visual detection system is designed to emulate industry-level DAA configurations, such as the CasiaX DAA system. CasiaX is a 360° multi-camera DAA system that uses computer vision and machine learning to detect non-cooperative aerial vehicles and help the pilot maintain close proximity to intruders [uAvioni, 2024].

To mimic the CasiaX configuration, the simulated sensor replicates the DAA camera characteristics by modeling five high-quality 8.9-megapixel GigE (Gigabit Ethernet) cameras with 8 mm low-distortion lenses, a resolution of 4200 \times 2160 pixels with a Bayer GB8 pattern, an individual horizontal field of view of 80°, a clip range of 2,096 m, and an update rate of 13 frames per second.

For the ownship 3D model, the five cameras are mounted with a 72° azimuthal separation and positioned at the extremities of the airframe to minimize occlusion. As shown in Figure 14, this configuration provides a 360° horizontal field of view with approximately 40° overlap between adjacent cameras. Before detection, the five camera views are stitched into a single image by trimming 7° from each horizontal field of view to improve stitching smoothness and continuity. This preprocessing step reduces the number of frames that must be processed and helps prevent duplicate intruder detections within the same stitched image.

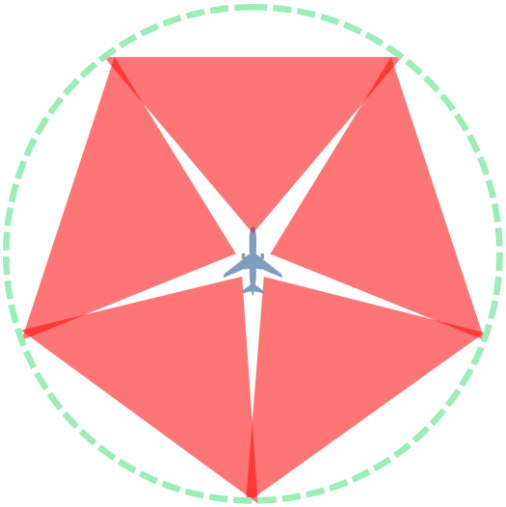


Figure 14. Casia X camera configuration in the UAV frame.

3.1.6 Visual Detection System

Because Casia X's visual detection system relies on a patented deep-learning classifier that is available only to direct customers, our simulation instead uses open-source visual detection methods that have already been trained for fixed-wing UAV detection. Specifically, the simulation employs the You Only Look Once (YOLO) framework, originally introduced by Joseph Redmon et al. [Redmon, 2016]. YOLO is a dense-prediction approach that performs real-time object

detection by predicting bounding boxes and class probabilities directly from full images in a single forward pass through one neural network. It combines a convolutional feature extractor that converts images into feature maps with detection heads that estimate class probabilities and bounding-box parameters for individual objects.

Over the past several years, YOLO has undergone multiple updates to improve detection efficiency and single-model, single-scale performance on the COCO val2017 dataset, a large-scale dataset designed for computer-vision training and evaluation [Lin, 2014]. The simulations presented in this report use the more recent YOLOv11 variant, as it provides higher confidence levels while maintaining lower processing times. Figure 15 presents the YOLO image detection structure used within the CasiaX DAA module.

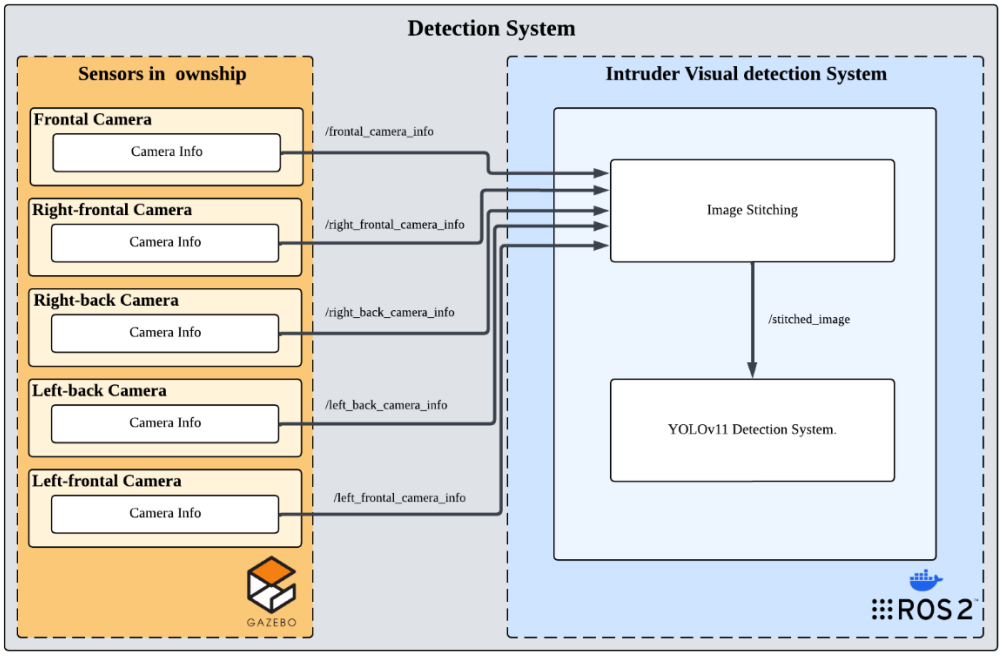


Figure 15. Simulation YOLO image detection structure.

YOLOv11 is available in multiple variants, ranging from the Nano to the X version. Higher-capacity variants increase the floating-point operations, which can improve detection performance but typically at the cost of longer inference times. During the simulations, the YOLO models demonstrated limitations in reliably detecting intruders in cluttered environments, such as urban settings, particularly when requiring confidence scores above 0.7. Therefore, multiple variants were evaluated to identify the best trade-off between detection performance and detection rate.

In YOLO, efficiency is commonly assessed using detection confidence together with preprocessing, inference, and postprocessing time. Preprocessing is the time required to resize the input image and normalize pixel values so the model can process it. Inference is the most critical step for real-time applications, as it represents the time required for YOLO to run the network and generate predictions. Postprocessing corresponds to refining the raw detections by removing redundant outputs (e.g., via non-maximum suppression) and applying the confidence threshold

[Redmon, 2018]. Table 1 summarizes the intruder detection distance for different YOLOv11 models and the average detection times in ideal conditions.

Table 1. Maximum intruder detection distance for different YOLOv11 models.

YOLO Model	Yolo11n	Yolo11s	Yolo11m	Yolo11l	Yolo11x
Ave. Maximum Distance (ft)	919	1300	1360	1364	1374
Ave. Preprocessing time (ms)	2.4	3.75	5.91	6.4	10.5
Ave. Inference time (ms)	5.51	7.76	16.08	21.21	36.72
Ave. Postprocessing time (ms)	1.3	1.18	1.23	0.78	0.83

Based on Table 1, the higher-capacity models of YOLOv11 converge to a maximum detection distance of approximately 1360 (ft) under ideal conditions (clear view, no occlusion, and without camera noise). Therefore, the YOLOv11m was chosen due to its high maximum detection distance with a moderate processing time.

3.1.7 Trajectory Follower

The trajectory follower was implemented onboard each UAV and was designed to track a trajectory defined by East, North, and altitude coordinates, together with a commanded airspeed (V_{cmd}). The system first determines the next waypoint to follow based on the UAV's current position and the corresponding hyperplane region [Park, 2004]. If the dot product is negative, the follower keeps tracking the current waypoint (W_2). Once it becomes positive, the follower switches to the next waypoint (W_3). Then, it calculates the commanded heading and altitude, expanding the projected vector of the UAV in the current waypoint connection with a user-defined fixed lookahead distance (d).

In addition, the trajectory follower employs a minimum switching radius (r_{min}), so that the controller advances to the next waypoint when the UAV is near and avoids limit-cycle oscillations near tracking points. This concept is illustrated in Figure 16.

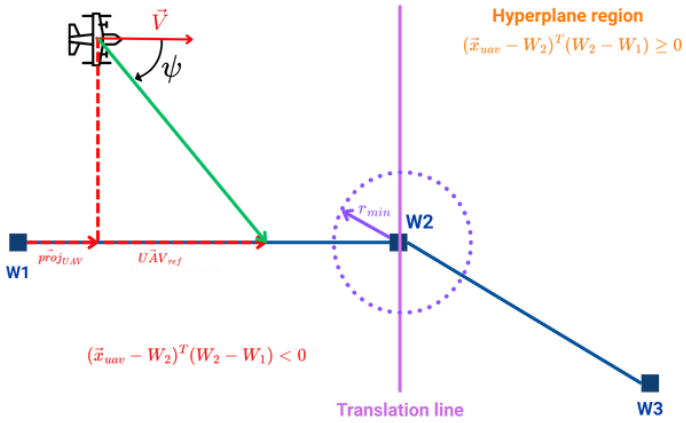


Figure 16. Hyperplane region example.

3.1.8 Fast Geometric Avoidance Control

In the simulation, the ownship avoidance trajectories are generated using the Fast Obstacle Collision Avoidance Algorithm proposed by Zijie Lin et al. [Lin, 2018]. The method is designed to rapidly create collision-free, optimized waypoints around circular avoidance zones attached to obstacles and UAV intruders. First, the algorithm computes the minimum safe avoidance radius (d_m) based on the obstacle-velocity margin (d_{mv}), intruder/vehicle-velocity margin (d_{mi}), vehicle-dimensions margin (d_{ma}), and airspace limitations (d_{lim}).

$$d_m = k_1 d_{mv} + k_2 d_{mu} + k_3 d_{ma} + k_4 d_{\text{lim}} \tag{18}$$

Where the constant k_i depends on the safety factor that the user wants to apply in the avoidance maneuver. Then, the minimum future distance is calculated using the relative velocity ($\vec{v}_{rel}$), and the relative position ($\vec{r}_{rel}$) between each intruder and the ownship UAV.

$$r_{min} = \|\vec{r}_{rel}\| \sin\left(\text{acos}\left(\frac{\vec{r}_{rel} * \vec{v}_{rel}}{\|\vec{r}_{rel}\| \|\vec{v}_{rel}\|}\right)\right) \tag{19}$$

An intruder becomes active when r_{min} is smaller than the respective d_m. If there is more than one intruder, the algorithm checks if any of the other obstacles overlap inside the projected collision zone of the ownship. If there is at least one, the algorithm combines the avoidance zones of the implicated intruders. This overlap is illustrated in Figure 17.

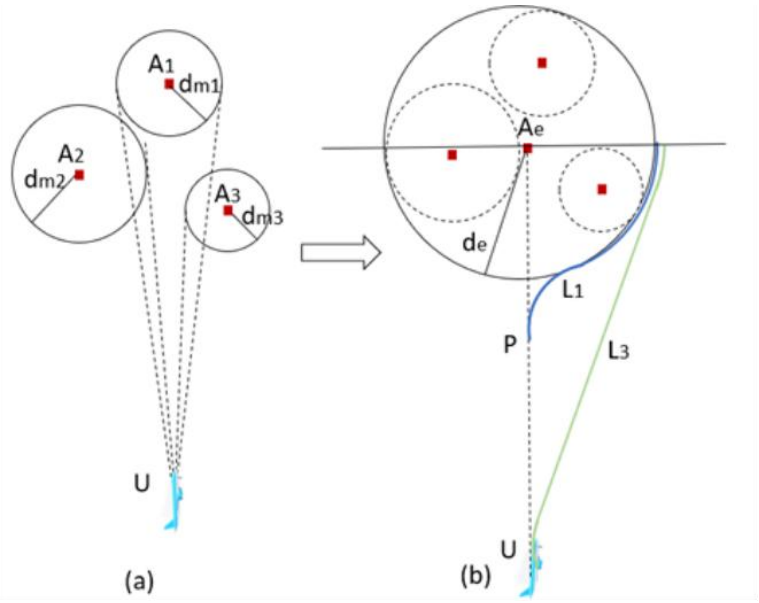


Figure 17. Multiple 2D obstacle avoidance in case of overlap [Lin, 2018].

The same logic applies to 3D, where the algorithm, instead of using circles, uses spheres to define the intruder avoidance zone. This is illustrated in Figure 18.

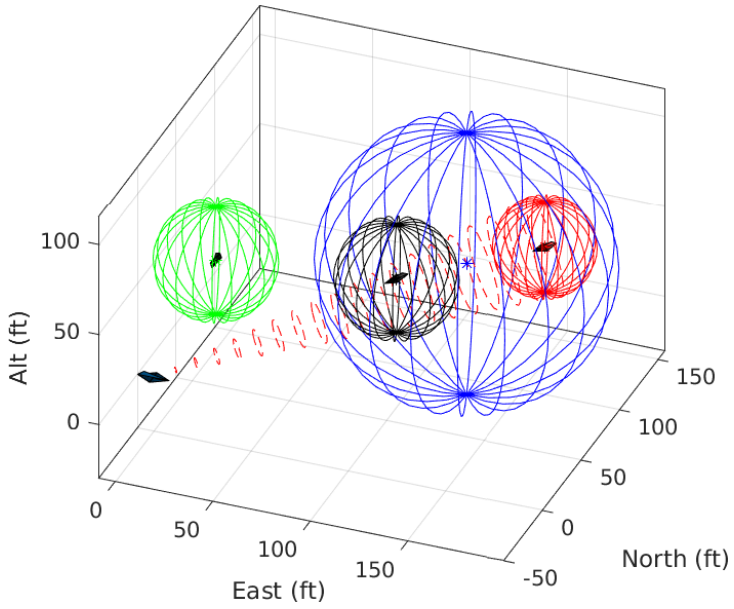


Figure 18. Multiple 3D obstacle avoidance in case of overlap.

The critical time is the ownship remaining time to safely avoid the intruder, considering d_m, the relative distance, the vehicle's velocities, and the ownship minimum turn radius (ρ_{min}).

$$t_c = \frac{\|\vec{r}_{rel}\| \cos\theta_{t_0} - \rho_{min} - (d_m - \|\vec{r}_{rel}\| \sin\theta_{t_0})}{\|\vec{v}_{rel}\|} \quad (20)$$

To determine the optimized avoidance zone, the algorithm finds the principal point in a perpendicular circle between the ownship velocity and the active avoidance zone. This principal point is used to define a circular trajectory, which is then added to the current set of waypoints followed by the ownship. The principal point is optimized using a cost function that considers the changes of course and flight path angle needed to avoid the intruder.

$$cost = \eta_1|\psi_{com} - \psi_0| + \eta_2|\gamma_{com} - \gamma_0| \tag{21}$$

Where the weighting coefficients (η_1 and η_2) are parameters related to the angle between the ownship and active intruder velocity to prioritize the avoidance opposite to the intruder movement. Additionally, the cost of longitudinal maneuvers is increased by a constant k_1 to reduce the risk that the UAV stalls or invades restricted airspace.

$$\eta_1 = \frac{1}{\alpha+0.5} \tag{22}$$

$$\eta_2 = \frac{1}{\alpha+0.5} + k_1 \tag{23}$$

The spherical avoidance zones resulted in unrealistic longitudinal maneuvers, so if the ownship chooses to avoid longitudinally, it could potentially stall or enter restricted flight altitudes. To address this limitation, intruder avoidance zones can be represented as ellipses, in which the single distant characteristic d_m is replaced by two parameters: the minimum safe semi-major length (a_m), and semi-minor length (b_m).

$$a_m = k_1 a_{mv} + k_2 a_{mu} + k_3 a_{ma} + k_4 a_{\text{lim}} \tag{24}$$

$$b_m = k_1 b_{mv} + k_2 b_{mu} + k_3 b_{ma} + k_4 b_{\text{lim}} = d_m \tag{25}$$

Here, the calculations of both t_c and r_{min} use the same equations with a small modification d_m is replaced using a specified minimum horizontal radius ($d_{m,s}$) equal to a scaled value of b_m with respect to the ratio between the intruder's relative altitude and a_m.

$$d_{m,s} = \left| b_m \left(\frac{r_{rel}[3]}{a_m} \right) \right| \tag{26}$$

In the presence of more than one obstacle, the same overlapping logic is applied for the ellipsoid intruder avoidance zones, as illustrated in Figure 19.

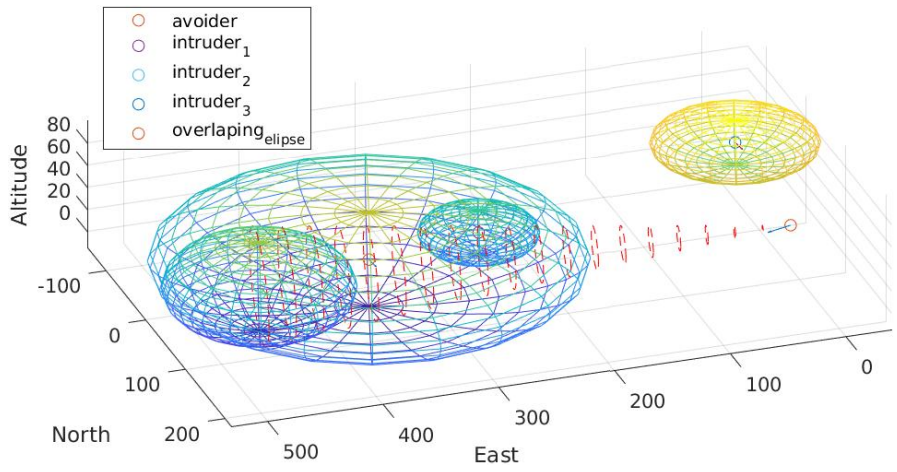


Figure 19. Multiple 3D obstacle avoidance using ellipsoids in case of overlap.

Finally, as illustrated in Figure 20, the avoidance trajectory waypoint generation uses the same cost function formulation to identify the optimal avoidance principal point.

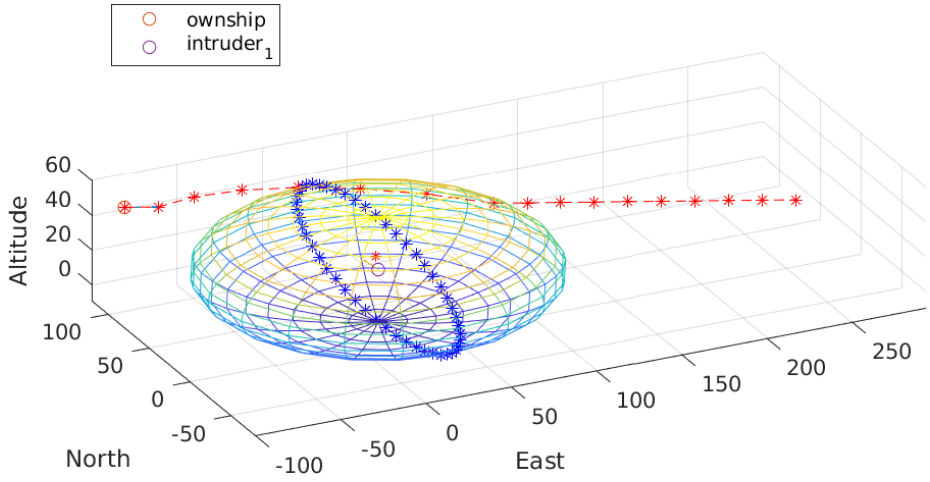


Figure 20. Obstacle avoidance waypoint generation using the cost function.

4 MONTE CARLO SIMULATIONS

This section describes how the simulation environment was used to evaluate the CasiaX DAA visual detection system across a range of flight conditions and operational scenarios. Preliminary simulations were also included to evaluate the performance of the ADS-B DAA system. Because most UAV encounters in the MIT dataset occur at low altitudes within Class D airspace, visual perception performance is strongly influenced by background clutter. Complex scenery can partially occlude intruders, increase image-processing latency, and trigger false-positive detections. To capture these effects, the simulation includes four representative visual environments from different regions of the United States: a 5 mi^2 area of the Sonoran Desert, a 5 mi^2 area centered on Times Square, a 5 mi^2 area in the Rocky Mountains, and a 20 mi^2 area surrounding Hanscom Air Force Base. These scenarios are shown in Figure 21.

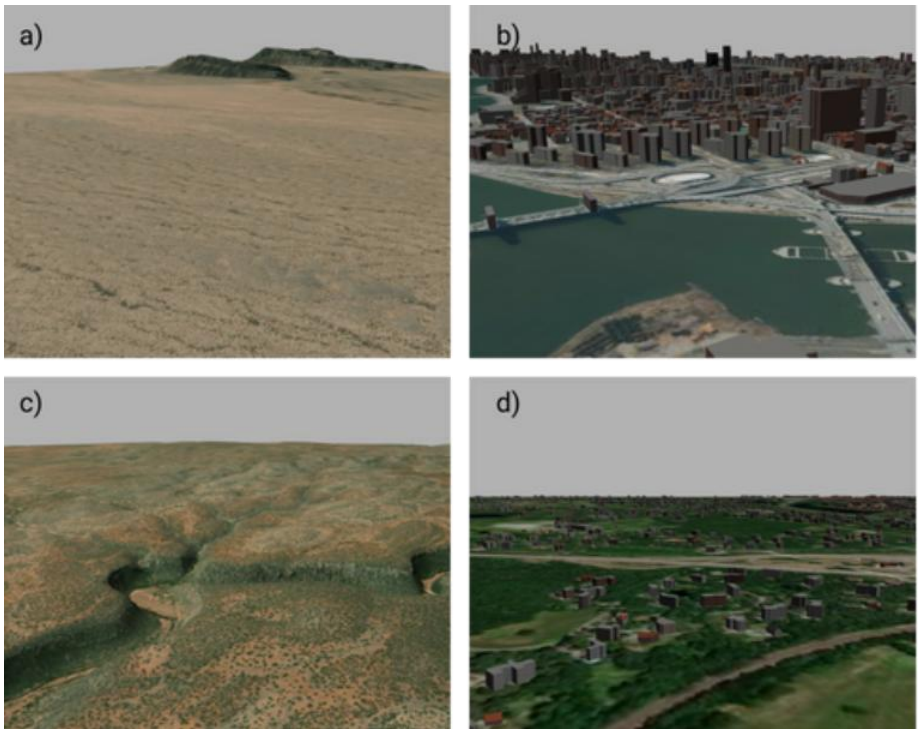


Figure 21. Real-world environments simulated in Gazebo: (a) Sonoran Desert, (b) Times Square, (c) Rocky Mountains, and (d) Hanscom Air Force Base surroundings.

Monte Carlo studies were then conducted under three configurations. First, MIT Encounter Set Simulations replay encounter trajectories near the CPA to quantify intruder detection confidence across varying fog levels. Second, MIT Encounter Set Simulations with Random Cameras repeat the same CPA-focused encounters and fog conditions while assessing detection performance under partial sensor failure, where one or more cameras in the CasiaX configuration are unavailable. Third, Detect-and-Avoidance ADS-B Preliminary Simulations evaluate end-to-end DAA scenarios in which GNSS outages are introduced via a dropout flag during the avoidance maneuver. The setup and results for each configuration are presented in the following subsections.

4.1 MIT Encounter Set Simulations

The MIT Encounter Set simulations consist of 1,000,000 runs of different encounter set trajectories near the CPA. Each flight was initialized at a random position along the trajectory of the encounter set and propagated for at least 75 sec, or until the trajectory is completed. In addition, to analyze the effect of visual clutter during the intruder detection process, the simulations are performed in two different environmental clutter conditions (empty world and Hanscom Air Force Base surroundings), and four different levels of exponential fog. In the simulation, the exponential fog represents an exponential decay of the visibility parameter ($\alpha(z)$) with respect to the distance between the objects (z) and a fog density (ρ) [Ogre Team, 2025].

$$\alpha(z) = e^{-\rho z} \tag{27}$$

As depicted in Figure 22, the four levels of fog used during the simulations are: no fog (ρ =0), minimum fog (ρ =0.001), medium fog (ρ =0.005), and high fog (ρ =0.01).

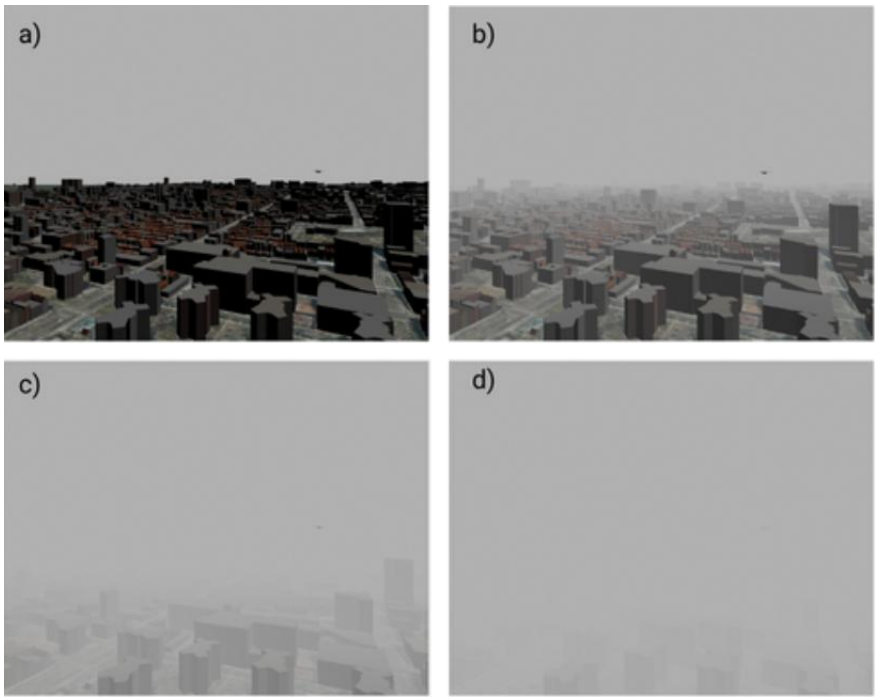


Figure 22. Fog levels applied in a New York city environment in Gazebo: a) No fog, b) Minimum fog, c) Medium fog, and d) High fog.

Baseline confidence detections were obtained in an uncluttered environment under different fog conditions, where the only models present are the two UAVs: the ownship and the intruder. For example, Figure 23 shows results near the CPA for Encounter Set 6160 at varying fog levels. As fog intensity increases, the number of detections decreases. Higher-confidence detections occur at shorter separation distances, and detection density increases when the two UAVs are in close proximity. In particular, this scenario illustrates that medium and high fog levels can fully occlude the intruder, preventing detection even at short distances.

Overall, the reference confidence detection distributions of the 1,000 MIT encounter set simulations can be expressed by transferring the detections with its respective intruder relative positions to the ownship body Reference Frame. Figure 24 illustrates the distribution of intruder detection confidence. As fog density increases, the maximum detection range decreases; detection rates rise at shorter separation distances, and the highest-confidence detections occur predominantly near the ownship. The overall detection distributions under no-fog and minimal-fog conditions are similar. In some cases, low fog combined with short separation distances can partially obscure the intruder's shape while also reducing background clutter, which may improve detection performance. This is also visible in the examples shown in Figure 25.

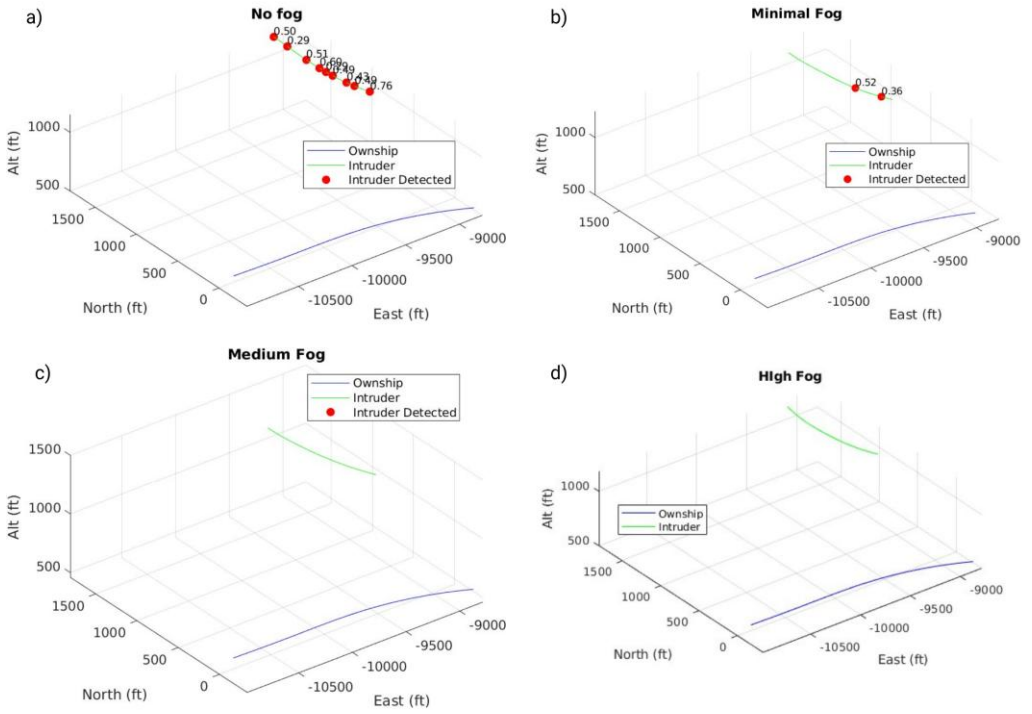


Figure 23. Detections for MIT encounter set 6160 at different levels of fog: a) No fog, b) Minimum fog, c) Medium fog, and d) High fog.

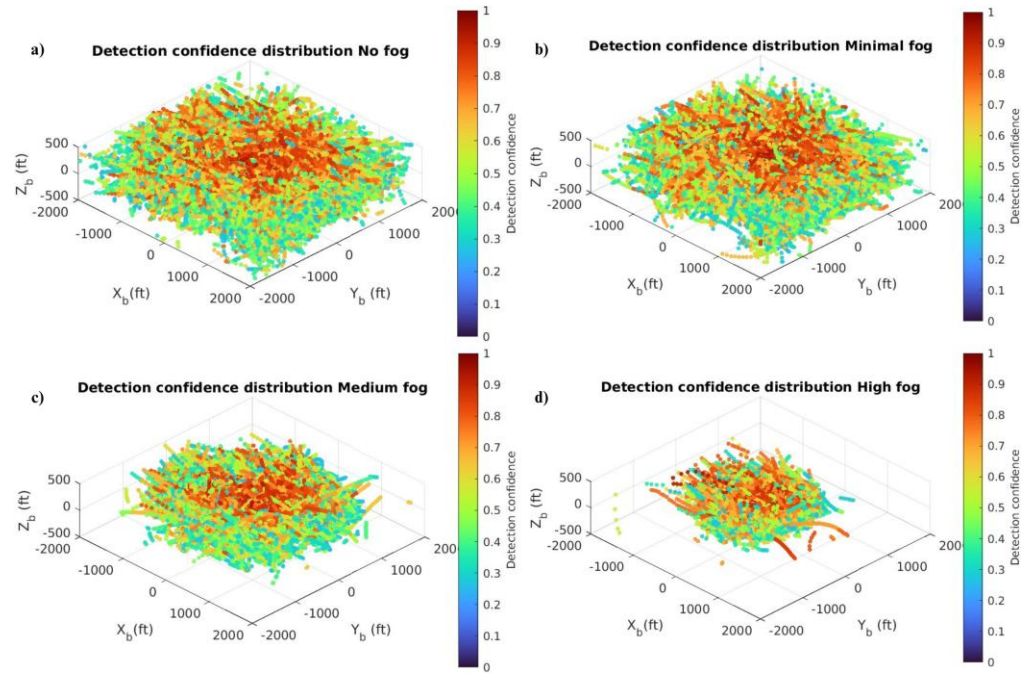


Figure 24. Reference Confidence Detections distributions with respect to the relative position of the intruder for the 1,000 MIT encounter sets at different levels of fog: a) No fog, b) Minimum fog, c) Medium fog, and d) High fog.

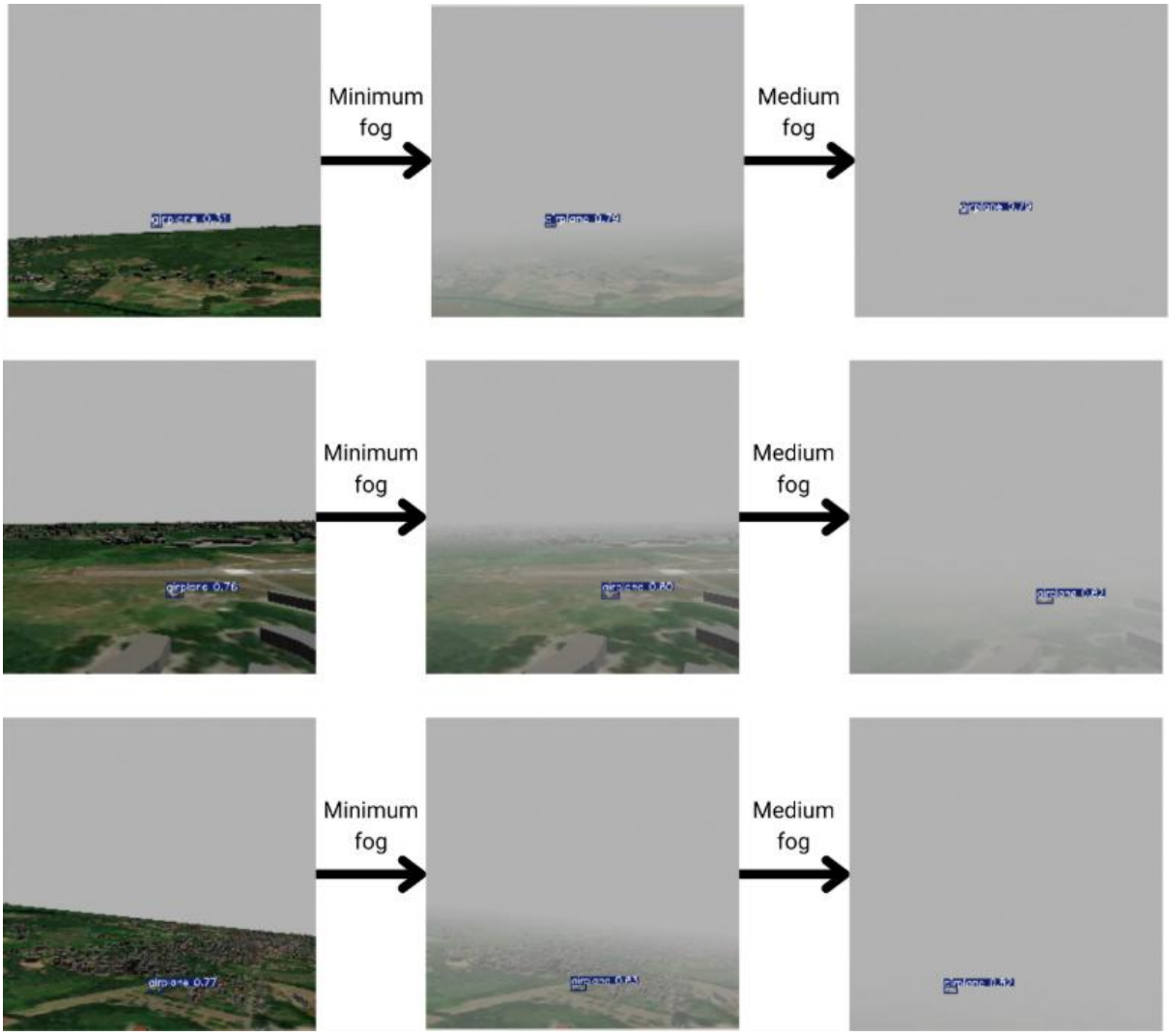


Figure 25. Examples of how fog increases the detection confidence due to a reduction of clutter and UAV shape shadowing.

A second set of simulations was conducted in a highly cluttered visual environment representative of the Hanscom Air Force Base surroundings. Intruder detections for the MIT encounter set 6160 were first compared against the uncluttered reference case. Relative to the results in Figure 24, Figure 26 shows that the Hanscom Air Force Base runs yield fewer detections overall and lower detection confidence, indicating a clear degradation in visual performance under high background complexity. The only notable improvement occurs under minimal-fog conditions, where shadowing can enhance intruder contrast and produce better detections than in the empty-world scenario.

The confidence distributions with respect to the ownship position across the 1,000 Hanscom Air Force Base simulations are shown in Figure 27. Similar to the empty-world confidence distributions, Figure 27 indicates that detection performance degrades under high fog conditions, detection percentages increase at shorter separation distances, and, under certain minimal-fog cases, detection performance can improve. These distributions enable a direct comparison of the

average confidence levels obtained in the Hanscom Air Force Base environment against the corresponding reference results.

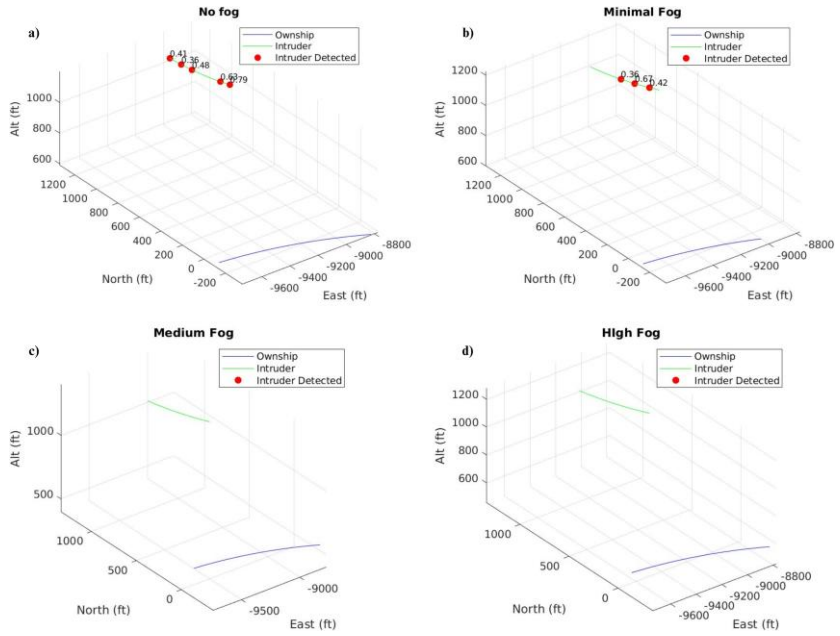


Figure 26. Detections done in the Hanscom Air Force Base during the MIT encounter set 6160 at different levels of fog: a) No fog, b) Minimum fog, c) Medium fog, and d) High fog.

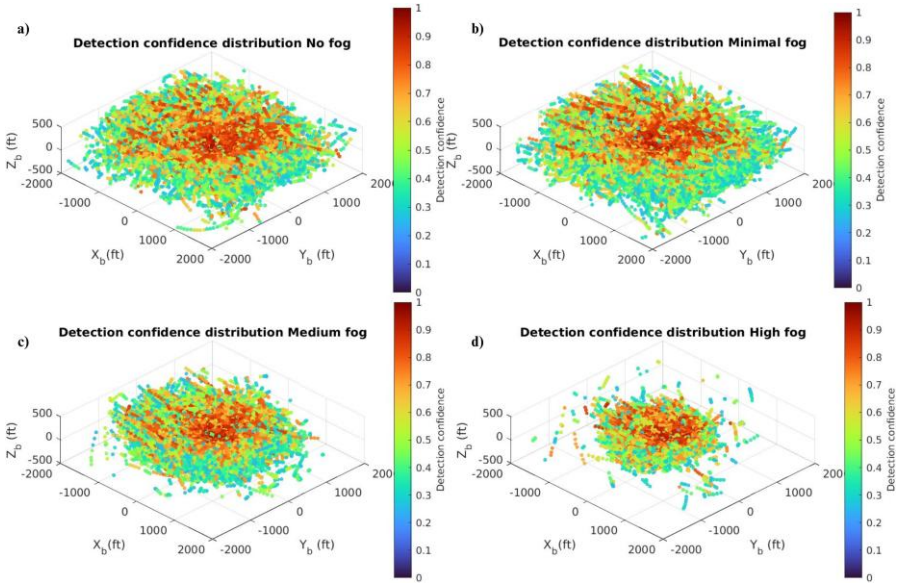


Figure 27. Confidence Detections distributions in the Hanscom Air Force Base with respect to the relative position of the intruder for the 1,000 MIT encounter sets at different levels of fog: a) No fog, b) Minimum fog, c) Medium fog, and d) High fog.

Figure 28 illustrates the impact of the cluttered Hanscom Air Force Base environment on intruder detection. Compared with the reference case, the cluttered scene reduces the number of detections across all fog conditions and lowers the average detection confidence. The results in Figure 28 also include filtered detections that remove most false positives by increasing the minimum confidence threshold (< 0.7) and discarding detections from cameras that are not oriented toward the intruder. Overall, these findings indicate that the high level of visual clutter in the Hanscom Air Force Base environment can induce detection dropouts and failures during the DAA maneuver.

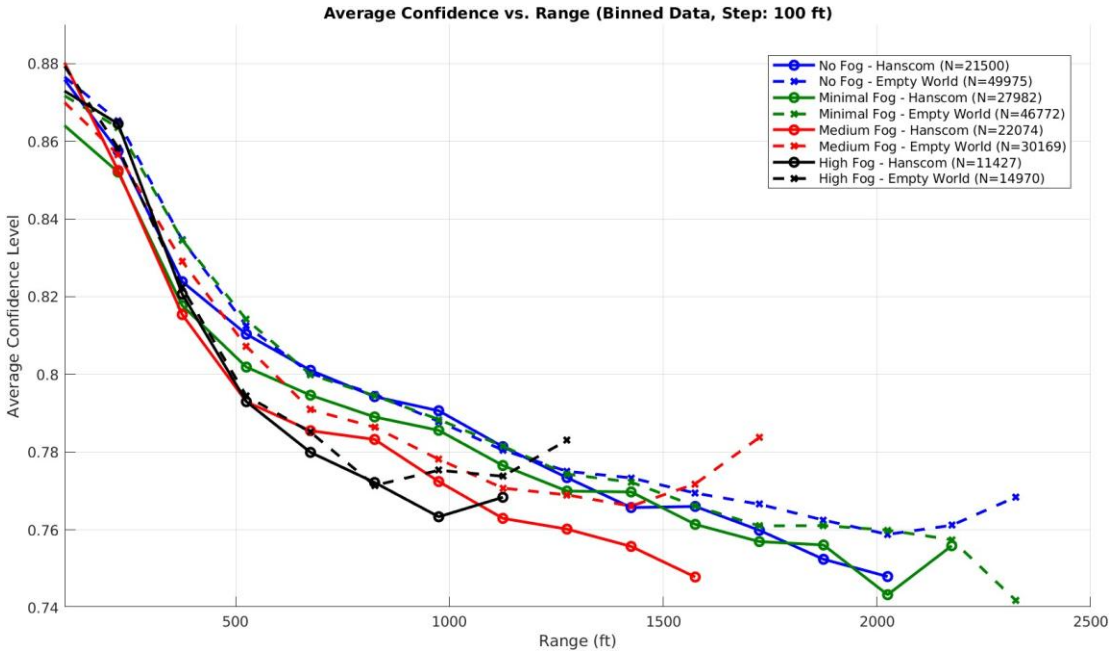


Figure 28. Average confidence detection levels vs range for different fog conditions in the Hansom Air Force Base and Reference simulations.

Table 2 demonstrates how the high clutter in the Hanscom Air Force Base simulations induce to intruder false positive detections with false positive percentages around 23% only considering intruder detections that were higher than the defined detection threshold (0.7).

Table 2. Percentage of False Positive detections in the Hanscom Air Force base simulations.

Conditions	**Hanscom Base clutter**		**Percentage false positive detections**
	Total detections	**False positive detections**	
No fog	27990	6490	23.19%
Minimal fog	38400	10418	27.13%
Medium fog	28978	6904	23.82%
High fog	14926	3499	23.44%

The distribution of detection ranges across fog conditions and clutter types is presented in Figure 29. In the Empty World environment, mean detection ranges decrease progressively from 1335 ft (no fog) to 724 ft (high fog), demonstrating the substantial impact of visibility degradation on

detection capability. Notably, the distributions become increasingly narrow and left-shifted as fog density increases, indicating both reduced detection range and lower variability in detection distances under degraded visibility conditions.

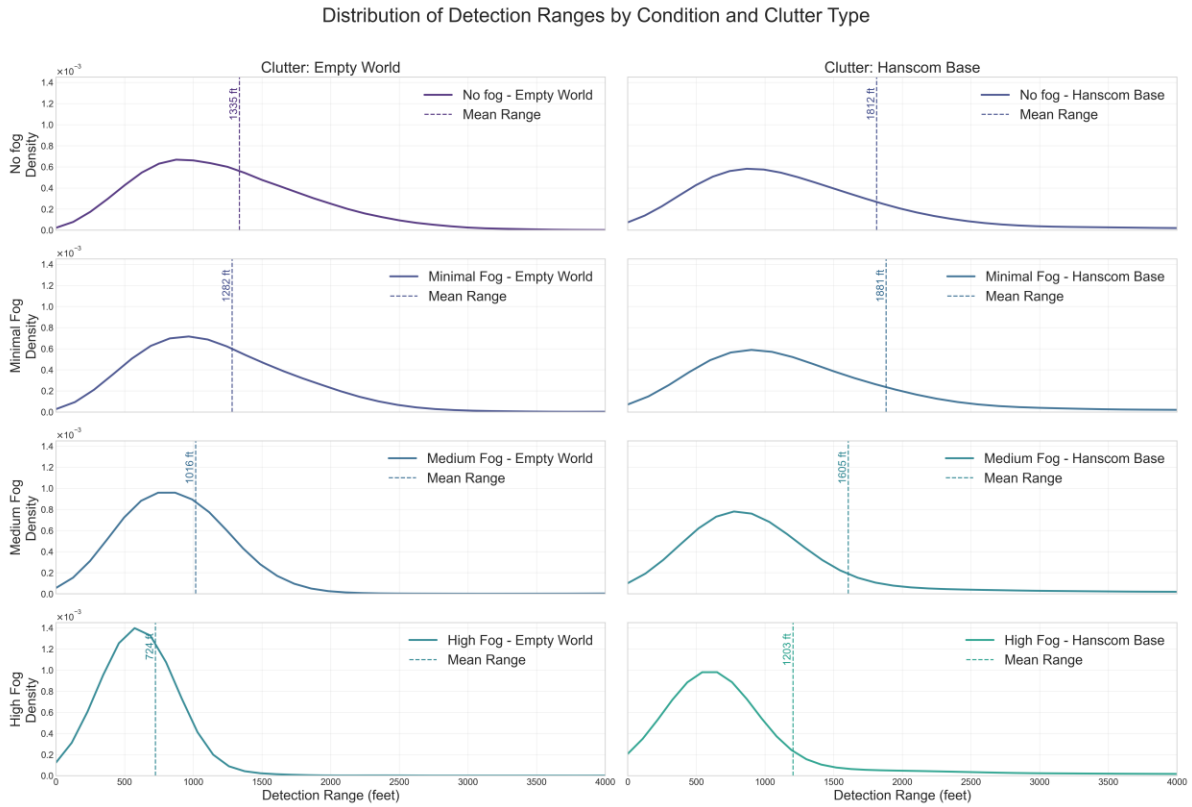


Figure 29. Distribution of Detection Ranges by Condition and Clutter Type for the 1,000 MIT encounter set simulations. Kernel density estimates show the probability density of detection ranges, with dashed vertical lines indicating the mean detection range for each condition. Left column: Empty World; Right column: Hanscom Air Force Base.

Figure 30 presents the detection success rate as a function of both range and fog condition. In both environments, detection success rates approach 100% at close ranges (400-800 ft) under no fog and minimal fog conditions, then decline progressively with increasing range. The high fog condition shows dramatically reduced success rates, particularly beyond 1000 ft, where detection probability falls below 20% in the Empty World scenario. The Hanscom Base environment demonstrates slightly better detection of persistence at longer ranges under medium and high fog conditions, consistent with the fog-induced clutter reduction effect described previously.

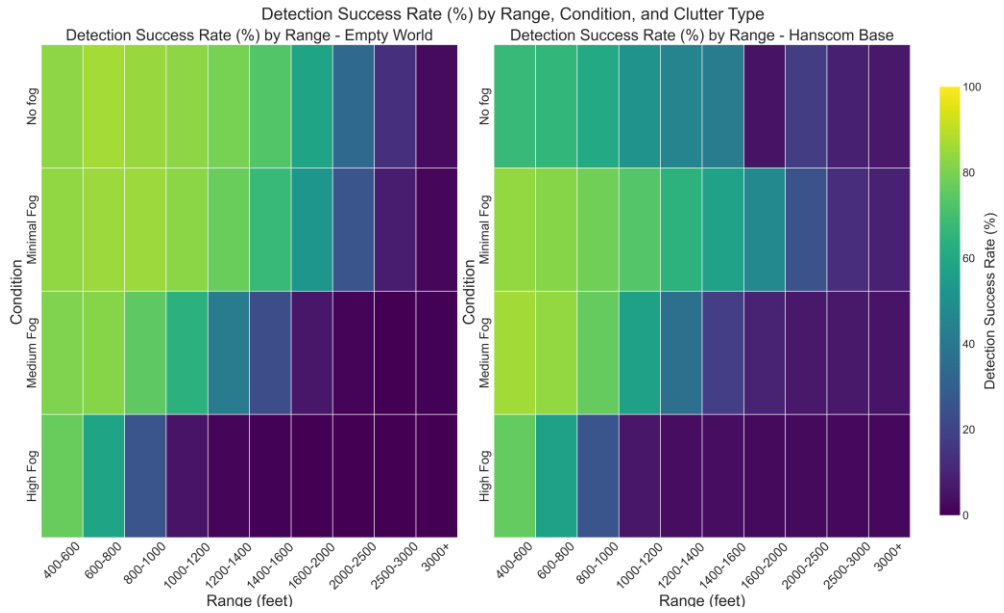


Figure 30. Detection success rate (%) as a function of range and fog conditions for Empty World (left) and Hanscom Base (right) environments. Color intensity indicates detection probability, with yellow representing high success rates and dark blue representing low success rates.

4.2 MIT Encounter Set Simulations with CasiaX random operating cameras

The previous simulations were repeated in the cluttered Hanscom Air Force Base environment while accounting for potential ownship camera failures during the DAA maneuver and for different fog levels. Accordingly, each time an MIT encounter set simulation is launched, the launch file randomly selects three of the five cameras available in the Casia X–inspired configuration.

Figure 31 shows the same MIT encounter set 6160 scenario with only three randomly selected cameras active. The camera numbering is as follows: (1) front camera, (2) right-front camera, (3) right-rear camera, (4) left-rear camera, and (5) left-front camera. In the plots, detections are unavailable because none of the sampled camera subsets included camera 3; consequently, the ownship could not observe the intruder during this portion of the DAA maneuver. The corresponding detection distributions for these simulations are also shown.

Figure 32 further illustrates that the detection distributions are less dense than in the nominal CasiaX configuration. Because camera 3 was selected less frequently, the detection density within its coverage sector is reduced, and intruder detection confidence decreases with increasing range. Finally, as shown in Figure 33, the average detection confidence for these degraded-camera cases is compared with the Hanscom Air Force Base simulations in which all cameras are available. Figure 33 demonstrates that reducing the number of active cameras decreases the number of intruder detections during the DAA maneuver, lowers the intruder detection confidence, and reduces the maximum detection range.

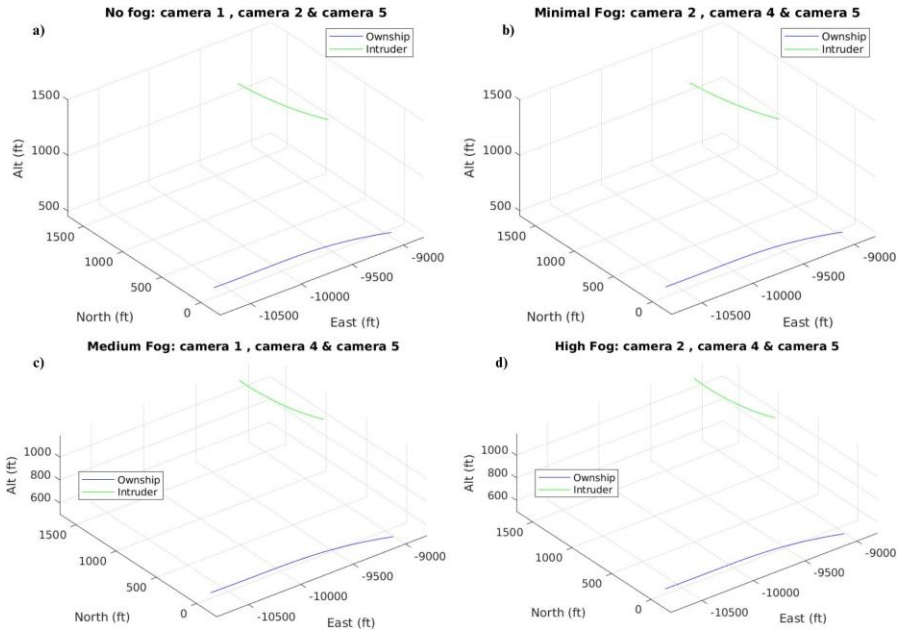


Figure 31. Detections in the Hanscom Air Force Base during the MIT encounter set 6160 at different levels of fog and with 3 random cameras: a) No fog, b) Minimum fog, c) Medium fog, and d) High fog.

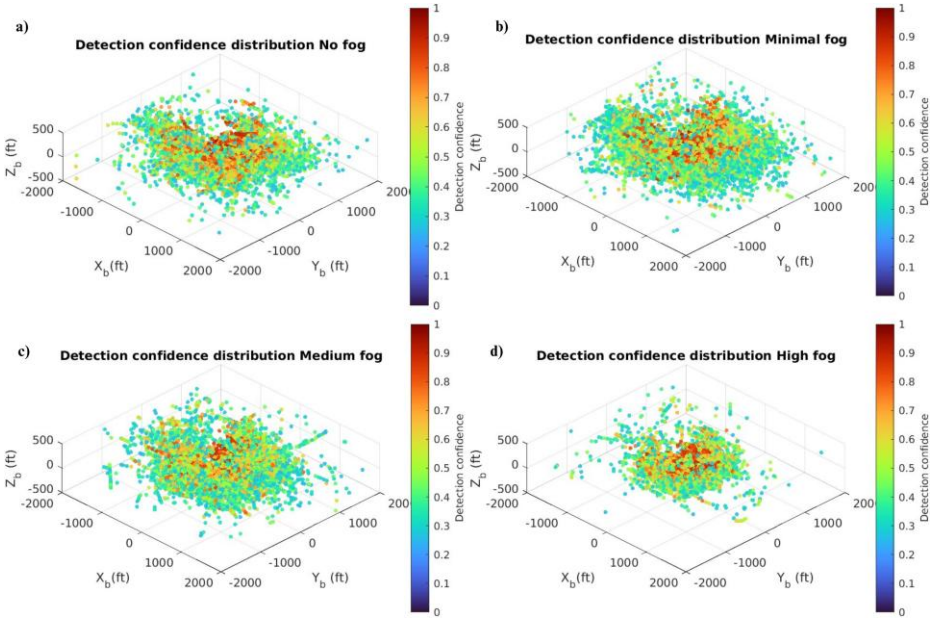


Figure 32. Confidence Detections distributions in the Hanscom Air Force Base with respect to the relative position of the intruder for the 1,000 MIT encounter sets at different levels of fog and 3 random cameras available: a) No fog, b) Minimum fog, c) Medium fog, and d) High fog.

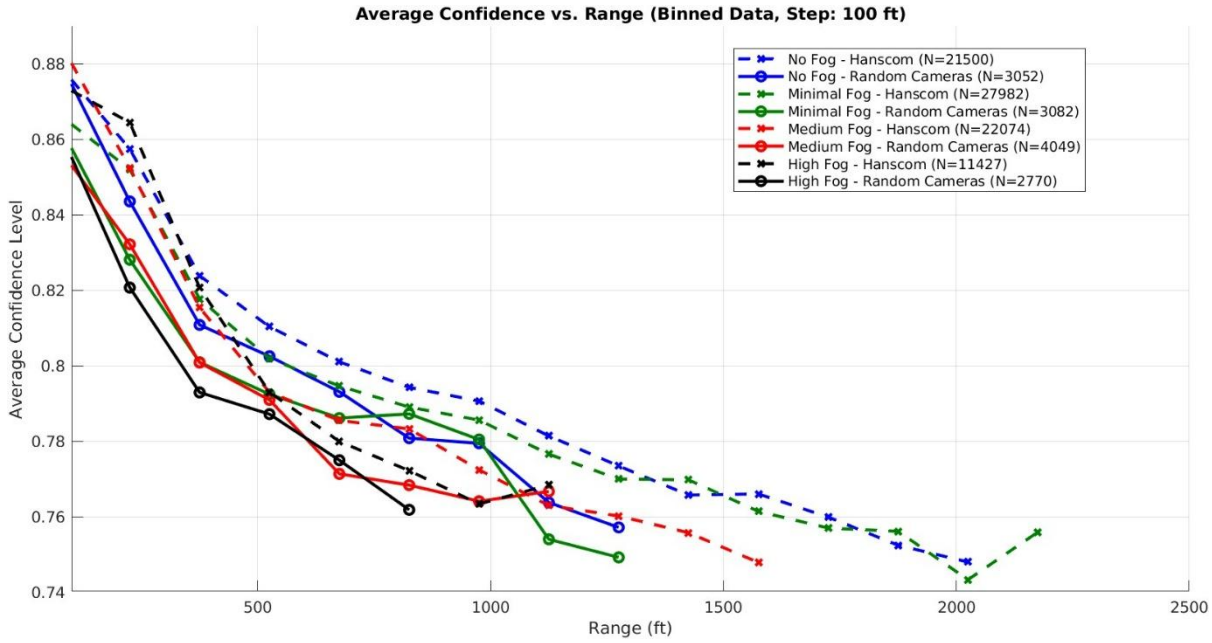


Figure 33. Average confidence detection levels vs range for different fog conditions in the Hansom Air Force Base and the Hanscom Air Force Base with random camera simulations.

As shown in Table 3, reducing the number of active cameras significantly degrades intruder detection performance during the DAA simulations, with the number of detections dropping to approximately 8% of the nominal (all-cameras) case. Therefore, reliable camera operation is essential for the visual detection system to ensure effective intruder detection throughout all phases of DAA missions.

Table 3. Percentage of detection on the Hanscom Air Force Base simulations with a random number of cameras available at different fog conditions.

Conditions	**Hanscom Base clutter**		**Detection percentage with random cameras**
	Total detections	**Total detections with Random cameras**	
No fog	27990	3052	10.90%
Minimal fog	38400	3082	8.03%
Medium fog	28978	4049	13.97%
High fog	14926	2770	18.56%

4.3 Detect and Avoid ADS-B Preliminary Simulations

The FGA algorithm, integrated into the ADS-B–based DAA system, was evaluated across multiple avoidance scenarios to characterize its performance under elevated risk conditions, including GNSS sensor failures (introduced via a dropout flag) and multi-intruder encounters. This section presents representative simulations conducted in the Gazebo environment for these risk cases of the ADS-B DAA system.

First, the simulation environment was used to reproduce four single-intruder encounters: frontal encounters, lateral encounters, sidewise encounters, and encounters with vertical separation at different altitude levels. The following figures illustrate these scenarios, showing the ownship avoidance trajectory, the intruder trajectory, and the intruder's spherical avoidance zone that the ownship must remain outside of when executing the avoidance algorithm.

Figures 34–37 demonstrate successful avoidance maneuvers in the presence of a single intruder under nominal conditions, using undisturbed ADS-B measurement inputs. However, Figure 35 shows a case in which the ownship briefly enters the intruder's avoidance zone, suggesting that larger safety margins (e.g., higher coefficients in the minimum safe avoidance radius) may be required to improve separation assurance. In addition, Figures 35 and 37 exhibit similar avoidance behavior due to the structure of the cost function, which favors lateral deviations to prevent the ownship from entering restricted areas. The same encounters were then repeated with a GNSS dropout applied during the time windows in which the avoidance maneuvers were previously executed.

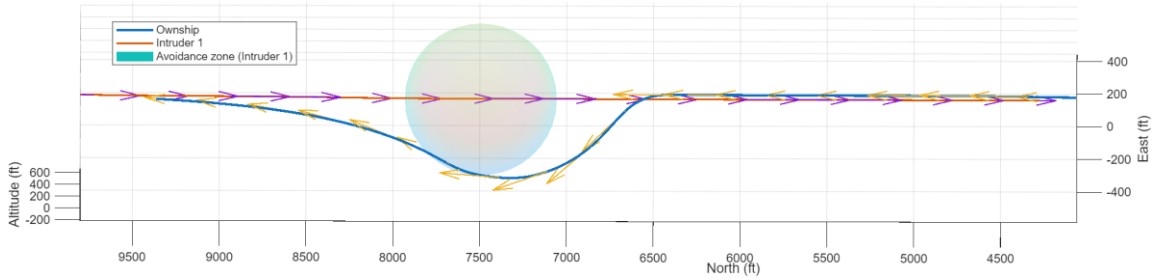


Figure 34. Frontal ADS-B DAA Avoidance in Gazebo.

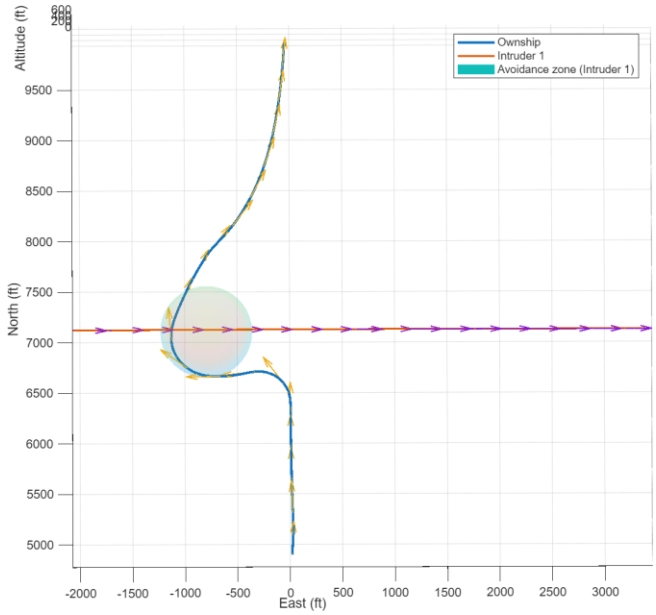


Figure 35. Lateral ADS-B DAA Avoidance.

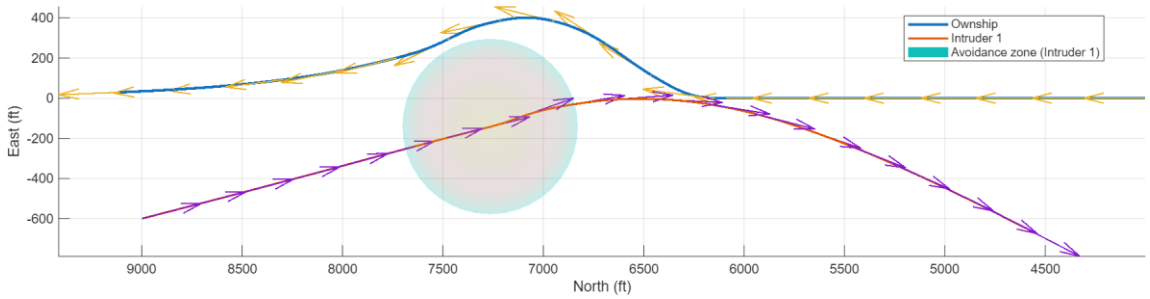


Figure 36. Sidewise DAA Avoidance in Gazebo.

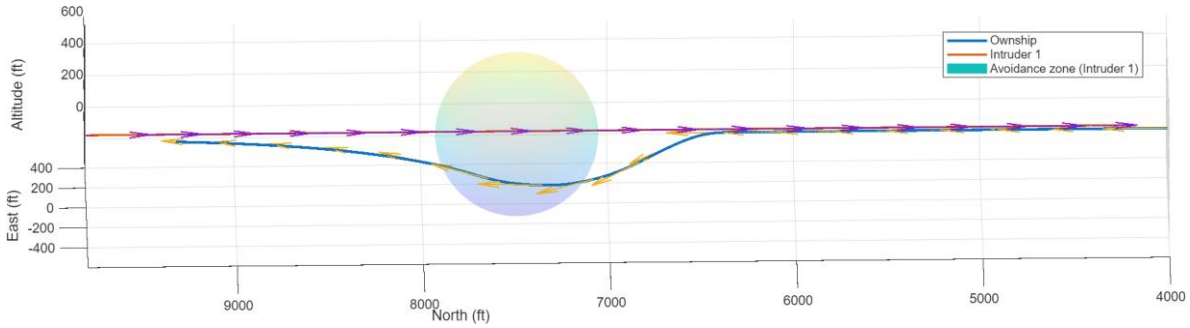


Figure 37. DAA Avoidance for UAVs at different levels of altitude.

Figures 38–41 illustrate the impact of a GNSS dropout on avoidance-trajectory generation by degrading the ADS-B position estimates used to localize intruders for DAA planning. This effect is evident in these figures, where the ADS-B–estimated intruder position during the frontal encounter is compared against the intruder's ground-truth position.

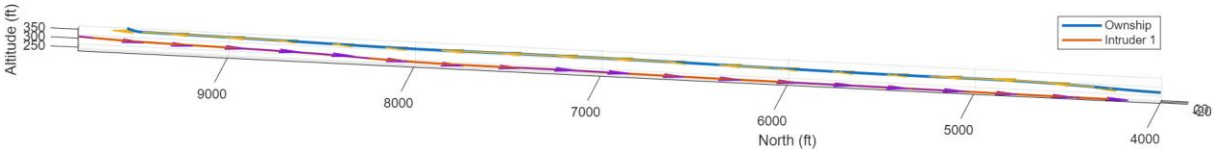


Figure 38. Frontal DAA Avoidance with GNSS Dropout in Gazebo.

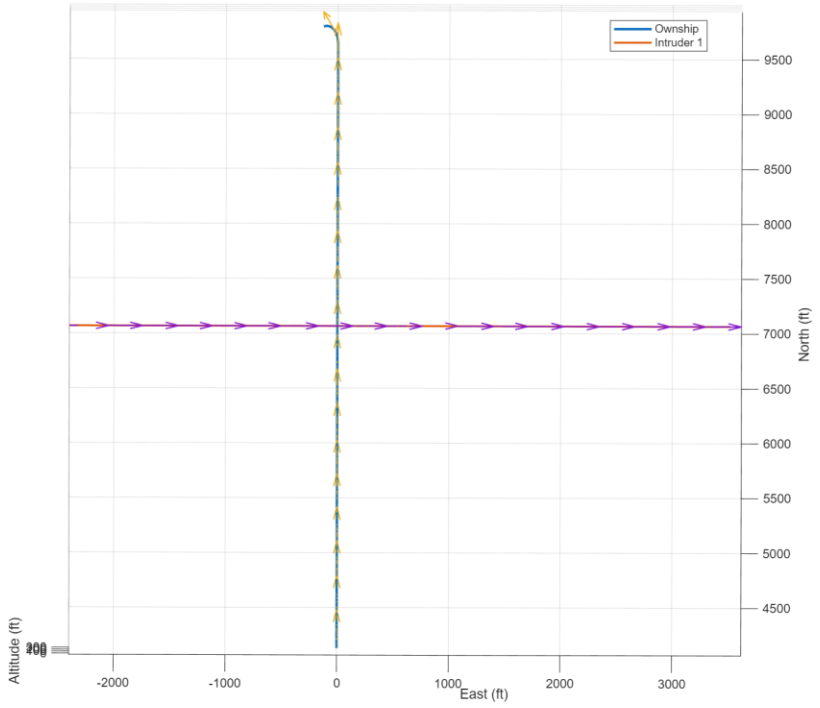


Figure 39. Lateral DAA Avoidance with GNSS Dropout in Gazebo.

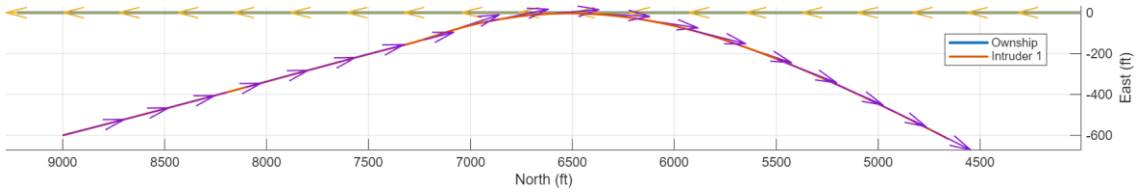


Figure 40. Sidewise DAA Avoidance with GNSS Dropout in Gazebo.

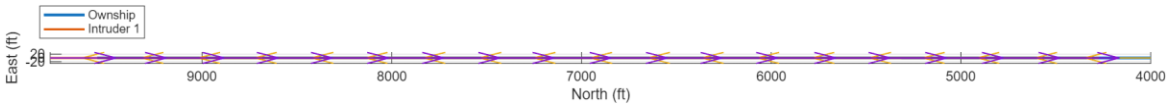


Figure 41. DAA Avoidance for UAVs at different levels of altitude with GNSS Dropout in Gazebo.

Figures 42–44 further show that GNSS dropouts can significantly degrade ADS-B estimates, preventing the DAA algorithm from reliably detecting intruders and generating appropriate avoidance trajectories. Moreover, the slow convergence of the ADS-B estimates after signal recovery can lead to overly conservative, unnecessarily large avoidance maneuvers, as observed near the end of the lateral avoidance scenario.

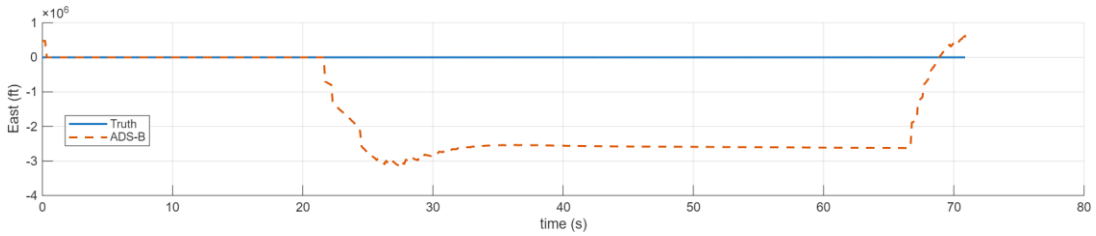


Figure 42. ADS-B East Estimation with GNSS dropout of the Intruder During Frontal DAA Avoidance.

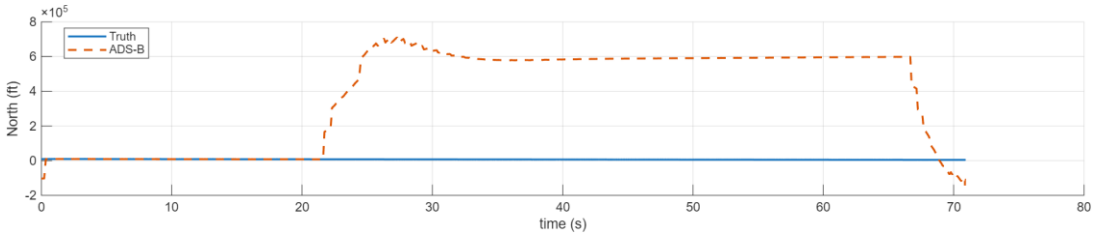


Figure 43. ADS-B North Estimation with GNSS dropout of the Intruder During Frontal DAA Avoidance.

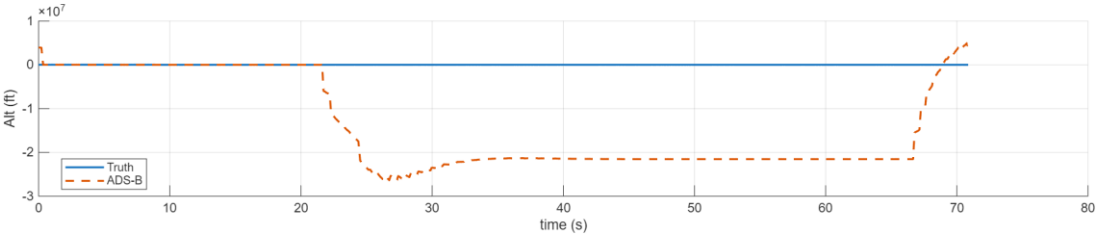


Figure 44. ADS-B Altitude Estimation with GNSS dropout of the Intruder During Frontal DAA Avoidance.

The simulation environment was also configured to easily increase the number of intruders and evaluate multi-intruder encounters during DAA simulations. Figures 38–41 present representative two-intruder scenarios in Gazebo, where both intruders broadcast ADS-B signals and the ownship executes the FGA algorithm. These results demonstrate successful avoidance maneuvers for these two-intruder cases. In Case 1 (Figure 45) and Case 3 (Figure 47), the intruders remain sufficiently separated that the algorithm generates distinct avoidance actions with respect to each intruder. In contrast, Case 4 (Figure 48) shows an overlap between the intruders' avoidance zones, producing an effectively larger combined avoidance region that the ownship must avoid. Finally, Case 2 (Figure 46) illustrates how the algorithm accounts for the relative locations of both intruders to determine an avoidance direction that reduces the likelihood of future conflicts.

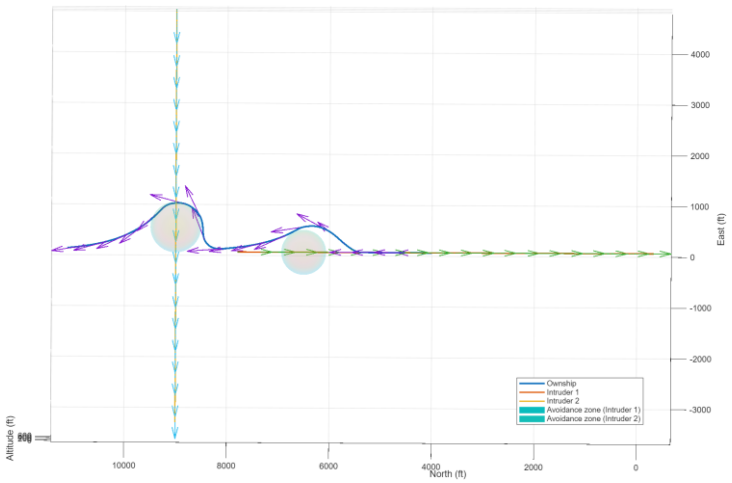


Figure 45. Case 1: Multi-Intruder DAA Scenario.

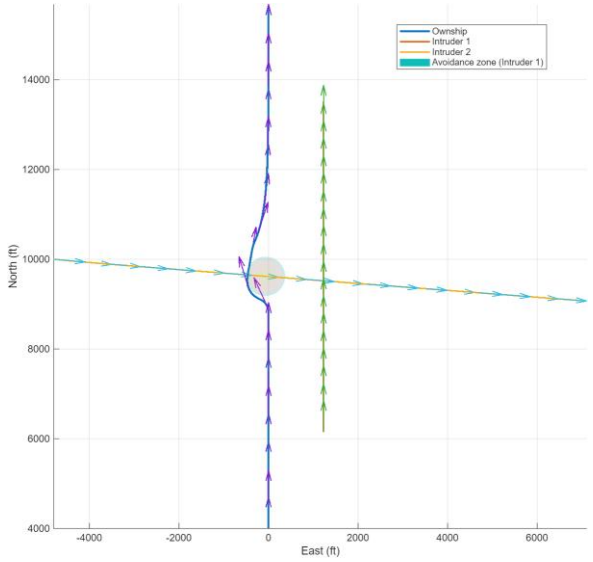


Figure 46. Case 2: Multi-Intruder DAA Scenario.

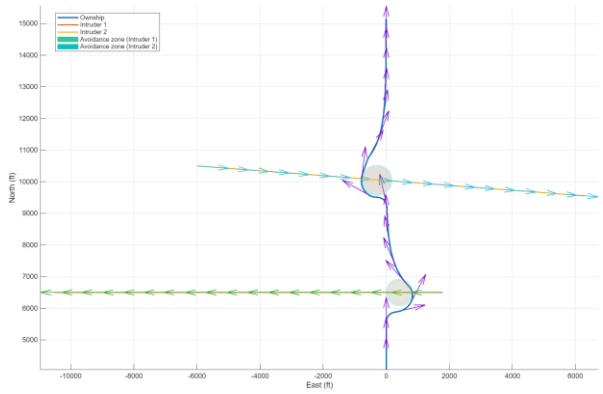


Figure 47. Case 3: Multi-Intruder DAA Scenario.

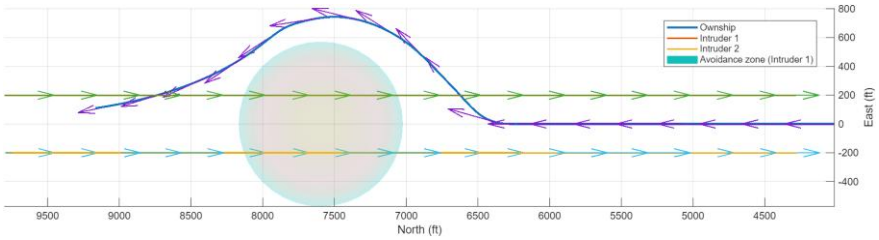


Figure 48. Case 4: Multi-Intruder DAA Scenario.

5 CONCLUSIONS

The work reported in this document supports Task 3 of the project and provides a simulation-driven safety risk management capability for unmanned-aircraft DAA systems when empirical performance data are sparse or unavailable. Several simulation tools were developed and integrated into a modular architecture to support repeatable, high-fidelity encounter studies with realistic sensing and estimation effects, including Casia X multi-camera vision configuration, GNSS multipath and dropouts, and ADS-B state estimation via EKF. Using MIT Lincoln Laboratory encounter trajectories and extensive Monte Carlo sampling, the probabilistic risk assessment framework derived in Task 2.1 was applied to estimate instantaneous detection probability as a function of separation distance under a range of operating conditions and system configurations. Collectively, these results demonstrate that combining high-fidelity simulation with machine-learning-based probabilistic modeling enables quantitative, scenario-specific characterization of DAA performance, supporting sensitivity analyses and informing safety-related assumptions needed for risk management documentation.

Across the evaluated cases, detection performance was strongly driven by both environmental complexity and system degradations. In the case of the CasiaX DAA system, visual clutter and fog reduced detection counts, confidence levels, and maximum detection range, with cluttered environments inducing detection dropouts even when confidence thresholds were increased to filter false positives. The analysis also showed that the Casia X architecture is sensitive to camera availability: random camera failures significantly reduced detection density in critical coverage sectors and drove detections down to a small fraction of the nominal all-cameras case. This highlights the importance of sensor reliability for maintaining detection continuity during DAA maneuvers. Similarly, GNSS dropouts degraded ADS-B state estimates and could prevent timely conflict detection or produce overly conservative avoidance trajectories due to slow estimator recovery. These findings show that credible safety risk management for DAA must explicitly account for coupled effects of sensing, estimation, and operational conditions, and that the developed simulation and PRA approach provides a practical pathway to quantify these effects and guide the selection of safety margins and mitigation strategies.

6 REFERENCE

[Beard, 2012] R. W. Beard and T. W. McLain, *Small Unmanned Aircraft: Theory and Practice*, Princeton University Press, Princeton, NJ, 2012, ch. 9.

[Lin, 2014] T.-Y. Lin *et al.*, “Microsoft COCO: Common Objects in Context,” in *European Conference on Computer Vision (ECCV)*, pp. 740–755, 2014. https://doi.org/10.1007/978-3-319-10602-1_48

[Lin, 2018] Z. Lin, L. Castano, H. Xu, "A fast obstacle collision avoidance algorithm for Fixed Wing UAS," *2018 International Conference on Unmanned Aircraft Systems (ICUAS).* https://doi.org/10.1109/icuas.2018.8453307

McKay, P. (2023). *TCCA Presentation to ASTM WK62669*. Transport Canada Civil Aviation, January 2023.

[Open Robotics, 2025] Open Robotics, “Tutorial: Using Gazebo Plugins with ROS,” *Gazebo Classic*, accessed Nov. 6, 2025. https://classic.gazebosim.org/tutorials?tut=ros_gzplugins

[OpenStreetMap, 2017] OpenStreetMap contributors, "Planet dump retrieved from https://planet.osm.org," *OpenStreetMap*, 2017. [Online]. Available: https://www.openstreetmap.org

[OGRE Team, 2025] OGRE Team (n. d.), "fog_override," *OGRE Manual v1.8*, sec. 3.1.2, Retrieved April 30, 2025. [Online]. Available: https://www.ogre3d.org/docs/manual18/manual_16.html

[Pant, 2022] K. A. Pant, Z. Yang, J. M. Goppert, and I. Hwang, "An Open-source Gazebo Plugin for GNSS Multipath Signal Emulation in Virtual Urban Canyons," 2022. https://arxiv.org/abs/2212.04018

[Park, 2004] S. Park, J. Deyst, and J. How, “A New Nonlinear Guidance Logic for Trajectory Tracking,” *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2004.

[PX4, 2025] PX4 Development Team, *Gazebo Vehicles — PX4 User Guide (v1.14)*, July 9, 2025. https://docs.px4.io/v1.14/en/sim_gazebo_gz/gazebo_vehicles.html

[Ray, 1999] J. K. Ray, M. E. Cannon, and P.C. Fenton, "Mitigation of static carrier-phase multipath effects using multiple closely spaced antennas," Navigation, 46(3):193-201, Sep 1999. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=dc99e55abbba1f0706f15ebc4a64b06fadd1cbef

[Redmon, 2016] J. Redmon, "You Only Look Once: Unified, Real-time Object Detection, " In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27-30 June, 2016.

https://ieeexplore.ieee.org/document/7780460

[Redmon, 2018] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018. Available: https://arxiv.org/abs/1804.02767

[uAvioni, 2024] uAvioni, *Casia X Datasheet*, Oct. 8, 2024, retrieved Jan. 13, 2025.

[vvoovv, 2025] vvoovv, *BLOSM documentation* [Computer software documentation], GitHub. Retrieved April 30, 2025. Available: https://github.com/vvoovv/blosm/wiki/Documentation