ENGG7291 – Project Proposal

UAV Guidance And Alerting Algorithm

Flynn Devoy

Student Number: 47442043

The University of Queensland / Revolution Aerospace

Date Submitted: March 16, 2026

Supervisors: Dr Chris James, Dr Terry Martin

List of Acronyms

ACAS	Airborne Collision Avoidance System
ACAS-X	Airborne Collision Avoidance System Experimental
ACAS-Xu	Airborne Collision Avoidance System Experimental Unmanned
ADS-B	Automatic Dependent Surveillance–Broadcast
BVLOS	Beyond Visual Line of Sight
CA	Collision Avoidance
DAA	Detect and Avoid
GCS	Ground Control Station
HITL	Human-in-the-Loop
ICAO	International Civil Aviation Organization
MAC	Mid-Air Collision
MDP	Markov Decision Process
MOPS	Minimum Operational Performance Standards
NMAC	Near Mid-Air Collision
NAS	National Airspace System
RA	Resolution Advisory
RWC	Remain Well Clear
STM	Surveillance and Tracking Module
TRM	Threat Resolution Module
UAS	Unmanned Aircraft System
UTM	Unmanned Aircraft System Traffic Management
WC	Well Clear

Acknowledgement of AI Use

AI tools were used in a limited capacity to assist with structuring information, generating illustrative diagrams, and providing suggestions for risk management approaches. All outputs were reviewed, modified, and verified by the author prior to inclusion in this report.

Table 1: Acknowledgement of AI use in this project proposal

AI Tool	**Purpose of Use**	**Indicative Prompt**	**Section**
GPT-5	Summarising project brief to clarify project scope and deliverables	"Summarise the project brief into clear aims, scope and required deliverables for a project proposal."	Introduction
GPT-5	Generation of Mermaid script to illustrate the architecture of a generic Detect-and-Avoid system	"Create a Mermaid flow chart with the main components (Kinematic Propgation, STM, State Estimation and Preprocessing, TRM and Command) of a generic Detect-and-Avoid (DAA) system and how information flows between modules."	Technical Background
GPT-5	Generation of Mermaid script representing the horizontal threat resolution logic used in collision avoidance systems	"Create a Mermaid flow chart describing the horizontal threat resolution process for an ACAS-Xu style collision avoidance system."	Literature Review
GPT-5	Suggestions for preventing, and mitigating project risks	"Suggest the most effective mitigation and prevention strategies for the project risks to ensure the overall project risk is minimised"	Risk Assessment
GPT-5	Referencing in Latex	"Please generate a reference for the following citation in latex formatting suitable for a .bib file"	References

Contents

1 Introduction **1**
1.1 Project Motivation 1
1.2 Project Aim 1
1.3 Project Scope 2
1.3.1 In Scope 2
1.3.2 Out of Scope 2
1.4 Required Deliverables 3
1.5 Supervisory Team Structure 3

2 Technical Background **4**
2.1 Conventional ACAS / TCAS Systems 4
2.1.1 Design Philosophy 4
2.1.2 Resolution Advisory Logic 4
2.1.3 Operational Limitations 4
2.1.4 Next-Generation Collision Avoidance: ACAS X 4
2.2 Detect and Avoid (DAA) Systems 5
2.2.1 Definition and Purpose of DAA 5
2.2.2 Remain Well Clear vs Collision Avoidance 5
2.2.3 Generic Detect-and-Avoid Functional Architecture 5
2.2.4 Safety Evaluation of DAA Systems 6

3 Literature Review **7**
3.1 DAIDALUS Within Detect-and-Avoid Architectures 7
3.2 ACAS-Xu Probabilistic Modelling 7
3.3 Markov Decision Process Formulation 8
3.4 Example Implementation of ACAS-Xu Horizontal Threat Resolution Logic 9
3.5 Existing Comparative Evaluation of ACAS-Xu and DAIDALUS 10
3.6 Literature Synthesis and Identified Research Gap 10

4 Project Planning and Methodology **11**
4.1 Stage 1: Project Proposal and Literature Review (9 Feb 2026 – 16 Mar 2026) 11
4.1.1 Literature Review and Background Research 11
4.1.2 System Scope and Architecture Definition 11
4.1.3 Preliminary Implementation of Horizontal Avoidance Logic 11
4.2 Section 2: Progress Review and Prototype Development (16 Mar 2026 - 5 May 2026) 12
4.2.1 Development of Vertical Avoidance Logic 12
4.2.2 Integration of Prototype Modules 12
4.2.3 Initial Benchmarking and Progress Review Preparation 12
4.3 Section 3: Results Analysis, Final Benchmarking and Project Completion (5 May 2026 - 2 Jul 2026) 12
4.3.1 Analysis of Initial Results and Prototype Refinement 13
4.3.2 Final Benchmarking and System Performance Evaluation 13
4.3.3 Oral Presentation Preparation and Final Report Completion 13
4.4 Project Timeline Overview 13

5 Project Risks and Opportunities **13**
5.1 Workplace Risks 14
5.2 Scheduling Risks 14
5.3 Technical Risks 15
5.4 Opportunities 15

A Project Schedule (Gantt Chart) **17**

B Project Risk Assessment **18**

List of Figures

1 High-level functional architecture of a generic DAA system showing closed-loop interaction between kinematic propagation, surveillance, state estimation, threat resolution, and guidance execution. 5
2 Horizontal Threat Resolution Module update process derived from ACAS-Xu algorithm design documentation. 9
3 Project Gantt Chart 17

List of Tables

1 Acknowledgement of AI use in this project proposal ii
2 Workplace Risks 14
3 Scheduling Risks 14
4 Technical Risks 15
6 Project Risk Matrix 18
7 Risk Tolerance Levels and Required Mitigation Actions 18

1 Introduction

1.1 Project Motivation

Unmanned Aeriel Vehicles (UAVs) offer significant operational and economic advantages across surveillance, logistics, inspection and defence applications. However, the safe integration of UAV's into civil airspace remains constrained by regulatory requirements mandating an Equivalent Level of Safety (ELOS) relative to crewed aviation operations.

Traditional airborne collision avoidance systems such as ACAS II (TCAS II) were designed for cooperative, turbine-powered, fixed-wing aircraft operating under instrument flight rules in structured airspace, [1]. The system interrogates Mode A/C and Mode S transponders and issues vertical Resolution Advisories (RAs) intended for pilot execution, [1]. The ICAO ACAS Manual explicitly states that ACAS was not originally designed for unmanned aircraft or tactical platforms and requires additional considerations when applied outside its intended operation context, [1].

Small and medium UAV operations differ significantly from those assumptions. UAVs frequently operate at lower altitudes, may have constrained vertical performance and lack an onboard pilot capable of visual acquisition and discretionary manoeuvring. Additionally, UAVs may encounter mixed cooperative and non-cooperative traffic in less structured environments. As noted in the ACAS sXu architecture overview, safe integration of small UAS into the National Airspace System requires decentralised and scalable Detect-and-Avoid (DAA) capability that is robust to diverse surveillance sources and uncertain traffic behaviour, [2].

Modern collision avoidance logic within the ACAS-X family addresses limitations of legacy rule-based systems by reformulating the avoidance problem as a Markov Decision Process (MDP), enabling advisories to be selected by minimising expected collision risk under uncertainty rather than through fixed threshold logic [3]. The ACAS-Xu and sXu variants adapt this optimisation-based framework to unmanned aircraft by discretising horizontal and vertical state spaces and solving probabilistic dynamic programming problems offline [2].

RevAero currently utilises a DAA capability based on NASA's DAIDALUS framework, which primarily provides well-clear boundary monitoring and guidance bands. While well-clear monitoring is an essential component of DAA architectures, it differs fundamentally from optimisation-based collision avoidance. ACAS-Xu provides advisory selection derived from explicit cost-function optimisation, allowing tunability and probabilistic robustness [3], [4].

Furthermore, Safety Risk Management (SRM) analyses conducted for DAA Phase 2 emphasise the need for structured modelling and simulation to quantify trade-offs between alert rate, Near Mid-Air Collision (NMAC) probability, and operational acceptability [4], [5]. These trade-offs cannot be fully assessed without comparative evaluation under identical encounter sets.

Accordingly, RevAero is investigating whether ACAS-Xu can serve as a contemporary replacement or enhancement to its existing DAA guidance and alerting capability. This project directly supports that investigation through structured implementation, integration, and comparative safety performance evaluation grounded in established ACAS-X methodologies and SRM principles [2], [4], [5].

1.2 Project Aim

The aim of this project is to design, implement, and evaluate a customised version of the Airborne Collision Avoidance System for Unmanned Aircraft (ACAS-Xu), optimised for integration within RevAero's DAA framework.

Specifically, the project seeks to:

- Develop a working ACAS-Xu model adapted for practical autonomous UAV integration;
- Integrate the customised ACAS-Xu logic within an existing DAA simulation and evaluation environment;
- Benchmark its performance against RevAero's DAIDALUS-based DAA implementation;
- Conduct comparative analysis against a commercial ACAS-Xu implementation (SageTech variant);
- Identify strengths, limitations, and operational trade-offs across implementations;
- Generate structured technical insights into the applicability of optimisation-based collision avoidance approaches for autonomous aircraft.

The outcomes of this work are intended to support the maturation of UAV collision-avoidance technologies within operationally relevant environments and inform future architectural decisions at RevAero.

1.3 Project Scope

1.3.1 In Scope

The project encompasses the development, adaptation, integration, and comparative evaluation of ACAS-Xu implementations. The primary focus is on algorithmic development, simulation-based validation, and quantitative performance assessment.

Key activities include:

- Using publicly available ACAS-Xu materials and reference implementations as a technical foundation;
- Tailoring the ACAS-Xu logic to enable functional integration with RevAero's existing DAA test framework;
- Configuring and validating state discretisation, advisory logic, and integration interfaces;
- Benchmarking performance against RevAero's DAIDALUS implementation under representative encounter scenarios;
- Evaluating a commercial SageTech ACAS-Xu variant to compare:
  - Academic/open-source implementation
  - Commercial implementation
  - In-house DAA implementation

Performance will be assessed using quantitative metrics including:

- Detection and Alerting Range,
- Alert timing relative to closest point of approach,
- False alert and nuisance alert rate,
- Near Mid-Air Collision (NMAC) proxy metrics
- Advisory stability and reversals
- Computational efficiency and runtime performance

Scenario-based Monte Carlo simulations will be conducted to evaluate system behavior under representative traffic encounter sets.

1.3.2 Out of Scope

The project does not include:

- Hardware certification or airworthiness approval;
- Real-world flight testing;
- Development of new surveillance sensor hardware;
- Regulatory approval processes.

The focus remains on software-level integration, algorithmic evaluation, and comparative safety assessment within a controlled simulation environment.

1.4 Required Deliverables

The project will produce the following deliverables:

1. **Working Adapted ACAS-Xu Implementation**
   A customised ACAS-Xu model capable of integration within RevAero's DAA simulation framework.

2. **Comparative Evaluation Reports**
   Structured benchmarking of ACAS-Xu against:

   - NASA's DAIDALUS implementation,
   - The SageTech commercial ACAS-Xu variant,

   across agreed quantitative performance metrics.

3. **Structured Encounter Dataset and Results Archive**
   A reproducible dataset of encounter scenarios and associated simulation outputs to support verification and future research.

4. **Technical Documentation**
   Comprehensive documentation of:

   - Integration architecture,
   - Configuration parameters,
   - Modifications performed,
   - Observed limitations,
   - Recommendations for optimisation and future deployment.

5. **Final Report and Presentation**
   A formal written report and presentation summarising:

   - System architecture,
   - Methodology,
   - Comparative findings,
   - Operational implications,
   - Recommendations for RevAero's future DAA development roadmap.

1.5 Supervisory Team Structure

The primary project supervisor is Dr Terry Martin, who provides technical oversight of the project and is responsible for guiding the overall direction of the work. In this role, Dr Martin supports the development of the prototype system, reviews project progress, and provides feedback on the technical implementation of the DAA model.

The university supervisor for the project is Dr Chris James. Dr James provides academic guidance throughout the placement and offers advice regarding the assessment requirements associated with the project. This includes feedback on report structure, presentation preparation, and any academic or workplace issues that may arise during the course of the project.

In addition to the formal supervisory roles, several members of the DAA team at Revolution Aerospace are available to provide technical support. These team members will be particularly involved during the later stages of the project when the developed prototype system is integrated with the existing RevAero framework.

Furthermore, communication has been established with researchers at MIT Lincoln Laboratory, who are widely recognised as leaders in the development of the ACAS-X family of collision avoidance systems. Their expertise will assist with obtaining or generating realistic aircraft encounter scenarios required for evaluating the ACAS-Xu prototype.

2 Technical Background

2.1 Conventional ACAS / TCAS Systems

2.1.1 Design Philosophy

The Airborne Collision Avoidance System (ACAS) was developed as an independent airborne safety net intended to reduce the risk of mid-air collisions in civil aviation. Its primary objective is to provide flight crews with timely advisories to avoid potential collisions, operating independently of ground-based air traffic control (ATC) systems [1]. ACAS therefore functions as a last-resort collision avoidance layer that supplements, but does not replace, procedural and ATC-based separation services.

ACAS operates through active interrogation of nearby Mode A/C and Mode S transponders. By processing transponder replies, the system estimates range, altitude, and closure rate, and determines whether projected encounter geometry satisfies predefined threat criteria [1]. The system architecture assumes cooperative, transponder-equipped aircraft operating within structured airspace environments.

Two principal variants have historically been implemented:

- ACAS I (equivalent to TCAS I), which provides Traffic Advisories (TAs) only.
- ACAS II (equivalent to TCAS II), which provides both TAs and Resolution Advisories (RAs).

ACAS II is mandated for turbine-powered aircraft above specified weight or passenger thresholds under ICAO Annex 6 provisions [1]. The system was designed for crewed, fixed-wing aircraft operating under civil flight procedures and assumes pilot interpretation and compliance with issued advisories. It was not originally developed for autonomous unmanned aircraft operations.

2.1.2 Resolution Advisory Logic

ACAS I provides Traffic Advisories intended to enhance pilot situational awareness and support visual acquisition of nearby traffic. It does not issue manoeuvre commands and therefore relies on pilot judgement for conflict resolution [1].

ACAS II extends this capability by issuing Resolution Advisories when predicted encounter conditions satisfy time-to-go and altitude separation thresholds. Threat detection logic evaluates projected relative geometry using range, relative altitude, and closure rate. When both horizontal and vertical criteria are met within sensitivity-level-dependent thresholds, an RA is generated [1].

Resolution Advisories are vertical manoeuvre commands such as climb, descend, increase climb, or maintain vertical speed. Advisories are typically issued approximately 15–35 seconds prior to predicted closest point of approach (CPA), depending on altitude and sensitivity level [1].

When both aircraft are ACAS II equipped, a coordination procedure via the Mode S data link ensures complementary advisories, preventing conflicting manoeuvres [1]. Horizontal manoeuvre advisories were explored in earlier system concepts but were not operationally certified, resulting in a vertical-only collision avoidance framework.

2.1.3 Operational Limitations

Despite its demonstrated safety benefits, conventional ACAS implementations exhibit several structural limitations.

First, ACAS relies on cooperative surveillance through transponder interrogation, preventing detection of non-cooperative aircraft. Second, ACAS II provides only vertical Resolution Advisories, as horizontal manoeuvre guidance was not operationally implemented. Third, advisory generation is based on fixed threshold parameters rather than optimisation of expected collision risk.

These limitations motivated the development of the next-generation Airborne Collision Avoidance System X (ACAS X), which replaces threshold-based logic with a probabilistic decision framework designed to explicitly minimise collision risk under uncertainty.

2.1.4 Next-Generation Collision Avoidance: ACAS X

ACAS X represents a methodological shift from threshold-based collision avoidance logic to probabilistic optimisation. The system models aircraft encounters as a Markov Decision Process (MDP) and computes optimal advisory policies offline using dynamic programming to minimise expected cumulative cost associated with collision risk and alerting burden [3]. The resulting policies are stored as lookup tables that can be queried in real time during flight operations.

2.2 Detect and Avoid (DAA) Systems

2.2.1 Definition and Purpose of DAA

Detect-and-Avoid (DAA) is the functional capability that enables an aircraft to detect nearby traffic, assess collision risk, and generate guidance to remain well clear of other aircraft and avoid mid-air collisions. For unmanned aircraft systems (UAS), DAA serves as the functional equivalent of the see-and-avoid requirement imposed on crewed aviation.

The objective of DAA systems is not solely collision avoidance, but layered safety assurance across a spectrum of encounter severities. According to the Phase 2 Safety Risk Management (SRM) analysis conducted by MIT Lincoln Laboratory, DAA must support both strategic and tactical separation while maintaining acceptable operational suitability within the National Airspace System (NAS) [4].

RTCA guidance further formalises DAA performance expectations within the SRM modelling framework, identifying key safety metrics including:

- Probability of Loss of Well Clear (LoWC),
- Probability of Near Mid-Air Collision (NMAC),
- Risk Ratio,
- Severity of Loss of Well Clear (SLoWC).

These metrics reflect that DAA is fundamentally a risk management system rather than a simple alerting mechanism [5].

2.2.2 Remain Well Clear vs Collision Avoidance

DAA architectures typically distinguish between two functional layers:

1. Remain Well Clear (RWC)
2. Collision Avoidance (CA)

The RWC function operates with larger separation volumes and longer look-ahead times. Its objective is to prevent aircraft from entering hazardous proximity regions in the first instance. Well-clear thresholds are generally defined using horizontal and vertical separation criteria combined with time-to-go parameters.

CA, by contrast, is a last-resort safety net. It operates at shorter time horizons and may command more aggressive manoeuvres intended specifically to prevent NMAC events once well-clear separation has already degraded.

2.2.3 Generic Detect-and-Avoid Functional Architecture

The high-level functional architecture of a generic DAA system is illustrated in Figure 1. The diagram emphasises the closed-loop nature of the system: manoeuvre commands issued to the guidance system modify ownship kinematics, which subsequently influence future surveillance observations and conflict assessments. This feedback structure highlights that DAA performance must be evaluated as a dynamic, stochastic control problem rather than as an isolated advisory generator.

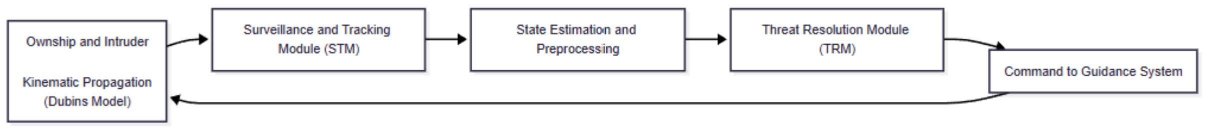

Ownship and Intruder
Kinematic Propagation (Dubins Model)
Surveillance and Tracking Module (STM)
State Estimation and Preprocessing
Threat Resolution Module (TRM)
Command to Guidance System

Figure 1: High-level functional architecture of a generic DAA system showing closed-loop interaction between kinematic propagation, surveillance, state estimation, threat resolution, and guidance execution.

As shown in Figure 1, the DAA architecture is implemented as a closed-loop system comprising five principal functional elements:

1. **Ownship and Intruder Kinematic Propagation** models the physical evolution of aircraft states. In simulation environments, this is commonly represented using simplified kinematic models such as Dubins Model.

2. **Surveillance and Tracking Module (STM)** integrates cooperative sources (ADS-B, Mode S) and non-cooperative sensors (radar, EO/IR), performing track maintenance and generating relative state measurements of surrounding traffic.

3. **State Estimation and Preprocessing** transforms surveillance data into structured state representations suitable for decision logic, incorporating filtering, uncertainty estimation, and coordinate transformations where required.

4. **Threat Resolution Module (TRM)** evaluates projected encounter geometry and determines avoidance guidance.

5. **Command to Guidance System** converts the selected advisory into control inputs, modifying ownship trajectory and closing the loop illustrated in Figure 1.

This structure highlights that DAA performance arises from the interaction between sensing, estimation, decision logic, and vehicle response, rather than from resolution logic alone.

2.2.4 Safety Evaluation of DAA Systems

DAA systems are evaluated using probabilistic safety modelling rather than deterministic test cases. Because Near Mid-Air Collision (NMAC) events are extremely rare, Monte Carlo simulation is the dominant evaluation methodology [4].

Let N denote the number of simulated encounters. The estimated NMAC probability is

$$\hat{P}_{\text{NMAC}} = \frac{N_{\text{NMAC}}}{N}$$

where N_{NMAC} is the number of encounters resulting in NMAC under the evaluated policy.

Performance is typically expressed in terms of *Risk Ratio*, defined as

$$\text{Risk Ratio} = \frac{P_{\text{NMAC, DAA}}}{P_{\text{NMAC, No DAA}}}$$

which measures the reduction in collision risk achieved by the DAA system relative to a no-avoidance baseline.

The MIT Lincoln Laboratory Phase 2 Safety Risk Management (SRM) modelling framework implements this evaluation within large-scale Monte Carlo environments such as DEGAS [4]. These simulations explicitly model the full DAA architecture, including surveillance uncertainty, state estimation errors, latency budgets, aircraft performance limits, and pilot or autonomous response dynamics. By propagating these uncertainties across millions of synthetic encounters, the framework captures the coupled effects of each architectural layer on overall system-level risk.

Sensitivity analyses are conducted by varying:

- Surveillance accuracy,
- Communication latency,
- Response delays,
- Encounter model distributions.

RTCA guidance emphasises that safety conclusions must be supported by statistically significant encounter sets and appropriate confidence interval reporting [5].

This evaluation methodology highlights that advisory generation is only one component of a broader probabilistic safety chain. Overall DAA performance depends not solely on resolution logic, but also on surveillance fidelity, estimator stability, timing constraints, and vehicle response characteristics.

3 Literature Review

3.1 DAIDALUS Within Detect-and-Avoid Architectures

NASA's DAIDALUS (Detect and AvoID Alerting Logic for Unmanned Systems) is a reference implementation of detect-and-avoid requirements for UAS operations [6]. It was developed to provide a practical and mathematically well-defined framework for airborne conflict detection and guidance, particularly in the context of integrating unmanned aircraft into shared civil airspace.

Within a broader DAA architecture, DAIDALUS operates in the decision and guidance layer. It takes processed traffic state information from surveillance and tracking systems and uses this information to determine well-clear status, generate alert levels, and produce avoidance guidance. In this sense, DAIDALUS does not perform sensing or state estimation itself; rather, it acts on the outputs of those upstream modules.

DAIDALUS is centred on the mathematical definition of the well-clear concept. It evaluates projected encounter geometry against prescribed horizontal, vertical, and temporal separation thresholds and determines whether the ownship is predicted to remain well clear of surrounding traffic. When a potential violation is detected, the system produces corrective guidance intended to maintain or recover safe separation. Its outputs include alerting logic and manoeuvre guidance bands over track, ground speed, vertical speed, and altitude [6].

A key strength of DAIDALUS is that it provides structured, interpretable guidance consistent with the remain-well-clear philosophy of DAA. Because its logic is based on deterministic kinematic projection and geometric threshold evaluation, it is well suited to applications in which transparency, predictability, and compliance with RTCA well-clear concepts are important. Accordingly, DAIDALUS is best understood as a remain-well-clear and conflict management framework rather than a probabilistic collision avoidance optimiser.

3.2 ACAS-Xu Probabilistic Modelling

ACAS-Xu applies the broader ACAS X collision avoidance framework to unmanned aircraft operations. Whereas systems such as DAIDALUS focus primarily on maintaining well-clear separation through deterministic geometric conflict detection, ACAS-Xu addresses shorter-horizon collision avoidance using a probabilistic decision framework.

The fundamental shift introduced by ACAS-Xu is that advisory selection is not based solely on whether projected trajectories cross predefined separation thresholds. Instead, the system evaluates candidate manoeuvres according to their expected safety outcomes under uncertainty. This allows the decision logic to account explicitly for uncertain encounter evolution, sensor estimation errors, intruder behaviour, and response delays.

This approach is grounded in research on probabilistic encounter modelling and mid-air collision risk estimation [3]. Within the ACAS-X architecture, potential actions are evaluated according to their ability to reduce the likelihood of hazardous outcomes such as loss of well-clear or near mid-air collision, while also discouraging unnecessary or operationally disruptive manoeuvres.

In the DAA architecture, ACAS-Xu operates within the Threat Resolution Module (TRM). It does not perform surveillance or state estimation itself; rather, it uses the encounter state estimates provided by upstream tracking systems to determine appropriate collision avoidance guidance. Consequently, ACAS-Xu should be viewed as an optimisation-based resolution component within a broader DAA system rather than a complete detect-and-avoid solution.

For unmanned aircraft applications, ACAS-Xu is particularly valuable because the framework can incorporate vehicle-specific dynamics, response delays, and operational constraints within its decision logic. More broadly, the progression from deterministic well-clear algorithms toward ACAS-Xu reflects a shift in DAA system design from purely geometric conflict detection toward probabilistic risk-based decision making.

The mathematical framework used to implement this probabilistic advisory selection process is commonly formulated as a Markov Decision Process (MDP), which is described in the following section.

3.3 Markov Decision Process Formulation

To implement the probabilistic advisory selection process described in the previous section, the ACAS-X framework formulates collision avoidance as a Markov Decision Process (MDP). This framework provides a principled way to evaluate candidate avoidance manoeuvres under uncertainty and determine actions that minimise the expected long-term risk of hazardous encounter outcomes [3], [7].

An MDP is defined as the tuple

$$\mathcal{M} = (S, A, P, C, \gamma)$$

where S denotes the state space, A the action space, $P(s'|s, a)$ the probability of transitioning from state s to state s' after action a, $C(s, a)$ the immediate cost associated with that action, and $\gamma \in (0, 1]$ a discount factor weighting future costs relative to present costs [7].

In airborne collision avoidance applications the state represents the relative geometry between the ownship and an intruder aircraft. A simplified state representation may be written as

$$s = (r, \dot{r}, h, \dot{h}, \tau)$$

where r is horizontal separation, $\dot{r}$ is horizontal closure rate, h is relative altitude, $\dot{h}$ is relative vertical rate, and τ represents the projected time-to-loss of separation.

The available manoeuvres are represented by the action space

$$A = \{\text{COC}, \text{Climb}, \text{Descend}, \text{Turn Left}, \text{Turn Right}\}$$

where COC denotes clear-of-conflict and the remaining actions correspond to vertical or horizontal avoidance manoeuvres subject to aircraft performance constraints.

The evolution of the encounter is governed by the transition model

$$P(s'|s, a)$$

which captures uncertainty in aircraft dynamics, intruder manoeuvre behaviour, sensor estimation error, and response delays [3].

Within this framework undesirable outcomes are penalised through a cost function

$$C(s, a) = C_{\text{NMAC}}\mathbb{I}_{\text{NMAC}} + C_{\text{LoWC}}\mathbb{I}_{\text{LoWC}} + C_{\text{alert}}\mathbb{I}_{a\neq\text{COC}}$$

where the indicator functions represent Near Mid-Air Collision (NMAC), loss of well-clear separation, and the issuance of an advisory respectively. The cost values are chosen such that

$$C_{\text{NMAC}} \gg C_{\text{LoWC}} \gg C_{\text{alert}}$$

ensuring collision avoidance risk dominates other operational considerations [7].

Rather than minimising only the immediate cost of an action, the system seeks to minimise the expected cumulative cost over the duration of the encounter

$$J^{\pi}(s_0) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t C(s_t, a_t)\right]$$

where the expectation is taken over the stochastic encounter dynamics.

Solving this optimisation problem directly in real time would be computationally intractable due to the large number of possible encounter states. Instead, ACAS-X computes the optimal policy offline using dynamic programming techniques [7]. By discretising the encounter state space and applying value iteration based on the Bellman optimality equation, the expected cumulative cost for each state-action pair can be computed.

The result of this optimisation process is a policy

$$\pi^* = \arg\min_{\pi} J^{\pi}(s_0)$$

which maps encounter states to the action that minimises expected future risk. These optimal decisions are stored in lookup tables queried in real time by the onboard collision avoidance system. Consequently, the computationally expensive optimisation is performed offline, allowing ACAS-Xu to issue advisories rapidly during flight while still benefiting from the probabilistic optimisation performed during policy generation.

3.4 Example Implementation of ACAS-Xu Horizontal Threat Resolution Logic

The Airborne Collision Avoidance System Experimental Unmanned, (ACAS-Xu), algorithm is implemented following the structure described in the Algorithm Design Description [8]. The documentation outlines the core decision-making logic used by the system and provides a detailed description of the internal processes used to generate collision avoidance advisories.

The algorithm includes both horizontal and vertical threat resolution logic; however, the present analysis focuses on the horizontal threat resolution component. This logic forms part of the TRM, which determines manoeuvre advisories based on the estimated encounter geometry between the ownship and nearby intruder aircraft.

Figure 2 illustrates the update cycle of the Horizontal Threat Resolution Module. The process begins with surveillance information from the STM, which provides estimates of the relative states of nearby intruders and the ownship. This information is passed to the TRM through the `stm_input` structure.

Several preprocessing steps then update the internal encounter representation. The functions `HorizontalTRMUpdatePrep()` and `HorizontalStateEstimation()` process the surveillance report and update the state variables used by the advisory logic, while `UpdateIntruderHRC()` retrieves coordination data associated with each intruder aircraft.

Before threat evaluation, validity checks ensure that the ownship has valid navigation data, that the TRM is operating in an advisory-capable mode, and that the aircraft is above the minimum altitude threshold. If these conditions are not satisfied the system defaults to a *Clear of Conflict* (COC) advisory.

If advisory generation is permitted, the algorithm determines whether any valid intruder aircraft are present. When intruders are detected, `PrioritizeAndFilterIntruders()` ranks potential threats and selects the highest-priority encounter for evaluation.

The advisory decision is then computed using `SelectHorizontalAdvisory()`, which evaluates the encounter geometry and determines whether a manoeuvre is required. Possible outputs include maintaining the current trajectory (*COC*), executing a *Turn Left*, or executing a *Turn Right* manoeuvre.

Additional filtering improves robustness to measurement noise and transient state variations. The function `UpdatePolicySpeedBins()` stores intruder speed information to stabilise advisory selection across update cycles. The final advisory is formatted through `HorizontalDisplayLogicDetermination()` and returned as an `HTRMReport` by `GenerateHTRMOutput()`.

The resulting report contains the selected horizontal advisory together with coordination and display information used by the DAA system. While the horizontal logic captures a core component of ACAS-Xu advisory generation, it represents only one part of the complete system, which also incorporates vertical resolution advisories and additional coordination mechanisms.

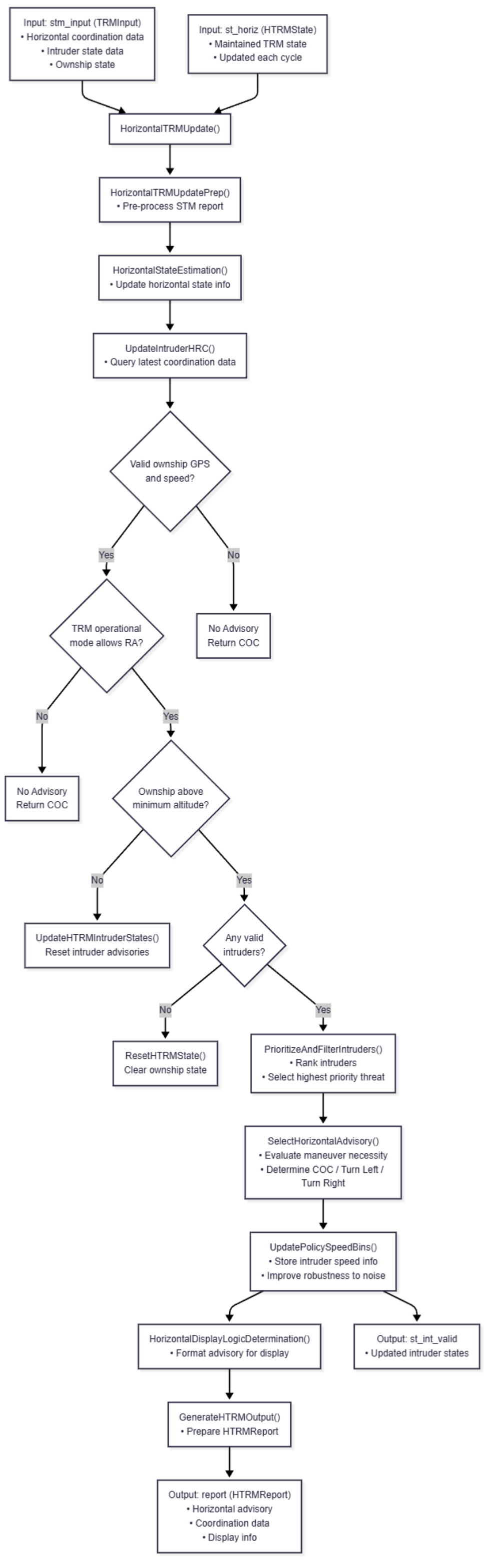

Input: stm_input (TRMInput)
• Horizontal coordination data
• Intruder state data
• Ownship state
Input: st_horiz (HTRMState)
• Maintained TRM state
• Updated each cycle
HorizontalTRMUpdate()
HorizontalTRMUpdatePrep()
• Pre-process STM report
HorizontalStateEstimation()
• Update horizontal state info
UpdateIntruderHRC()
• Query latest coordination data
Valid ownship GPS and speed?
Yes
No
No Advisory
Return COC
TRM operational mode allows RA?
No
Yes
No Advisory
Return COC
Ownship above minimum altitude?
No
Yes
UpdateHTRMIntruderStates()
Reset intruder advisories
Any valid intruders?
No
Yes
ResetHTRMState()
Clear ownship state
PrioritizeAndFilterIntruders()
• Rank intruders
• Select highest priority threat
SelectHorizontalAdvisory()
• Evaluate maneuver necessity
• Determine COC / Turn Left / Turn Right
UpdatePolicySpeedBins()
• Store intruder speed info
• Improve robustness to noise
HorizontalDisplayLogicDetermination()
• Format advisory for display
Output: st_int_valid
• Updated intruder states
GenerateHTRMOutput()
• Prepare HTRMReport
Output: report (HTRMReport)
• Horizontal advisory
• Coordination data
• Display info

Figure 2: Horizontal Threat Resolution Module update process derived from ACAS-Xu algorithm design documentation.

3.5 Existing Comparative Evaluation of ACAS-Xu and DAIDALUS

A significant comparative study between ACAS-Xu and the NASA DAIDALUS reference implementation was conducted using scripted flight-test encounters during ACAS-Xu Flight Test 2 (FT2) [9]. The study evaluated alert timing, horizontal guidance bands, and directive manoeuvre behaviour under realistic surveillance conditions.

Results indicated that ACAS-Xu's Remain Well Clear (RWC) alerts occurred at timelines comparable to DAIDALUS corrective alerts, while ACAS-Xu's Collision Avoidance (CA) alerts were issued later than DAIDALUS warning alerts. ACAS-Xu guidance was generally more conservative, protecting a wider range of headings. The study further demonstrated that deterministic projection methods in DAIDALUS exhibited alert toggling under noisy vertical state estimates, whereas ACAS-Xu's probabilistic framework maintained stable alerting behaviour.

However, the comparison was conducted on a limited set of scripted encounters rather than statistically representative Monte Carlo encounter distributions. The study therefore provides valuable behavioural insight but does not establish probabilistic performance equivalence under controlled and matched simulation assumptions.

Additionally, discrepancies were observed in directive guidance sense under certain geometries, indicating sensitivity to cost weighting and optimisation horizon assumptions [9]. These findings highlight the importance of systematic benchmarking and sensitivity analysis when evaluating optimisation-based collision avoidance systems.

3.6 Literature Synthesis and Identified Research Gap

The existing body of literature establishes a clear technical progression in airborne collision avoidance and DAA system development.

Legacy ACAS II provides a certified and operationally validated baseline for vertical collision avoidance in cooperative, crewed aviation environments [1]. Its deterministic threshold-based logic has demonstrated a reduced mid-air collision risk, but its design assumptions reflect structured airspace, pilot compliance, and cooperative surveillance.

ACAS X represents a methodological shift from threshold logic to probabilistic optimisation. By framing collision avoidance as a Markov Decision Process, ACAS X enables explicit minimisation of expected cumulative risk under uncertainty [3]. This framework enables formal trade-offs between safety performance and operational burden previously embedded implicitly in tuned thresholds.

ACAS sXu extends this approach to decentralised small UAS environments through dynamic protection volumes, autonomous response modelling, and horizontal–vertical decomposition to maintain computational tractability [2].

In parallel, DAA evaluation methodologies have evolved toward probabilistic Safety Risk Management (SRM) frameworks. Monte Carlo simulation across large encounter sets is widely accepted for evaluating Loss of Well Clear (LoWC), Near Mid-Air Collision (NMAC) probability, and Risk Ratio with statistical confidence [4], [5].

Comparative studies between ACAS-Xu and deterministic DAA systems such as DAIDALUS have identified differences in alert timing, guidance conservatism, and robustness to surveillance uncertainty [9]. However, these analyses relied primarily on limited scripted encounters rather than statistically representative Monte Carlo encounter distributions.

The ACAS-Xu algorithm design documentation further describes the internal system structure, including the Surveillance and Tracking Module (STM) and the Threat Resolution Module (TRM), which performs advisory selection based on encounter geometry and policy evaluation.

Despite these advances, several gaps remain in the literature:

1. Limited end-to-end benchmarking of ACAS-Xu within complete DAA architectures using matched encounter sets and consistent performance metrics.

2. Insufficient sensitivity analysis of cost-weight selection, encounter model assumptions, and surveillance uncertainty propagation.

3. Incomplete evaluation of safety–operational trade-offs using statistically defensible Monte Carlo frameworks.

This thesis addresses these gaps by implementing a customised ACAS-Xu module within an integrated DAA architecture and evaluating its performance against established baselines using controlled encounter distributions and statistically rigorous safety metrics.

4 Project Planning and Methodology

The project has been divided into three sections aligned with the due dates of the major assessment components. Each section contains specific objectives, which are outlined below, along with the subtasks planned for completion within the corresponding timeframes.

This structure was adopted to ensure the project timeline aligns as closely as possible with the fixed assessment deadlines. While individual tasks may ultimately require more or less time than initially anticipated, the overall framework has been designed with sufficient flexibility to accommodate these variations.

4.1 Stage 1: Project Proposal and Literature Review (9 Feb 2026 – 16 Mar 2026)

The objectives of this stage include:

1. Develop a strong theoretical understanding of DAA systems and the ACAS-Xu framework through targeted literature review and background research.

2. Define the scope, architecture, and operational assumptions of the prototype collision avoidance system.

3. Begin early development of the horizontal avoidance logic and associated encounter generation tools to reduce implementation risk.

4.1.1 Literature Review and Background Research

This stage will begin with targeted research into airborne collision avoidance systems and DAA technologies used for unmanned aircraft systems. Particular focus will be placed on the development of the ACAS-X family of algorithms and the ACAS-Xu variant designed for unmanned aircraft. The literature review will examine the transition from rule-based systems such as TCAS to the probabilistic decision-making framework used in ACAS-X, which models collision avoidance as a Markov Decision Process.

Existing implementations and simulation frameworks will also be reviewed, including publicly available research code and repositories. This investigation will help establish how ACAS-Xu style avoidance logic is implemented in practice and identify common approaches for representing aircraft encounters, state estimation, and advisory generation. The results of this research will form the technical background and literature review sections of the project proposal.

4.1.2 System Scope and Architecture Definition

Following the literature review, the operational assumptions and scope of the prototype system will be defined. This includes identifying the key variables that describe aircraft encounters, such as relative position, heading, velocity, and time to potential conflict. System inputs and outputs will also be established, including the information required from surveillance or tracking systems and the avoidance manoeuvres that the algorithm may generate.

High-level flow charts will be developed to illustrate the overall DAA decision process and the interaction between the major components of the system. These diagrams will provide a conceptual representation of how surveillance information, threat evaluation, and avoidance guidance interact within the prototype architecture.

4.1.3 Preliminary Implementation of Horizontal Avoidance Logic

Initial development of the avoidance software will begin during this stage, with a primary focus on implementing and testing the horizontal avoidance logic. Beginning development early in the project is critical, as software implementation and integration represent one of the largest technical risks to the project. Early development allows potential challenges in algorithm implementation, encounter representation, and simulation behaviour to be identified and addressed before later stages of the project.

Encounter scenarios will also be generated to allow preliminary testing of the avoidance logic. These scenarios will simulate representative aircraft encounters and provide a controlled environment in which the behaviour of the algorithm can be evaluated and refined.

4.2 Section 2: Progress Review and Prototype Development (16 Mar 2026 - 5 May 2026)

The objectives of this stage include:

1. Develop the remaining components of the prototype collision avoidance system, including the vertical avoidance logic.
2. Integrate all prototype modules into a complete DAA model capable of generating avoidance advisories.
3. Conduct initial benchmarking of the ACAS-Xu prototype and prepare the progress review presentation.

4.2.1 Development of Vertical Avoidance Logic

Following the implementation of the horizontal avoidance logic in Stage 1, this stage will focus on the development and testing of the vertical avoidance logic. Vertical manoeuvres represent an additional dimension of conflict resolution and are an important component of the ACAS-Xu framework.

The implementation will involve extending the prototype system to consider aircraft altitude, vertical rate, and time to loss of horizontal separation when generating avoidance guidance. Preliminary testing will be conducted using simulated encounters to verify that the vertical logic behaves consistently with the design assumptions and interacts appropriately with the horizontal avoidance logic.

4.2.2 Integration of Prototype Modules

A key task in this stage is the integration of all prototype components into a complete DAA system. This involves combining the horizontal and vertical avoidance logic with the encounter generation framework and the overall system architecture developed in Stage 1.

This integration step is critical to the success of the project, as a fully integrated system is required before any meaningful evaluation or benchmarking can take place. Ensuring that all modules interact correctly and produce coherent avoidance advisories will therefore be a major focus of this stage. Significant emphasis will be placed on validating the interaction between system components and resolving any issues that arise during integration.

4.2.3 Initial Benchmarking and Progress Review Preparation

Once a complete prototype system has been successfully integrated, initial benchmarking of the ACAS-Xu model will be conducted. This benchmarking will involve running the avoidance algorithm against a set of representative encounter scenarios in order to observe its behaviour and evaluate its ability to generate effective avoidance manoeuvres.

The results from this preliminary evaluation will be used to assess the basic performance of the prototype system and identify areas that may require refinement in later stages of the project. In parallel with these activities, a progress review presentation will be developed to summarise the work completed, demonstrate the functionality of the prototype, and outline the remaining tasks required to complete the project. Initial drafting of the final report and documentation of the prototype system will also begin during this stage.

4.3 Section 3: Results Analysis, Final Benchmarking and Project Completion (5 May 2026 - 2 Jul 2026)

The objectives of this stage include:

1. Analyse the results obtained from the initial benchmarking of the ACAS-Xu prototype and identify areas for improvement.
2. Refine the prototype model and conduct final benchmarking against alternative approaches.
3. Evaluate the overall system performance and communicate the results through the oral presentation and final report.

4.3.1 Analysis of Initial Results and Prototype Refinement

Following the initial benchmarking conducted in Stage 2, the results will be analysed to assess the behaviour and performance of the prototype collision avoidance system. Particular attention will be given to the effectiveness of the generated avoidance manoeuvres and the overall behaviour of the system during representative aircraft encounters.

Based on this analysis, refinements will be made to the prototype model where necessary. These refinements may include adjustments to encounter modelling assumptions, avoidance logic parameters, or improvements to the interaction between system components. The objective of this process is to improve the reliability and performance of the model prior to conducting the final benchmarking stage.

4.3.2 Final Benchmarking and System Performance Evaluation

Once the refined prototype model has been established, final benchmarking will be conducted to evaluate the performance of the ACAS-Xu implementation. This benchmarking will involve comparing the behaviour of the developed prototype against alternative approaches or baseline avoidance strategies across a range of simulated encounter scenarios.

The results of these experiments will be analysed to evaluate the overall effectiveness of the system, including its ability to generate safe and appropriate avoidance manoeuvres. The benchmarking outcomes will form a key component of the technical evaluation of the project and will provide quantitative evidence supporting the performance of the prototype system.

4.3.3 Oral Presentation Preparation and Final Report Completion

The final stage of the project will focus on communicating the outcomes of the work through both the oral presentation and the final written report. Preparation for the oral presentation will involve summarising the methodology, prototype development, benchmarking results, and key findings of the project in a clear and structured format.

In parallel, drafting of the final report will continue, incorporating the results of the benchmarking and system evaluation. The report will document the full development process of the ACAS-Xu prototype, including the theoretical background, system architecture, implementation methodology, and performance evaluation. The final report will then be reviewed, refined, and submitted as the final deliverable of the project.

4.4 Project Timeline Overview

The staged structure outlined in this methodology provides a clear progression from theoretical understanding to prototype development and finally to system evaluation and reporting. By dividing the project into three distinct phases aligned with the major assessment milestones, the approach ensures that conceptual development, software implementation, and experimental evaluation occur in a logical and manageable sequence.

Early emphasis is placed on developing a strong theoretical foundation and establishing the system architecture, which reduces technical risk during the later implementation stages. The second stage focuses on the development and integration of the prototype DAA system, ensuring that a complete and functional model is available before performance evaluation is undertaken. The final stage concentrates on analysing the behaviour of the system, refining the prototype where necessary, and conducting benchmarking experiments to assess the effectiveness of the developed collision avoidance approach.

A detailed schedule of the tasks described in this section is provided in the project Gantt chart included in Appendix A. The Gantt chart illustrates the planned sequencing of activities, key milestones, and overlaps between development, evaluation, and reporting tasks throughout the duration of the project. This timeline serves as a reference framework for managing progress and ensuring that each stage of the project contributes effectively to the final objectives.

5 Project Risks and Opportunities

This section identifies potential risks that may affect the successful completion of the project. The severity of each risk is assessed using the project risk matrix, with associated tolerance levels and response requirements, as presented in Appendix B. The risk matrix and associated tolerance tables were developed with reference to The University of Queensland Health and Safety Risk Assessment Procedure [10].

The identified risks are grouped into three main categories: workplace risks, scheduling risks, and technical risks. Workplace risks relate to general work activities undertaken during the project, scheduling risks relate to delays in development or completion of the project, and technical risks relate to challenges associated with implementation of the ACAS-Xu prototype.

5.1 Workplace Risks

Workplace risks refer to hazards that may arise during the normal working activities associated with the project. As the majority of the work will involve computer-based development and analysis, the overall level of physical risk is relatively low.

Table 2: Workplace Risks

Risk	**Consequence**	**Risk level**	**Prevention**	**Mitigation**
Traffic incident while commuting	Personal injury or inability to attend work.	**Medium**	• Follow road safety rules. • Avoid driving while fatigued.	• Seek medical assistance if required. • Notify supervisor of absence.
Prolonged sitting during development work	Back pain or musculoskeletal strain.	**Low**	• Maintain correct workstation ergonomics. • Take regular breaks from sitting.	• Adjust workstation setup. • Seek ergonomic advice if discomfort persists.
Extended screen use	Eye strain or headaches.	**Low**	• Take regular breaks from the screen. • Maintain appropriate screen brightness and distance.	• Rest eyes and reduce screen time temporarily. • Seek medical advice if symptoms persist.
Fire or emergency incident at workplace	Risk of injury or evacuation of workplace.	**Medium**	• Follow workplace safety procedures. • Be aware of emergency exits and evacuation plans.	• Follow evacuation procedures and emergency instructions.

5.2 Scheduling Risks

Scheduling risks refer to events that may delay the progress of the project or affect completion within the allocated timeframe. As the project involves software development and system integration, some tasks may require more time than initially anticipated.

Table 3: Scheduling Risks

Risk	**Consequence**	**Risk level**	**Prevention**	**Mitigation**
Development tasks take longer than expected	Project milestones may be delayed, reducing the time available for benchmarking and analysis.	**Medium**	• Begin development tasks early. • Break work into smaller milestones.	• Prioritise critical components of the prototype. • Reduce scope of non-essential features if necessary.
Integration and debugging delays	Additional time may be required to resolve unexpected behaviour between system modules.	**Medium**	• Test individual components early. • Maintain modular software design.	• Allocate additional debugging time. • Simplify system architecture if necessary.

Risk	**Consequence**	**Risk level**	**Prevention**	**Mitigation**
Software crash or loss of work	Loss of development time and potentially unrecoverable progress.	**Medium**	• Maintain frequent backups. • Use version control for development.	• Restore work from latest backup or repository version.
Project incomplete at final submission date	The prototype system may not be fully implemented by the end of the placement period.	**Medium**	• Monitor progress against the project schedule. • Prioritise essential project objectives.	• Ensure documentation and methodology are sufficiently detailed so development can continue after project completion.

5.3 Technical Risks

Technical risks arise from challenges associated with the implementation and evaluation of the ACAS-Xu prototype system and simulation environment.

Table 4: Technical Risks

Risk	**Consequence**	**Risk level**	**Prevention**	**Mitigation**
Encounter dataset unavailable or incompatible	Simulation scenarios may not represent realistic aircraft encounters, reducing the validity of benchmarking results.	**Low**	• Identify available encounter datasets early in the project.	• Use alternative publicly available encounter models if required.
Integration challenges between avoidance modules	Difficulty combining horizontal and vertical avoidance logic may delay prototype completion.	**Medium**	• Develop and test modules independently. • Maintain clear system architecture.	• Simplify model interactions or isolate problematic modules.
Simulation results difficult to validate	Prototype behaviour may be difficult to compare with published ACAS-Xu results.	**Medium**	• Review literature and reference implementations.	• Validate model behaviour using simplified encounter scenarios.
Software environment or dependency issues	External libraries or simulation tools may fail to install or operate correctly.	**Low**	• establish development environment early in the project.	• modify environment configuration or replace incompatible tools.

5.4 Opportunities

In addition to the identified risks, the project presents several opportunities related to the development and evaluation of unmanned aircraft collision avoidance systems. The implementation of a prototype framework inspired by ACAS-Xu enables the structured simulation of aircraft encounter scenarios and the evaluation of collision avoidance strategies under representative conditions. This provides an opportunity to systematically benchmark advisory behaviour and assess how different manoeuvre strategies influence collision risk, contributing to improved understanding of detect-and-avoid system performance for unmanned aircraft.

Initial research has also identified a neural network implementation of ACAS-Xu available through the ACASXu Closed-Loop Simulator repository [11]. If development time permits toward the end of the project, this approach could be trialled within the simulation environment to evaluate whether it offers any advantages over the traditional lookup table implementation. Such a comparison may provide insight into the potential benefits of neural network policy representations, particularly in terms of computational efficiency and scalability for future collision avoidance system development.

References

[1] International Civil Aviation Organization, "Airborne collision avoidance system (acas) manual," International Civil Aviation Organization, Montreal, QC, Canada, Tech. Rep. Doc 9863 AN/461, 2006, First Edition.

[2] L. E. Alvarez, I. Jessen, M. P. Owen, J. Silbermann, and P. Wood, "Acas sxu: Robust decentralized detect and avoid for small unmanned aircraft systems," in *IEEE/AIAA Digital Avionics Systems Conference (DASC)*, IEEE, 2019.

[3] M. J. Kochenderfer, "On estimating mid-air collision risk," MIT Lincoln Laboratory, Lexington, MA, USA, Tech. Rep., 2008.

[4] MIT Lincoln Laboratory, "Detect and avoid phase 2: Simulation safety risk management document (srmd) final report," MIT Lincoln Laboratory, Lexington, MA, USA, Tech. Rep., 2021.

[5] RTCA, "Safety risk management (srm) modeling and simulation guidance for detect and avoid systems," RTCA, Inc., Washington, DC, USA, Tech. Rep., 2022.

[6] C. Munoz et al., "Daidalus: Detect and avoid alerting logic for unmanned systems," in *IEEE/AIAA Digital Avionics Systems Conference (DASC)*, NASA Technical Reports Server, Document ID 20160007498, 2015.

[7] M. J. Kochenderfer and J. P. Chryssanthacopoulos, "Robust airborne collision avoidance through dynamic programming," MIT Lincoln Laboratory, Lexington, MA, Tech. Rep. ATC-371, Jan. 2011, Prepared for the Federal Aviation Administration (FAA).

[8] Traffic Alert and Collision Avoidance System (TCAS) Program Office, "Algorithm design description of the airborne collision avoidance system xu," Federal Aviation Administration, Technical Report ACAS ADS-21-001 V3R0a, 2021, ACAS-Xu Algorithm Design Description.

[9] J. T. Davies and M. G. Wu, "Comparative analysis of acas-xu and daidalus detect-and-avoid systems," NASA Ames Research Center, Tech. Rep. NASA/TM–2018–219773, 2018.

[10] The University of Queensland, *Health and safety risk assessment procedure*, Accessed: 11 March 2026, 2026. [Online]. Available: `https://policies.uq.edu.au/document/view-current.php?id=318`

[11] S. Bak, *Acasxu closed loop simulator*, `https://github.com/stanleybak/acasxu_closed_loop_sim`, Accessed: 2026-03-01, 2020.

[12] L. E. Alvarez, I. Jessen, M. P. Owen, J. Silbermann, and P. Wood, "Acas sxu: Robust decentralized detect and avoid for small unmanned aircraft systems," MIT Lincoln Laboratory and Johns Hopkins University Applied Physics Laboratory, Lexington, MA and Baltimore, MD, USA, Tech. Rep., 2019.

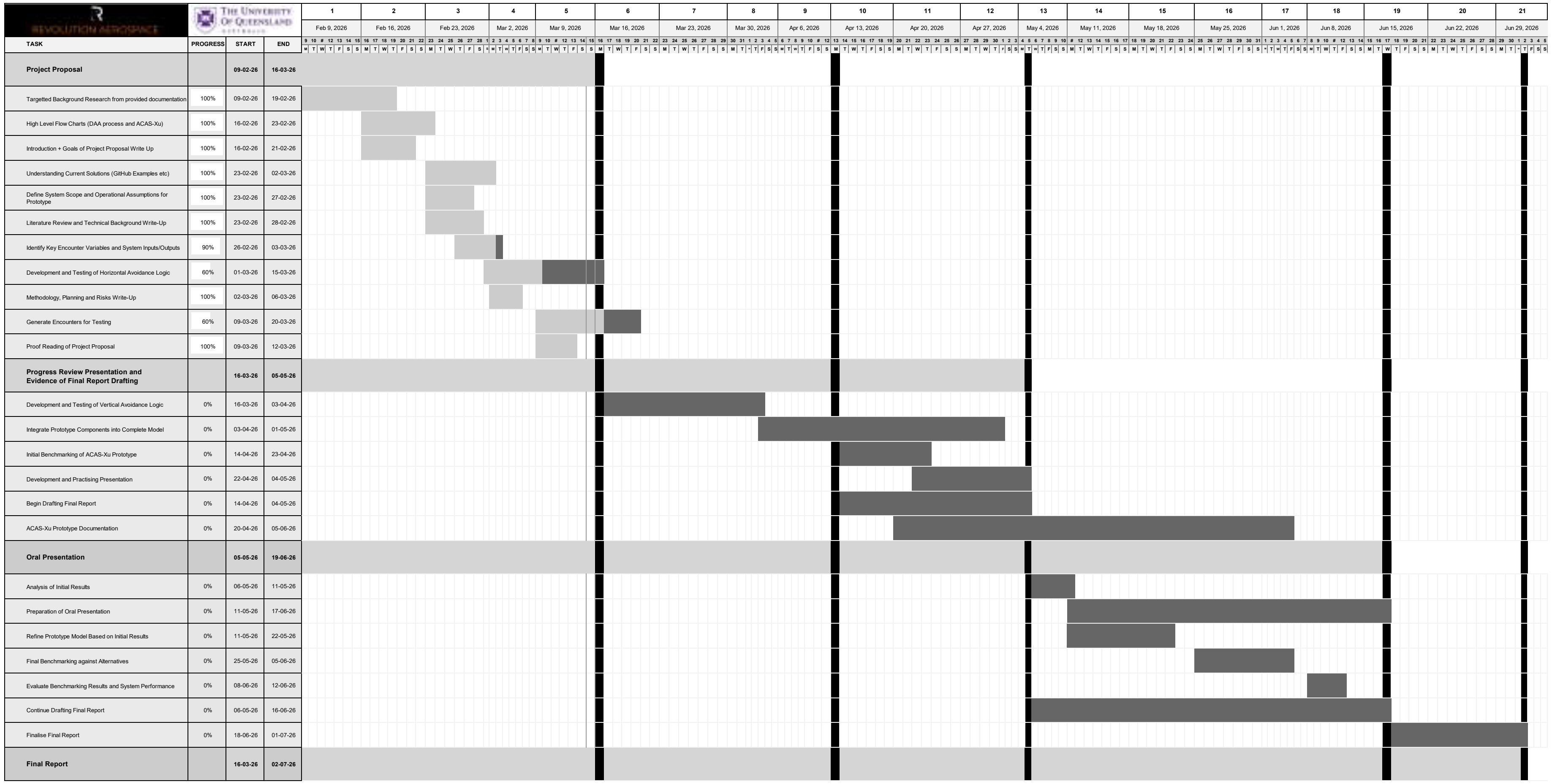
[13] D. Smith and et al., "Human factors considerations in acas xu implementation," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 2020.

[14] F. Kunzi, "Development of a high precision ads-b based conflict alerting system for operations in the airport environment," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, 2013.

[15] RTCA Special Committee 147, "Minimum operational performance standards for airborne collision avoidance system xu (acas xu) (vol ii: Algorithm design description)," RTCA, Inc., Tech. Rep. DO-386 Volume II, 2020, Prepared by SC-147 Traffic Alert & Collision Avoidance System (TCAS).

A Project Schedule (Gantt Chart)

The Gantt chart presented in Figure 3 outlines the planned schedule for the development, testing, and evaluation of the ACAS-Xu prototype throughout the duration of the project.



TASK	PROGRESS	START	END
Project Proposal		**09-02-26**	**16-03-26**
Targetted Background Research from provided documentation	100%	09-02-26	19-02-26
High Level Flow Charts (DAA process and ACAS-Xu)	100%	16-02-26	23-02-26
Introduction + Goals of Project Proposal Write Up	100%	16-02-26	21-02-26
Understanding Current Solutions (GitHub Examples etc)	100%	23-02-26	02-03-26
Define System Scope and Operational Assumptions for Prototype	100%	23-02-26	27-02-26
Literature Review and Technical Background Write-Up	100%	23-02-26	28-02-26
Identify Key Encounter Variables and System Inputs/Outputs	90%	26-02-26	03-03-26
Development and Testing of Horizontal Avoidance Logic	60%	01-03-26	15-03-26
Methodology, Planning and Risks Write-Up	100%	02-03-26	06-03-26
Generate Encounters for Testing	60%	09-03-26	20-03-26
Proof Reading of Project Proposal	100%	09-03-26	12-03-26
Progress Review Presentation and Evidence of Final Report Drafting		**16-03-26**	**05-05-26**
Development and Testing of Vertical Avoidance Logic	0%	16-03-26	03-04-26
Integrate Prototype Components into Complete Model	0%	03-04-26	01-05-26
Initial Benchmarking of ACAS-Xu Prototype	0%	14-04-26	23-04-26
Development and Practising Presentation	0%	22-04-26	04-05-26
Begin Drafting Final Report	0%	14-04-26	04-05-26
ACAS-Xu Prototype Documentation	0%	20-04-26	05-06-26
Oral Presentation		**05-05-26**	**19-06-26**
Analysis of Initial Results	0%	06-05-26	11-05-26
Preparation of Oral Presentation	0%	11-05-26	17-06-26
Refine Prototype Model Based on Initial Results	0%	11-05-26	22-05-26
Final Benchmarking against Alternatives	0%	25-05-26	05-06-26
Evaluate Benchmarking Results and System Performance	0%	08-06-26	12-06-26
Continue Drafting Final Report	0%	06-05-26	16-06-26
Finalise Final Report	0%	18-06-26	01-07-26
Final Report		**16-03-26**	**02-07-26**

Figure 3: Project Gantt Chart

B Project Risk Assessment

This appendix presents the qualitative risk assessment framework used for the project. The framework classifies risks according to their likelihood and consequence, allowing risks to be prioritised and appropriate mitigation actions to be identified.

		Consequence				
		Insignificant [1]	**Minor [2]**	**Moderate [3]**	**Major [4]**	**Critical [5]**
Type of Impact	**Health & Safety**	• No injury or medical attention required	• Minor injury requiring basic first aid	• Injury requiring professional medical care	• Serious injury requiring hospitalization	• Fatality or permanent disability
	Project & Scheduling	• No noticeable impact on schedule	• Small delay to project progress	• Measurable delay affecting delivery timeline	• Major schedule disruption or scope reduction	• Project unable to continue as planned
	Technical	• No effect on technical deliverables	• Minor reduction in technical quality	• Moderate effect on system performance	• Major degradation of system capability	• Core technical objectives cannot be achieved
Likelihood	Very High (> 90%) [5]	Medium	Medium	High	Extreme	Extreme
	High (60–90%) [4]	Low	Medium	High	High	Extreme
	Medium (40–59%) [3]	Low	Low	Medium	High	Extreme
	Low (10–39%) [2]	Low	Low	Medium	Medium	High
	Very Low (< 10%) [1]	Low	Low	Low	Medium	High

Table 6: Project Risk Matrix

Assessed Risk Level	**Tolerance**	**Immediate Response**	**Prevention**
Extreme	Intolerable	• Work must cease immediately and not continue. • Risk must be reduced to an acceptable level before resuming.	• Develop and implement a formal risk management plan. • Introduce control measures as soon as possible.
High	Low tolerance	• Activity should only proceed under exceptional circumstances. • Approval must be obtained from the supervisor or responsible authority.	• Implement mitigation procedures to reduce the risk. • Establish a documented risk management action plan.
Medium	Moderate tolerance	• Task may proceed once the risk assessment has been reviewed. • Supervisor acknowledgement or approval is recommended.	• Regularly review risks and apply mitigation where feasible. • Update the risk management plan if operating conditions change. • Monitor the operating environment for new hazards.
Low	Generally acceptable	• Work may proceed following standard risk assessment procedures.	• Maintain existing control measures. • Continue routine monitoring and maintenance of safeguards.

Table 7: Risk Tolerance Levels and Required Mitigation Actions