



THE UNIVERSITY  
OF QUEENSLAND  
AUSTRALIA

CREATE CHANGE

# UAV Guidance And Alerting Algorithm

Flynn Devoy

Student Number: 47442043

The University of Queensland / Revolution Aerospace

# Presentation Context

- UAV utilisation growing exponentially however commercial utility inhibited by inability to gain regulatory approval to fly BVLOS (Beyond Visual Line of Sight)
- Critical Precursor to BVLOS is safe separation of UAVOPS from other crewed and uncrewed aircraft
- DAA is a key technology supporting this separation and DAA has several functional elements
  1. Detection
  2. Tracking
  3. Alerting and Guidance
  4. Command and Execution
- The key content for this presentation is Alerting and Guidance

# Presentation Overview

- Project Background, Aim and Objectives
- Preliminary Challenges
- Summary of Methodology
- Preliminary Results and Analysis
- Project Progress and Next Steps
- Questions

**Figure 1:** U.S. Navy MQ-4C Triton unmanned aerial vehicle in flight. The platform is reported to incorporate advanced detect-and-avoid capabilities, including variants of the ACAS Xu collision avoidance system developed by MIT Lincoln Laboratory. Source: <https://www.navy.mil/Press-Office/News-Stories/Article/2713120/navy-conducts-first-mq-4c-triton-test-flight-with-multi-intelligence-upgrade/> navy.mi



# Project Background

- Integration into civil airspace requires meeting Equivalent Level of Safety (ELOS) standards
- Existing systems such as ACAS II were designed for crewed aviation environments and assume structured airspace and pilot response
- UAV operations introduce new challenges:
  - Mixed cooperative and non-cooperative traffic
  - Lower performance and limited manoeuvrability
  - Distributed or remote pilot control
- Traditional systems rely on deterministic or rule-based logic (e.g. DAA frameworks like DAIDALUS)
- ACAS Xu introduces an optimisation-based approach using a Markov Decision Process (MDP) to handle uncertainty and generate advisories

# Project Aim and Objectives

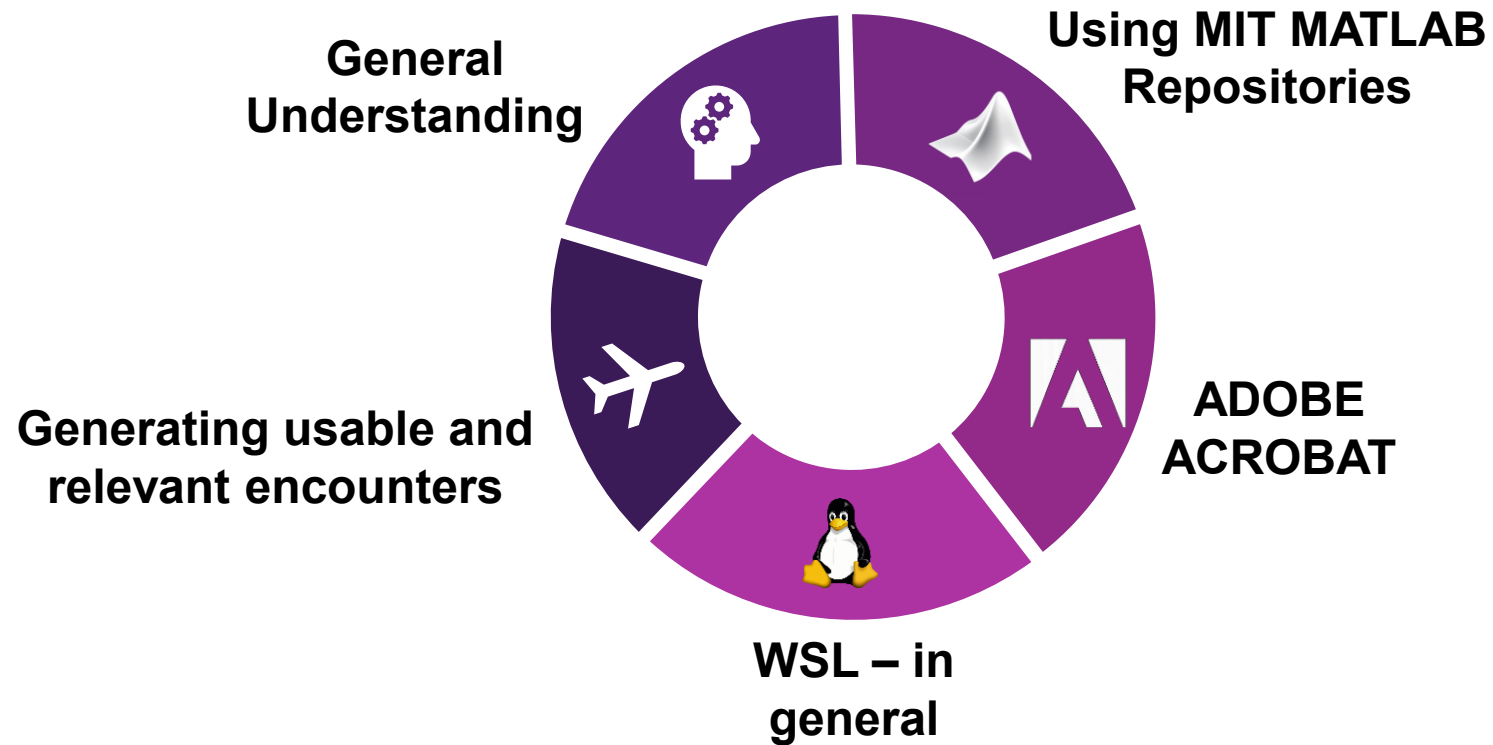
## Aim

- The aim of this project is to develop and evaluate the suitability of ACAS Xu for integration into a UAV Detect and Avoid system.

## Objectives

- Develop and Implement a functional version of the ACAS Xu collision avoidance logic
- Integrate the system within an existing DAA simulation environment
- Evaluate system performance using large-scale simulation
- Compare performance against an existing DAIDALUS-based DAA implementation (and ACAS-Xu if possible)

# Preliminary Challenges



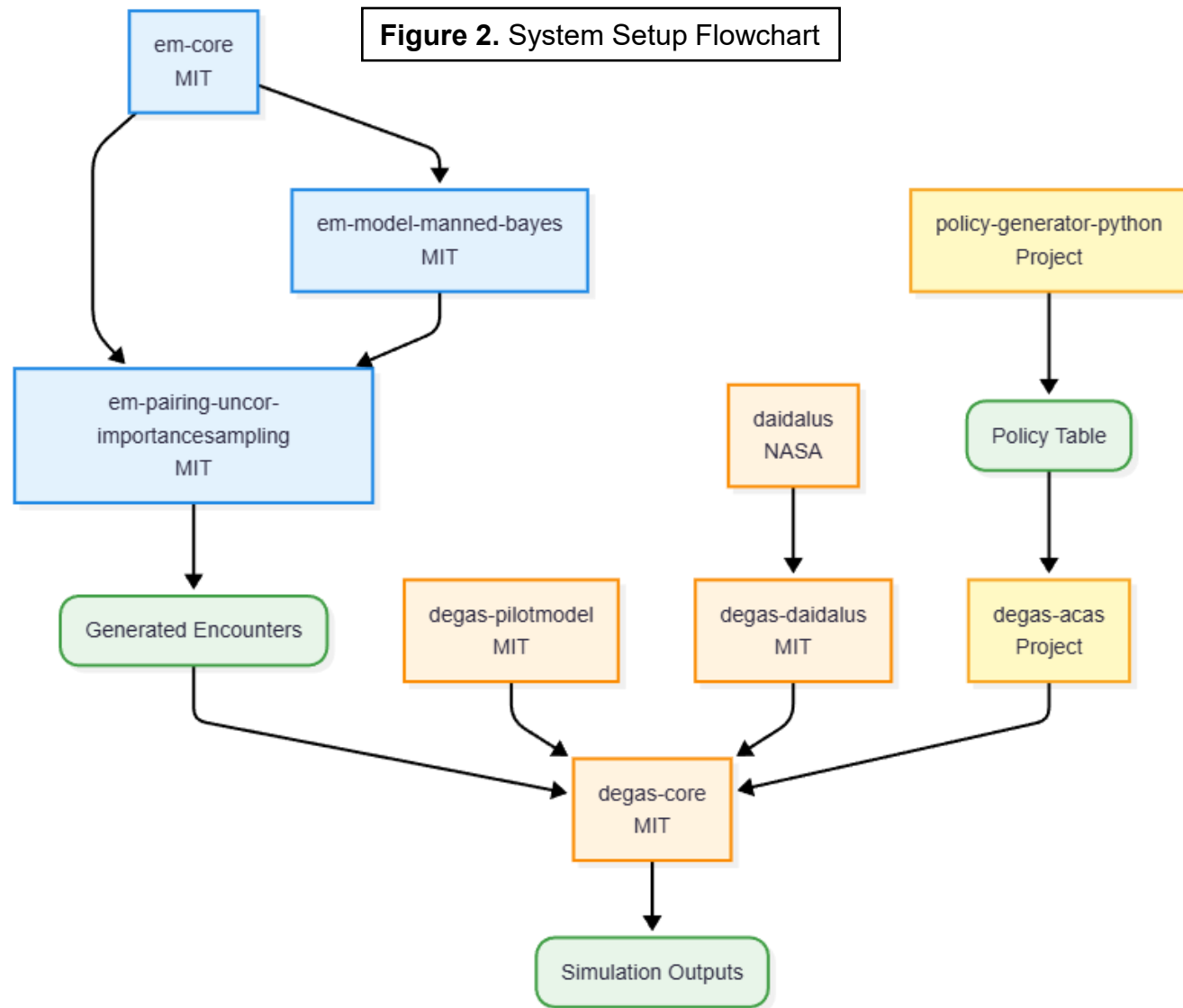
# Summary of Methodology

1. System Setup
2. Encounter Generation (em-core)
3. Testing Environment (DEGAS)
4. Vertical Policy Table Generation
5. Horizontal Policy Table Generation



# System Setup

- Blue blocks correlate to the Encounter Generation Pipeline
- Orange correlate to DEGAS Pipeline
- Yellow correlates to code I have developed
- Green are “outputs”



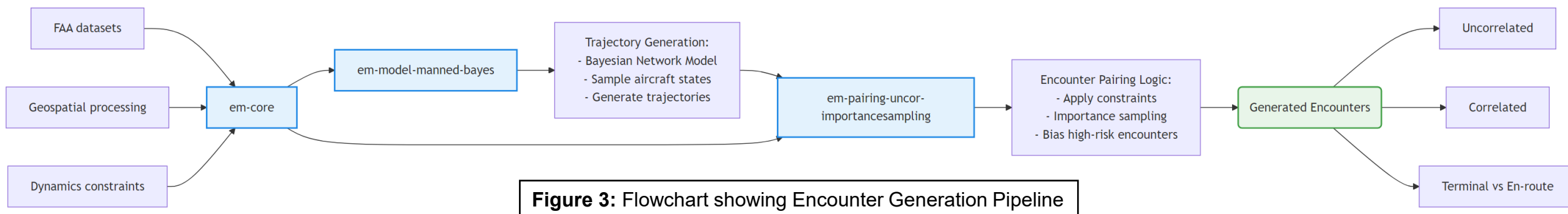
# Encounter Generation Pipeline

EM simulation framework:

- *em-core* provides core utilities and datasets

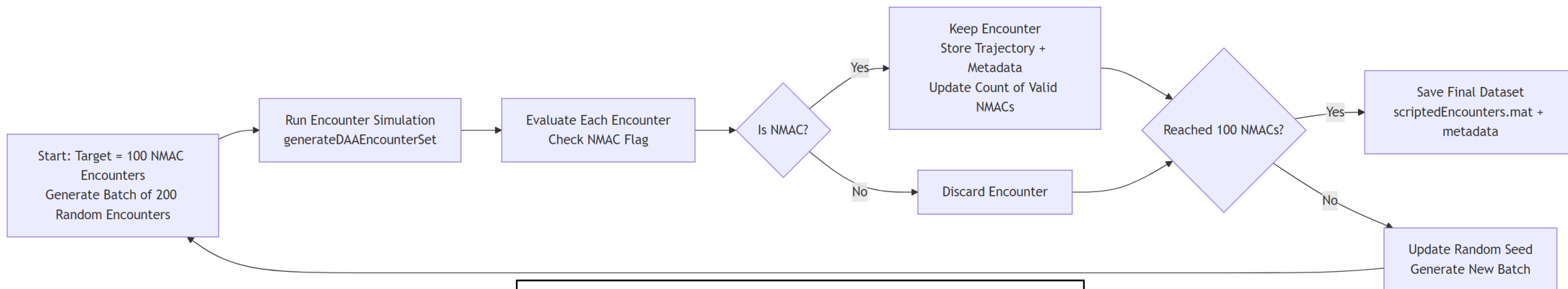
Integrated modules:

- *em-model-manned-bayes* generates realistic aircraft behaviour
- *em-pairing-uncor-importancesampling* forms encounter scenarios



# NMAC Encounters for Testing

- NMACs are extremely rare events (*Kunzi, 2013*)
- Probability per encounter is very low ( $\sim 10^{-4}$ – $10^{-6}$ ) (*Owen et al., 2016*)
- Random sampling  $\rightarrow$  mostly safe trajectories (*Kochenderfer et al., 2010*)
- Large batches enable efficient extraction of high-risk cases
- Developed a MATLAB script using the MIT pipeline to generate NMAC-only encounters
- Script is located within *em-pairing-uncor-importancesampling*



**Figure 4:** Flowchart showing Custom NMAC Generation Script

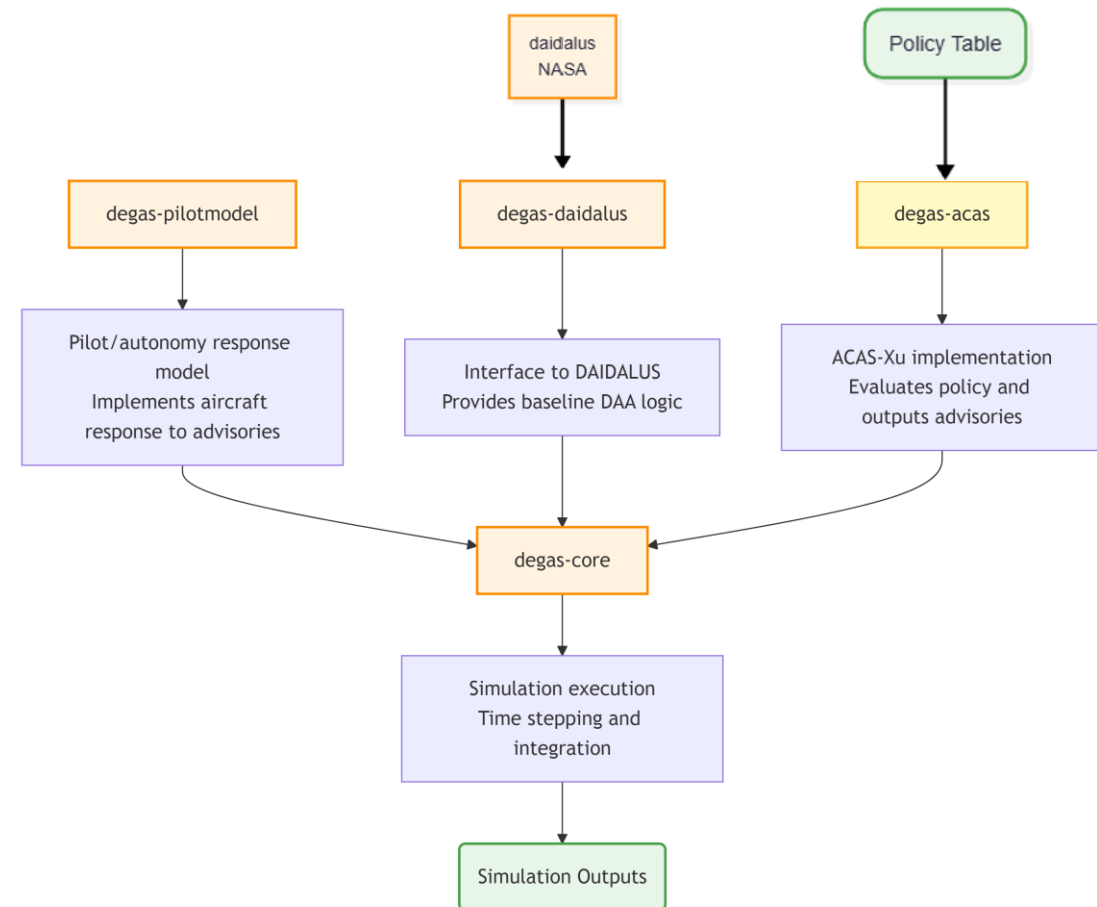
# Degas Simulation Pipeline

DEGAS simulation framework:

- *degas-core* acts as the central simulation engine

Integrated modules:

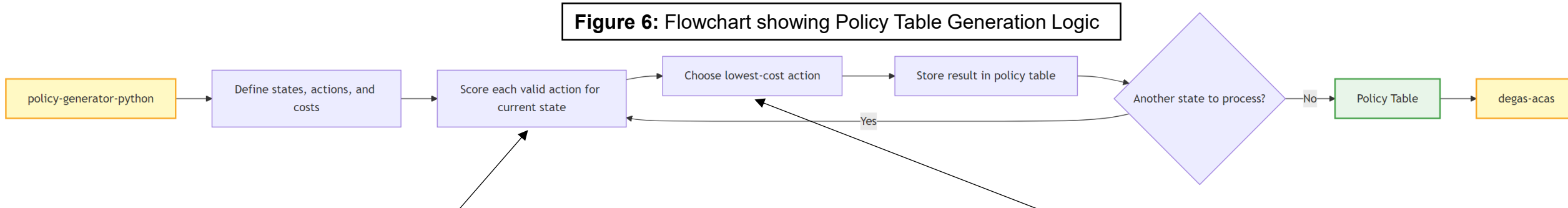
- *degas-pilotmodel* simulates aircraft response
- *degas-daidalus* provides baseline DAA logic
- *degas-acas* integrates our ACAS-Xu Policy Logic



**Figure 5:** Flowchart showing Custom NMAC Generation Script

# Policy Table Development

**Figure 6:** Flowchart showing Policy Table Generation Logic



## Cost of Action

$$Q(s, a) = c(s, a) + E[V(s')]$$

Evaluates quality of each possible action accounting for:

- Immediate cost  $c(s, a)$
- Expected value of next states  $E[V(s')]$

## Optimal Action Selection

$$V(s) = \arg \min_a Q(s, a)$$

Selects min expected cost action producing optimal policy for each state.

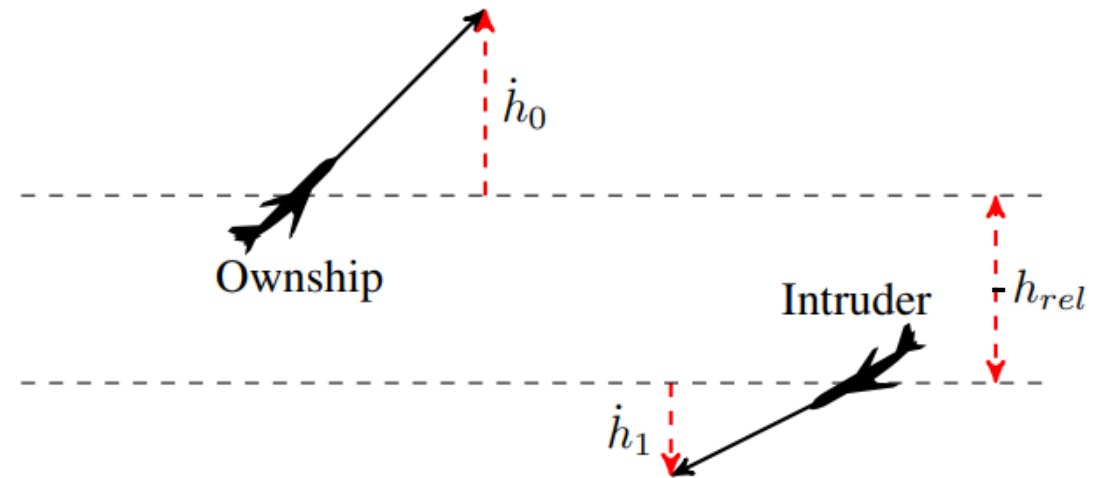
Policy Table Development based on Kochenderfer & Chryssanthacopoulos (2011)

# Vertical Policy Table Development

Table 1. Vertical Logic State Vector Descriptions and Discretisation Schema

Variable	Description	Discretisation
$h_{rel}$	Vertical Separation	-1000, -900, ..., 100 ft
$\dot{h}_0$	Own-ship Vertical Rate	-2500, -2250, ..., 2500 ft/min
$\dot{h}_1$	Intruder Vertical Rate	-2500, -2250, ..., 2500 ft/min
$\tau$	Time to loss of <b>horizontal</b> separation	0, 1, ..., 40 s
$a_{prev}$	Previously Issued Advisory	N/A (already discrete)

Figure 7. Geometry for ACAS Xu Vertical Logic Table



Vertical States based on Kochenderfer & Chryssanthacopoulos (2011)

# Vertical Policy Table Development

Table 2. Vertical Advisory Sets

Name	Min Vert Rate (ft/m)	Max Vert Rate (ft/m)	Strength (g)	Available from
COC	$-\infty$	$\infty$	0	All
DES1500	$-\infty$	1500	$1/4$	COC
CL1500	1500	$\infty$	$1/4$	COC
SDES1500	$-\infty$	1500	$1/3$	CL1500, SCL1500, SDES2500, SCL2500
SCL1500	1500	$\infty$	$1/3$	DES1500, SDES1500, SDES2500, SCL2500
SDES2500	$-\infty$	2500	$1/3$	DES1500, SDES1500
SCL2500	2500	$\infty$	$1/3$	CL1500, SCL1500

Table 3. Vertical Event Costs

NMAC	Alert	Strengthening	Reversal	Clear of Conflict
1	0.01	0.009	0.01	-0.0001

Vertical Costs based on Kochenderfer & Chryssanthacopoulos (2011)

# Initial Optimal Action Plots – a) Vertical Advisory

Example Optimal Action Plot (Kochendorfer, 2011) compared with Optimal Action Plot from generated Vertical Policy Table (Ownship Vertical Rate: 0, Intruder Vertical Rate: 0, Previous Advisory: COC)

Figure 8. Kochendorfer 2011 Optimal Action Plot (Own-ship Vertical Rate: 0, Intruder Vertical Rate: 0, Previous Advisory: COC)

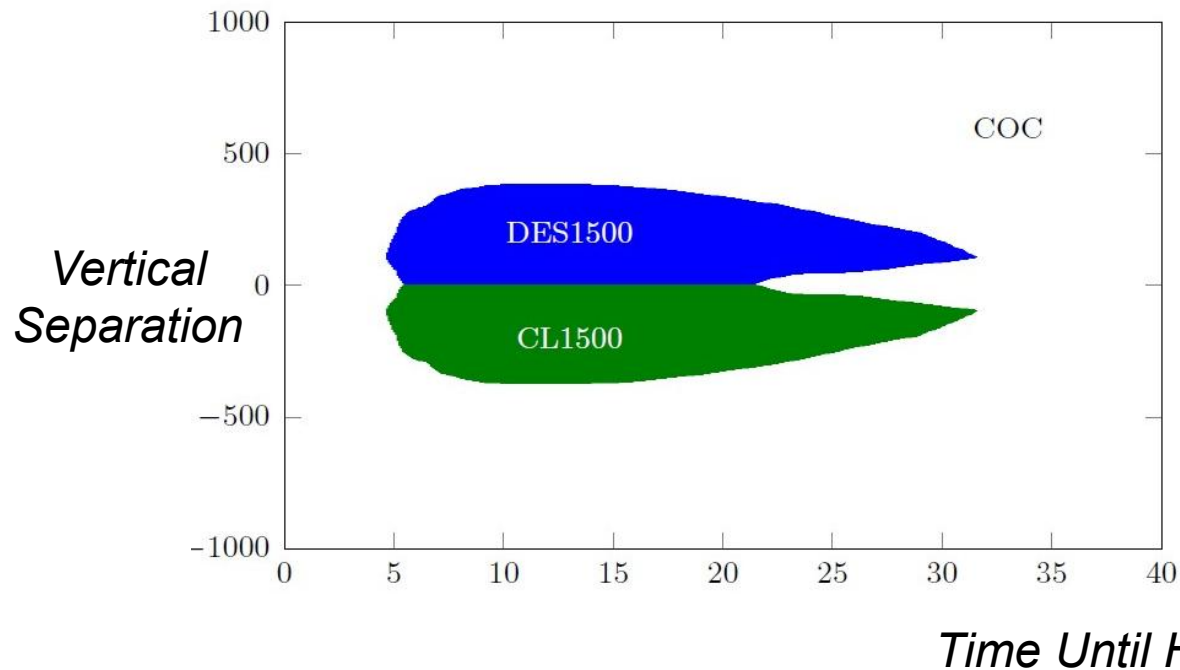
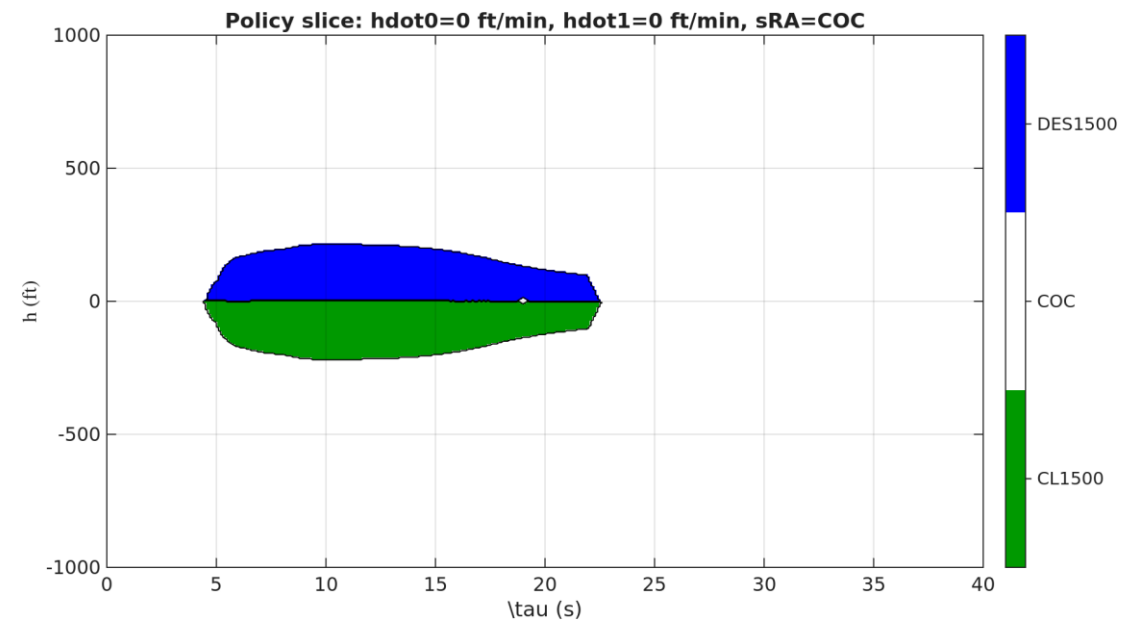


Figure 9. Devoy 2026 Optimal Action Plot Replication (Own-ship Vertical Rate: 0, Intruder Vertical Rate: 0, Previous Advisory: COC)



# Initial Optimal Action Plots – b) Vertical Advisory

Example Optimal Action Plot (Kochendorfer, 2011) compared with Optimal Action Plot from generated Vertical Policy Table (Ownship Vertical Rate: 1500, Intruder Vertical Rate: 0, Previous Advisory: COC)

Figure 10. Kochendorfer 2011 Optimal Action Plot (Own-ship Vertical Rate: 0, Intruder Vertical Rate: 0, Previous Advisory: COC)

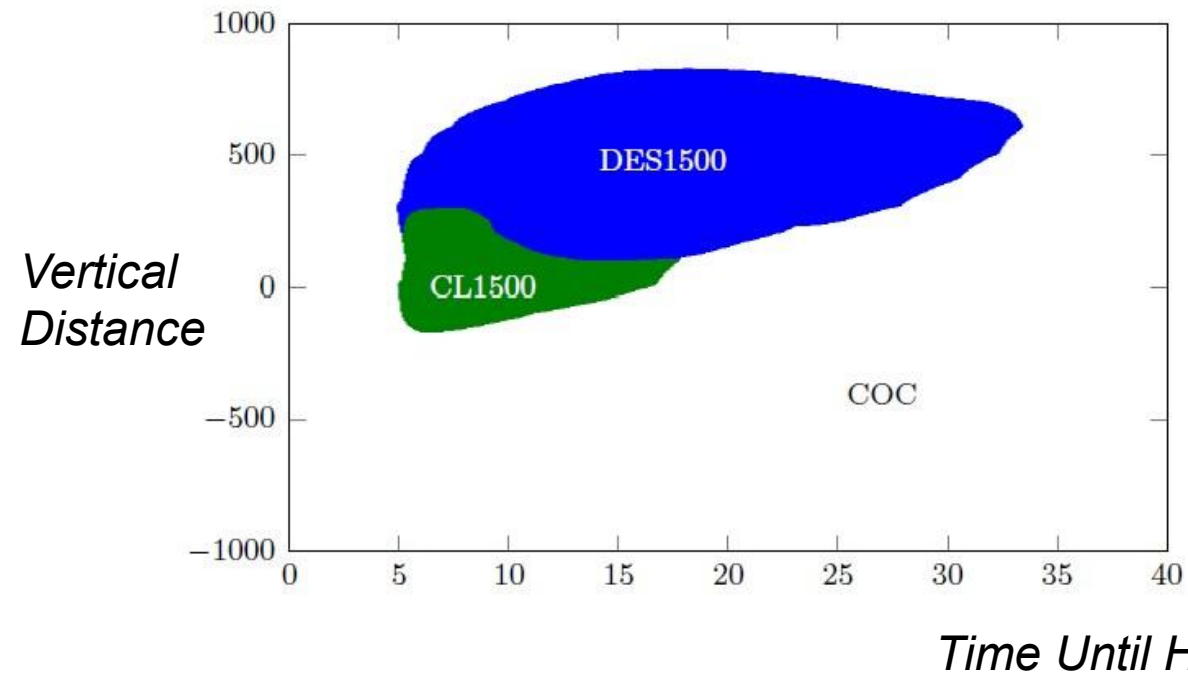
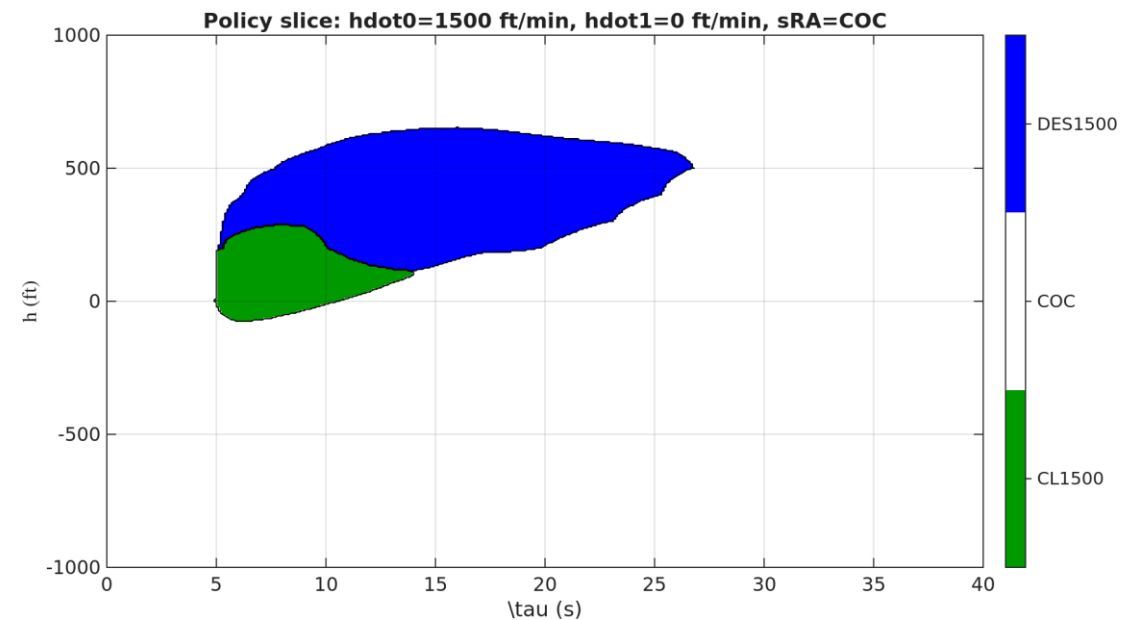


Figure 11. Devoy 2026 Optimal Action Plot Replication (Own-ship Vertical Rate: 1500, Intruder Vertical Rate: 0, Previous Advisory: COC)

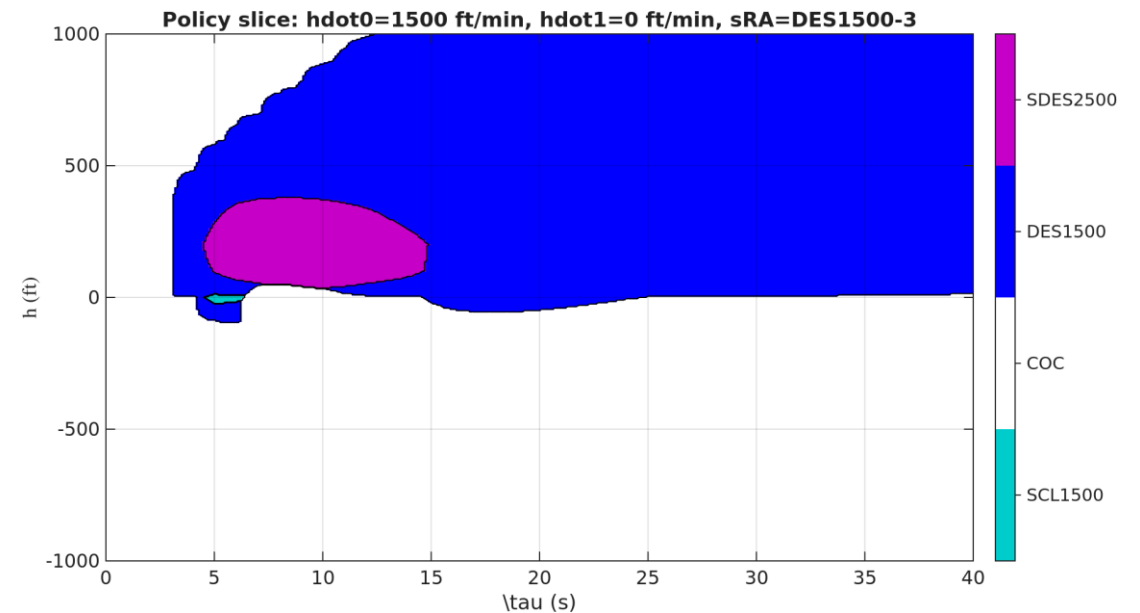
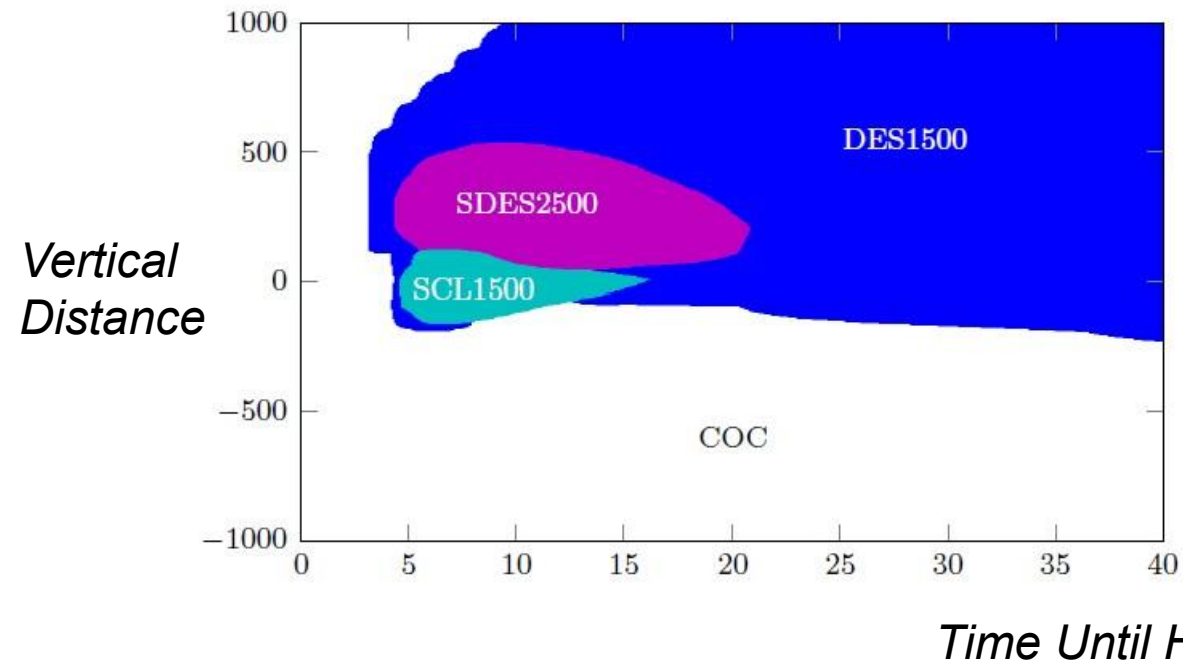


# Initial Optimal Action Plots – c) Vertical Advisory

Example Optimal Action Plot (Kochendorfer, 2011) compared with Optimal Action Plot from generated Vertical Policy Table (Ownship Vertical Rate: 1500, Intruder Vertical Rate: 0, Previous Advisory: DES1500-3)

Figure 12. Kochendorfer 2011 Optimal Action Plot (Own-ship Vertical Rate: 1500, Intruder Vertical Rate: 0, Previous Advisory: DES1500-3)

Figure 13. Devoy 2026 Optimal Action Plot Replication (Own-ship Vertical Rate: 1500, Intruder Vertical Rate: 0, Previous Advisory: DES1500-3)



# Parameter Sweep

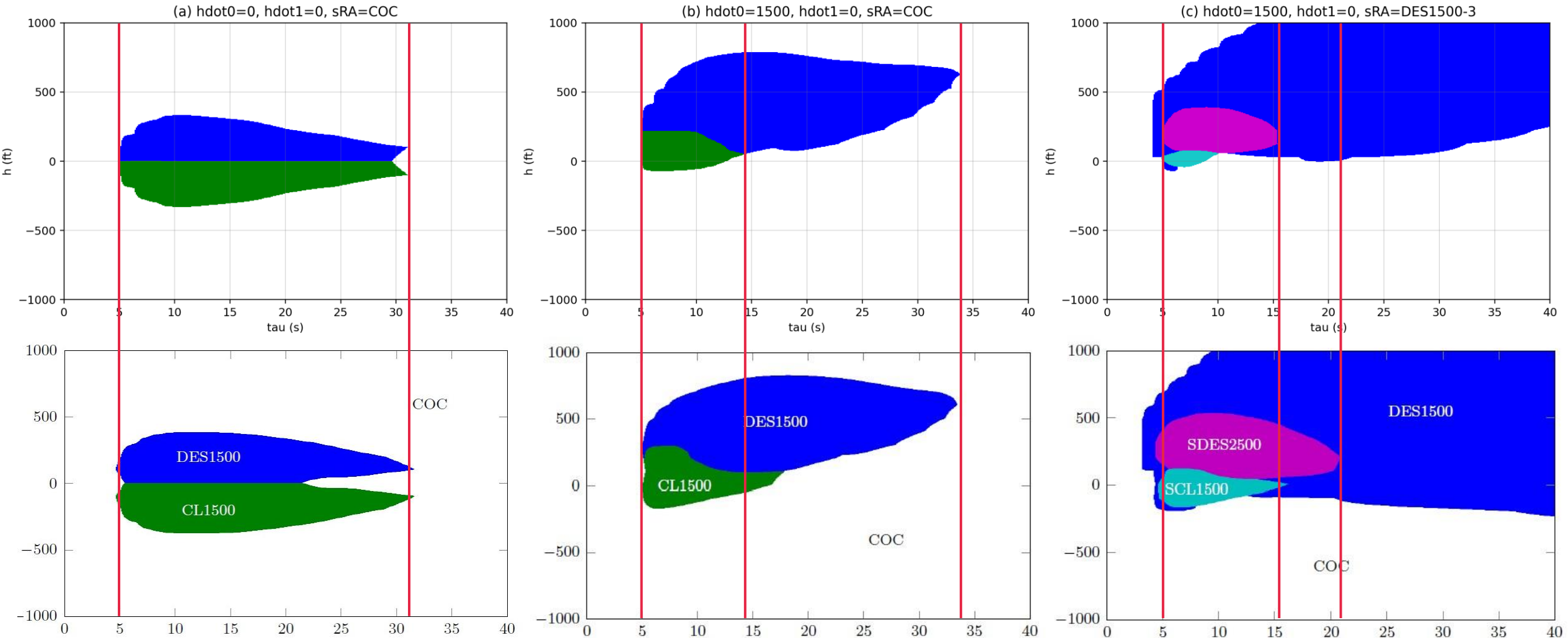
Parameter	Description	Values Tested
<i>cost-alert</i>	<i>Penalise Alerting</i>	<i>0.0002, 0.0005, 0.001, 0.002, 0.003, 0.004, 0.005, 0.0075, 0.01</i>
<i>reward-coc</i>	<i>Rewards COC Advisory</i>	<i>0.0, -1e-5, -5e-5, -1e-4</i>
<i>cost-nmac</i>	<i>Penalise NMAC Outcomes</i>	<i>Fixed at 1.0 (Same as Kochenderfer)</i>
<i>cost-strengthening</i>	<i>Penalise Strengthening Advisories (weak left -&gt; strong left)</i>	<i>Fixed at 0.009 (Same as Kochenderfer)</i>
<i>cost-reversal</i>	<i>Penalise Reversals (weak left -&gt; weak right)</i>	<i>Fixed at 0.01 (Same as Kochenderfer)</i>
<i>sigma_hdd</i>	<i>Standard deviation of vertical acceleration noise</i>	<i>Fixed at 3.0 (Same as Kochenderfer)</i>

**Formula: Score = |A-31| + |B-33| where lower is better**

**Best Parameters: *cost-alert* = 0.001, *reward-coc* = -0.0001**

# Parameter Sweep – Optimal Advisory Plots

COC
  DES1500
  CL1500
  SCL1500
  SDES2500



# Why we replicated Kochendorfers Optimal Advisory Plots

## Kochendorfer ACAS-Xu Results

Figure 14. Example encounter comparing the performance of the ACAS logic against the current TCAS logic.

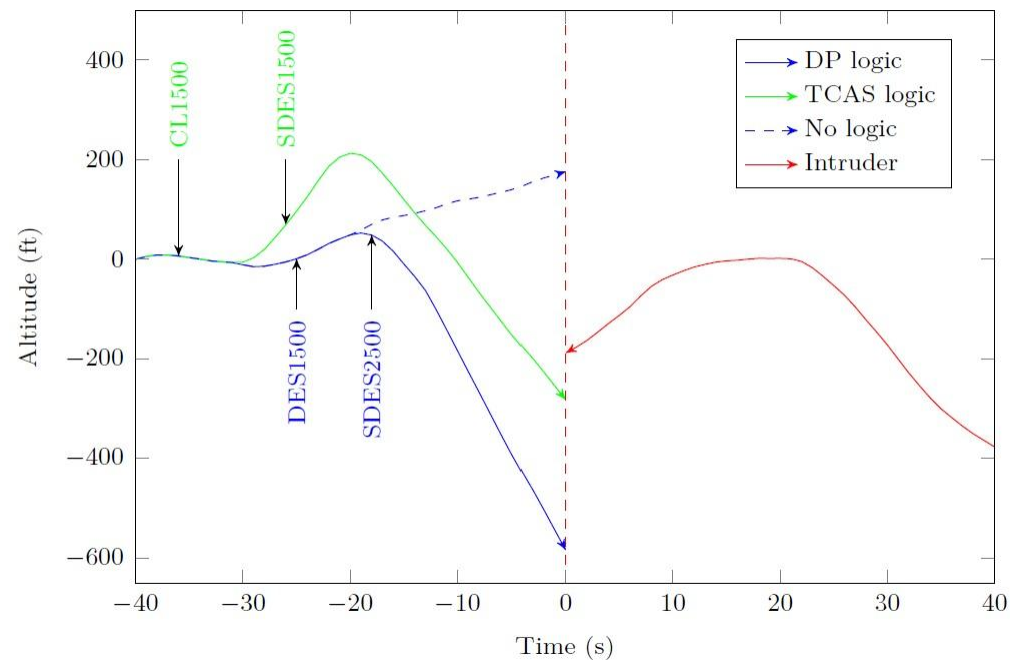


Table 4. Performance evaluation of 1 million head-on encounters

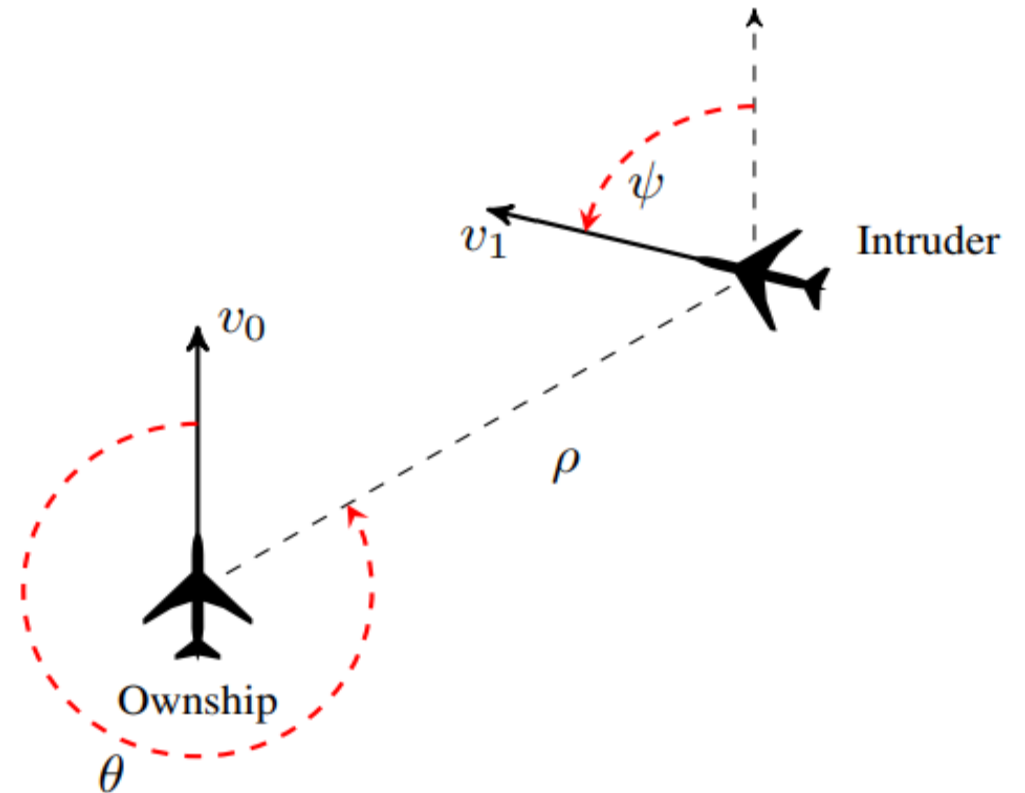
Metric	DP Logic	TCAS Logic
NMACs	3	169
Alerts	690,406	994,317
Strengthenings	92,946	40,470
Reversals	9569	197,315

# Horizontal Policy Table Development

Table 5. Horizontal Logic State Vector Descriptions and Discretisation

Variable	Description	Discretisation
$r$	Range to Intruder	0, 1000,...40000 ft
$\theta$	Relative intruder Bearing	0, 15,...,345 deg
$\varphi$	Relative Intruder Heading	0, 15,...,345 deg
$v_0$	Own-ship Ground Speed	100, 200, 300, 400
$v_1$	Intruder Ground Speed	100, 200, 300, 400
$\tau$	Time to loss of <b>vertical</b> separation	0, 1,..., 40 s
$a_{prev}$	Previously Issued Advisory	N/A (already discrete)

Figure 15. Geometry for ACAS Xu Horizontal Logic Table



Horizontal States based on Owen & Kochenderfer (2016)

# Horizontal Policy Table Development

Table 6. Horizontal Advisory Sets

Action	Description
Clear of Conflict	No manoeuvring necessary
Turn Right at least 3 °/s	Avoid NMAC
Turn Left at least 3 °/s	Avoid NMAC
Turn Right at least 1.5 °/s	Avoid a separation Violation
Turn Left at least 1.5 °/s	Avoid a separation Violation

Table 7. Horizontal Costs

NMAC	Separation	Alert	Strengthen	Reversal	Weaken	COC
100	5	0.0125	0.15	0.3	0.21	0.0

Horizontal Advisories based on Owen & Kochenderfer (2016)  
Costs roughly based on Corteguera (2020)

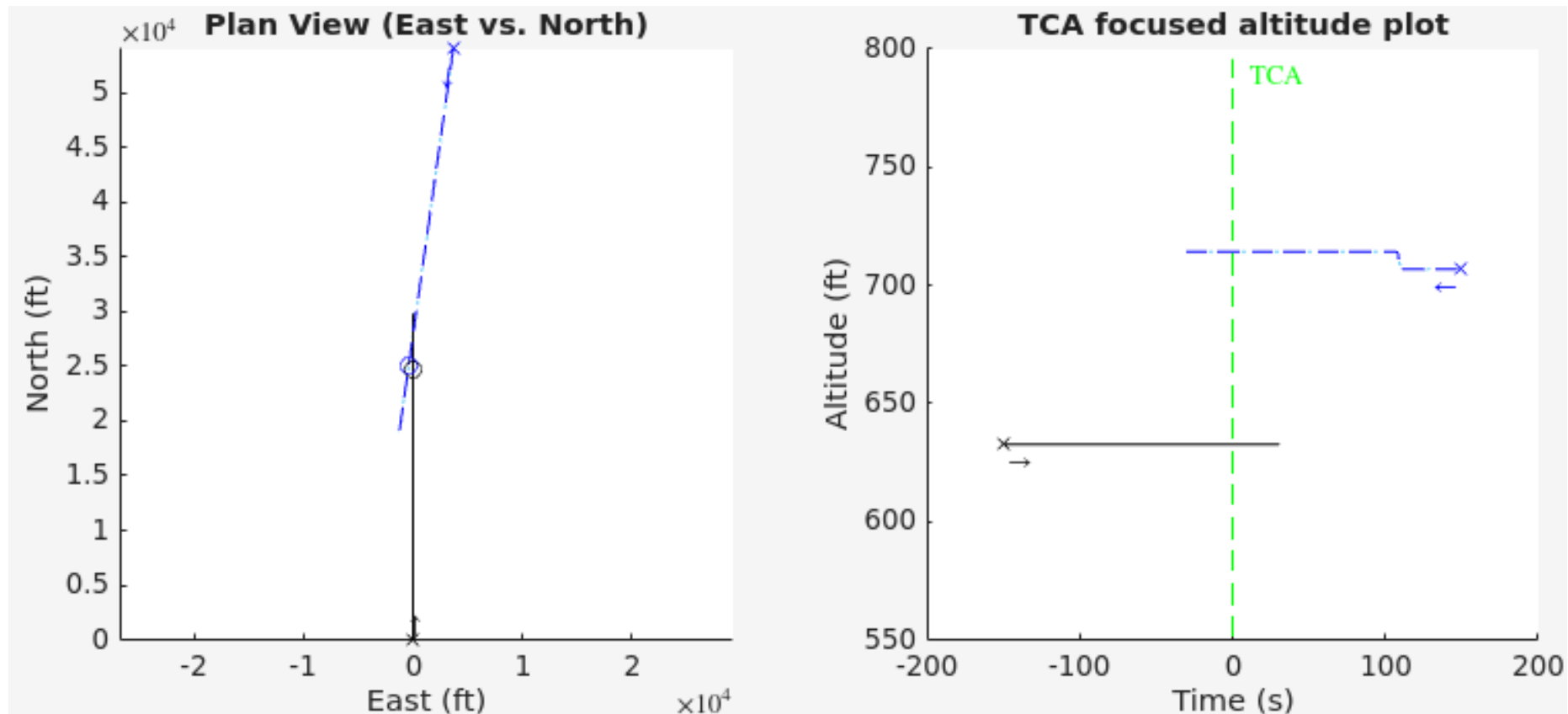
# Preliminary Results and Analysis

1. Example Encounter
2. Metrics for Comparison
3. 10 Encounter Monte Carlo Simulation



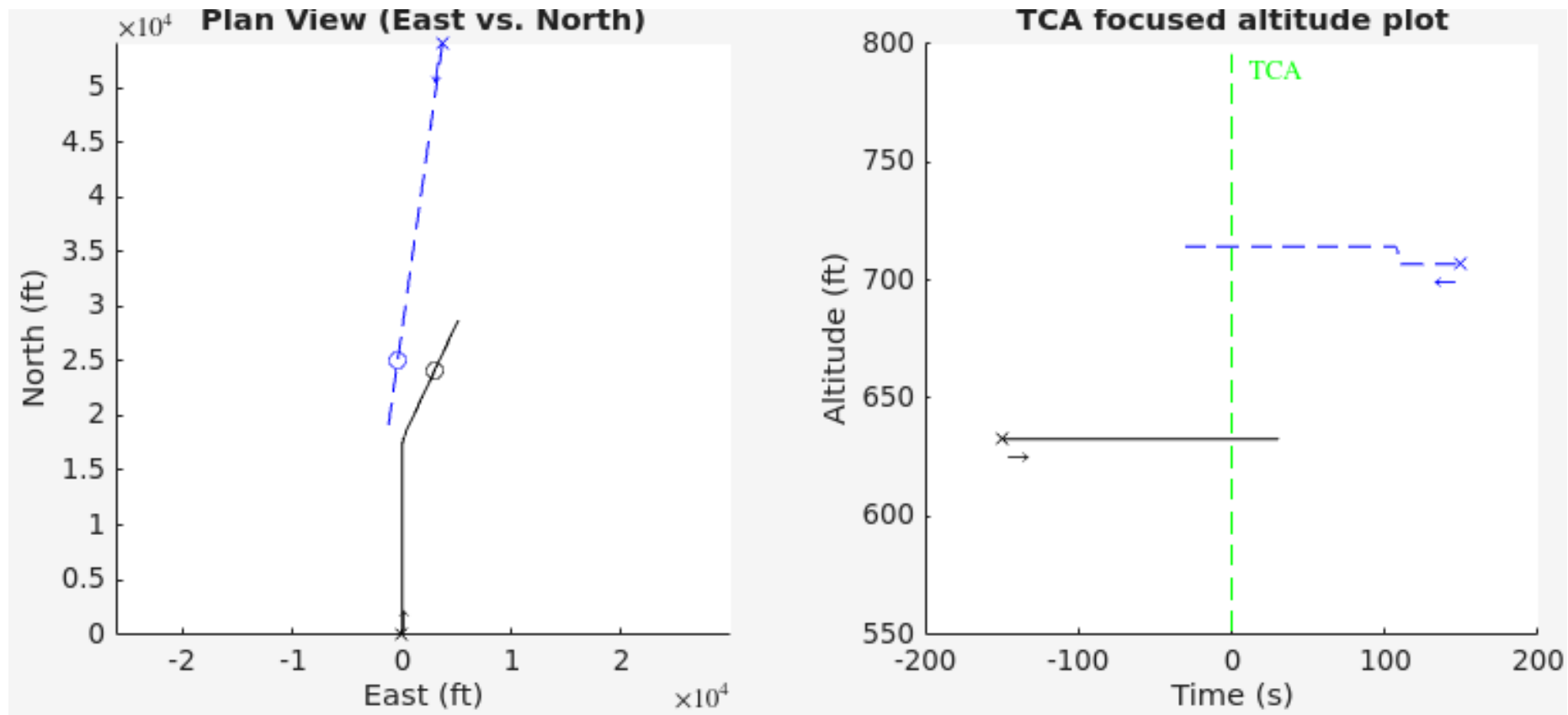
# Example NMAC Encounter 1: Nominal Advisory

VMD = 82 ft, HMD = 489 ft, NMAC = 1



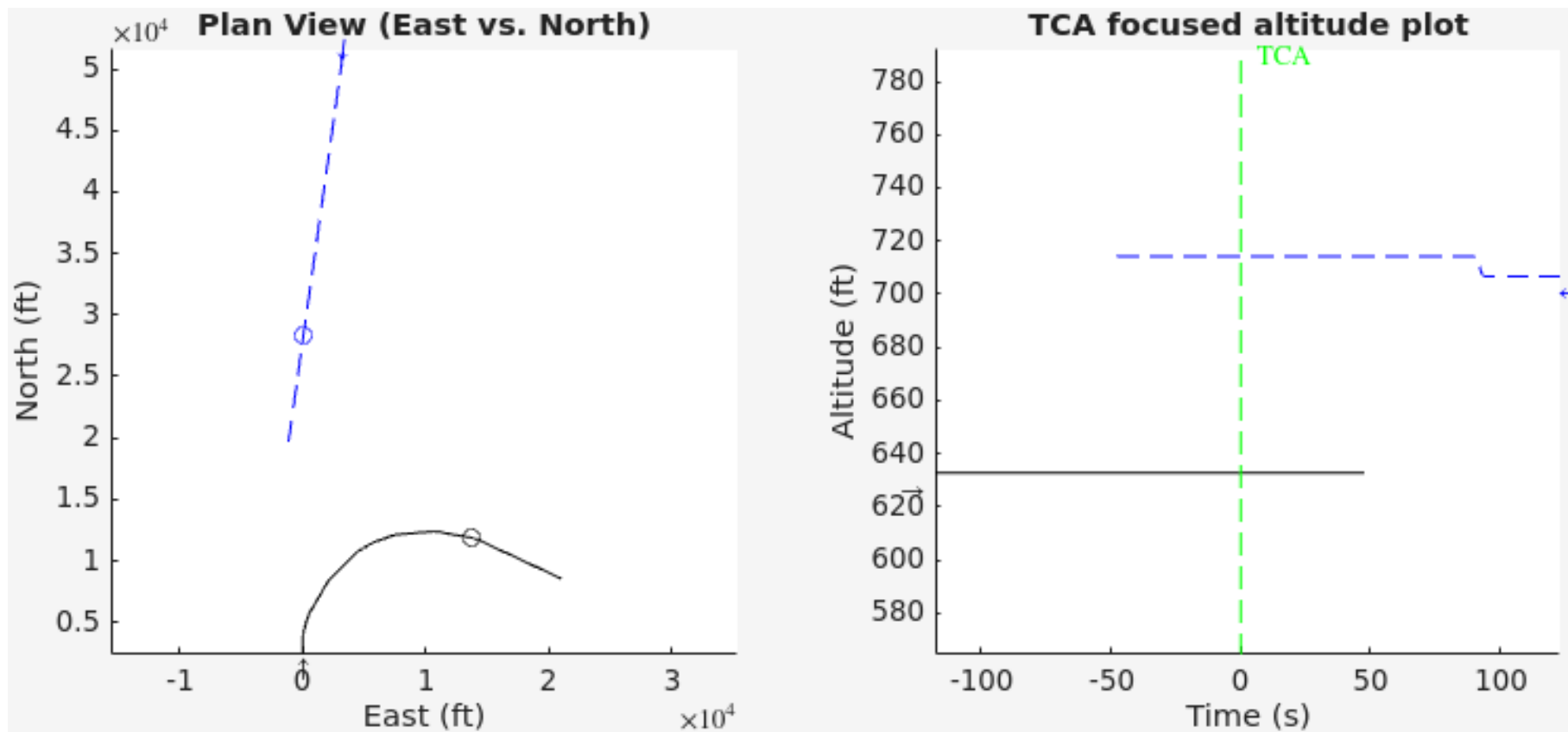
# Example NMAC Encounter 1: DAIDALUS Advisory

VMD = 82 ft, HMD = 3487 ft, NMAC = 0, WCV = 0



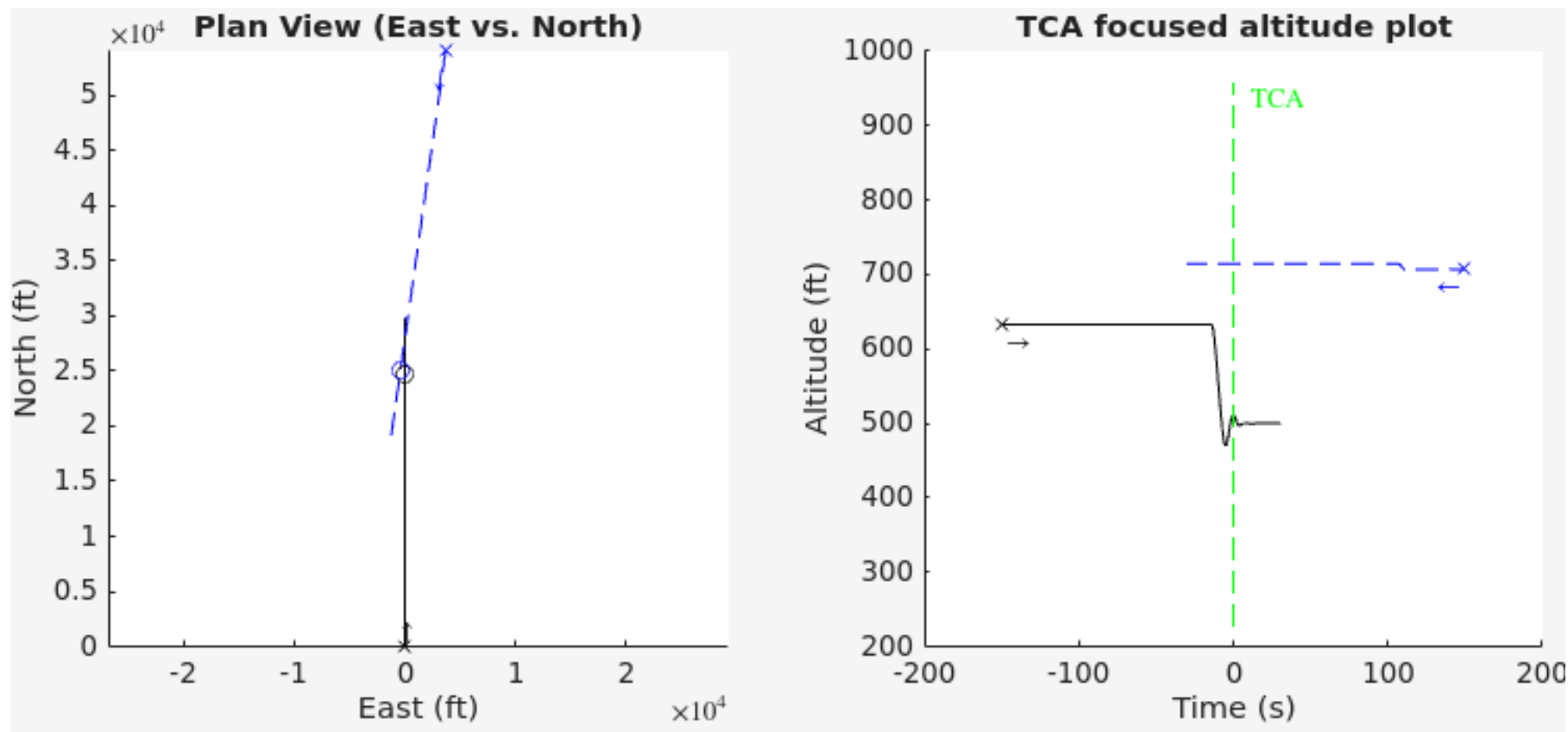
# Example NMAC Encounter 1: ACAS Horizontal Advisory

VMD = 82 ft, HMD = 21452 ft, NMAC = 0, WCV = 0



# Example NMAC Encounter 1: ACAS Vertical Advisory

VMD = 204 ft, HMD = 496 ft, NMAC = 0, WCV = 1



# Metrics of Comparison

Table 8. Metric Definitions

Metric	HMD (ft)	VMD (ft)	Colour
Well Clear (WC)	$\geq 2201$	$\geq 451$	Green
Well Clear Violation (WCV)	$501 \leq ft \leq 2200$	$101 \leq ft \leq 450$	Yellow
Near Mid Air Collision (NMAC)	$\leq 500$	$\leq 100$	Red

- Well clear threshold: SC-228 / DAA non-cooperative well-clear definition used in DEGAS
- NMAC threshold: Standard UAS encounter metric used in the simulation summaries

## Notes:

- Only counts as NMAC if both safety thresholds are not met (same for WCV)
- When running the larger scale simulations  $P_{NMAC}$  and *Risk Ratio* will be used however for 1 to 10 runs, this is unnecessary.

# Small Monte Carlo Simulation

## Small Scale Simulation using 10 NMAC Encounters

Table 9. Small Monte Carlo Simulation Results

ENCOUNTER INFO			NOMINAL		DAIDALUS		HORIZONTAL		VERTICAL	
EncNum	initialRange_ft	initialVerticalSep_ft	hmd_ft	vmd_ft	hmd_ft	vmd_ft	hmd_ft	vmd_ft	hmd_ft	vmd_ft
1	54143	74	419	74	3487	82	21452			211
2	60682	2179	374	1	3472	46	499			475
3	26478	90	409	90	3558	90	5565			447
4	36600	93	17	81	1076	83	1975			166
5	48607	54	421	54	3392	54	20236			441
6	59497	85	123	85	3334	90	9702			452
7	42033	2133	201	0	3278	734	502			469
8	57084	53	309	53	3482	62	22188			2478
9	28712	84	416	84	3310	84	8984			302
10	40051	99	147	79	3310	88	22440			2642

Nominal: 10 WCV's, 10 NMAC's

DAIDLUS: 0 WCV's, 0 NMAC's

ACAS-Xu: 1 WCV, 0 NMAC's

# Project Progress & Next Steps

1. Timeline
2. Plans for Remaining Tasks
3. Questions





# Plans for Remaining Tasks

## Dynamic Logic Selection for Unmanned Aircraft Separation

Michael P. Owen\* and Mykel J. Kochenderfer†  
\*Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, MA, 02420  
†Department of Aeronautics and Astronautics, Stanford University, Stanford, CA, 94305

**Abstract**—The ACAS Xu program has developed both a horizontal and vertical collision avoidance logic to enable unmanned aircraft to avoid manned aircraft. Each logic supports a variety of surveillance sources and is individually capable of providing a significant safety benefit. This paper proposes a logic selection function that can effectively arbitrate between the horizontal and vertical logics to achieve an overall safety benefit without requiring the use of simultaneous horizontal and vertical maneuvers. Monte Carlo simulations show that the logic selection function can provide a performance benefit for a variety of surveillance sources.

### I. INTRODUCTION

Aircraft collision avoidance systems based on Markov decision processes (MDPs) have been developed and demonstrated for both manned and unmanned aircraft [1–3]. An advantage of an MDP approach is that the models explicitly account for uncertainty inherent in real-world problems due to imperfect observations and stochastic outcomes. Due to the complexity of the airborne collision avoidance problem when posed as a Markov decision process, research has largely focused on either a horizontal or vertical set of actions, but not both. With either approach, the number of dimensions requires a very large number of states to adequately represent the collision avoidance problem. Current collision avoidance logics based on MDPs have made approximations to make the solutions computationally tractable and reduce the memory requirements [4]. While useful and effective, these approximations limit these solutions to either a vertical or horizontal action set.

The ACAS Xu logic being developed by the FAA as a replacement to the existing TCAS system [5] for manned aircraft uses a vertical action set that directly corresponds to the existing TCAS action set. For unmanned aircraft collision avoidance, ACAS Xu has been demonstrated with both a vertical maneuver-based logic, similar to ACAS Xu, and a horizontal maneuver-based logic designed for aircraft that have limited vertical maneuvering capability [6, 7]. The current ACAS Xu prototype uses separate vertical and horizontal maneuver logics due to the computational and storage challenges in creating a single logic that allows both types of actions.

The horizontal and vertical collision avoidance logics developed for ACAS Xu both attempt to solve the same problem, that of avoiding near mid-air collisions (NMACs), which occurs when aircraft come within 100ft horizontally and 500ft vertically. However, the two logics are affected differently by different sources of uncertainty, including uncertainty in the current state due to sensor error as well as uncertainty

in the future trajectories of the aircraft. The horizontal and vertical logics have somewhat orthogonal susceptibilities to both types of uncertainty, which suggests that intelligently selecting which logic to use can improve performance. This paper proposes a method for selecting between the horizontal and vertical logics to exploit the specific advantages of a particular logic in the given situation.

This paper reviews both the horizontal logic development and the logic selection process. The intelligent logic selection criteria is outlined and evaluated. Experiments show that system risk can be significantly reduced compared to the performance of a single logic, with minimal increases in overall alert rate. Performance benefits are assessed for ADS-B, radar, and active interrogation surveillance sources in a Monte Carlo simulation framework. This paper is organized as follows. Section II discusses the solution of the collision avoidance problem using an MDP and the resulting performance. Section III motivates and reviews the logic selection capability. Section IV reviews system performance results when using the logic selection function, and Section V concludes.

### II. PROBLEM FORMULATION

As with ACAS Xu [1], ACAS Xu formulates the collision avoidance problem as an MDP. While the principles remain the same, the implementations are different for the vertical and horizontal elements of the ACAS Xu collision avoidance logic. Both are discussed in this section after a general review of MDPs.

#### A. Markov Decision Processes

Associated with an MDP is a set of states and a set of actions. The probability of transitioning from a state  $s$  to a state  $s'$  is given by  $T(s', s, a)$ . The reward for transitioning from state  $s$  to  $s'$  by action  $a$  is given by  $R(s, a, s')$ . The objective in an MDP is to choose the best possible action from the current state in order to maximally accumulate reward over time. Solutions can be found by discretizing the state space and applying dynamic programming [8].

A partially-observable Markov decision process (POMDP) generalizes the concept of an MDP to include uncertainty in the current state. Instead of observing the current state exactly, only imperfect observations are obtained. The probability of observing  $o$  after taking action  $a$  and transitioning to state  $s'$  is given by  $O(o | s', a)$ . Bayes' rule can be used to infer a belief distribution  $b$  over the current state from the past sequence of

## Large Scale Monte Carlo Simulation

By Week 15

### Policy Compression for Aircraft Collision Avoidance Systems

Kyle G. Jahnke\*, Dennis Lippert†, Jeffrey S. Brudak†, Michael P. Owen\* and Mykel J. Kochenderfer\*  
\*Department of Aeronautics and Astronautics, Stanford University, Stanford, CA, 94305  
†United Research Laboratories, Boston Institute of Technology, Boston, MA, 02127  
†Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, MA, 02420

**Abstract**—This approach to developing the decision-making logic for an aircraft collision avoidance system is based on the problem of policy compression. The resulting logic is more compact and easier to implement than the original logic. This approach has been used in the development of the ACAS Xu logic for collision avoidance systems. The ACAS Xu logic is a large Markov decision process (MDP) that is used to generate the logic for the ACAS Xu system. The ACAS Xu logic is a large MDP that is used to generate the logic for the ACAS Xu system. The ACAS Xu logic is a large MDP that is used to generate the logic for the ACAS Xu system.

**I. INTRODUCTION**  
Results of research have explained a variety of different approaches to designing the decision-making logic for an aircraft collision avoidance system. In this paper, we present a new approach to designing the decision-making logic for an aircraft collision avoidance system. This approach is based on the problem of policy compression. The resulting logic is more compact and easier to implement than the original logic. This approach has been used in the development of the ACAS Xu logic for collision avoidance systems. The ACAS Xu logic is a large Markov decision process (MDP) that is used to generate the logic for the ACAS Xu system.

is currently large, requiring hundreds of gigabytes of floating point storage. A single technique to reduce the size of the ACAS Xu logic is to compress the logic into a smaller program size. In addition, the ACAS Xu logic is a large MDP that is used to generate the logic for the ACAS Xu system. The ACAS Xu logic is a large MDP that is used to generate the logic for the ACAS Xu system.

This paper explains one approach for compressing the ACAS Xu logic. The ACAS Xu logic is a large MDP that is used to generate the logic for the ACAS Xu system. The ACAS Xu logic is a large MDP that is used to generate the logic for the ACAS Xu system.

By Week 18

## Define Vertical / Horizontal Logic

By Week 13

## Integration of Policy Table in RTC ACAS-Xu Code

By Week 21

## Completion of Project

Traffic Alert & Collision Avoidance System (TCAS) Program Office

### Algorithm Design Description of the Airborne Collision Avoidance System sXu

ACAS\_ADS\_21\_001\_V3R0a, V3R0



February 19, 2021

Prepared By: Joel Wike Date: 2/19/2021  
                  Joel Wike, sXu STM Lead  
Prepared By: Luis Alvarez Date: 02/19/2021  
                  Luis Alvarez, sXu TRM Lead  
                  NEAL A SUCHY  
Approved By: Neal Suchy Date: \_\_\_\_\_  
                  Neal Suchy, Program Manager, AJM-42  
                  FAA TCAS Program Office

# Any Questions?

Flynn Devoy | Masters Student

School of Mechanical Engineering

f.devoy@uq.edu.au



**Figure 16: A picture I took while we were flight testing the current Detect and Avoid system that Revolution Aerospace have developed. This allowed me to see how ACAS-Xu could be utilised in the real world.**

# Feedback from Thien-Ming (UQ Supervisor)

## Pros:

- + The student has a clear vision of the project's motivation and objective, which seeks to develop an MDP-based alert and guidance system following the ACAS-Xu framework in the literature.
- + They have thoroughly reviewed the existing work and literature, showing an intimate level of understanding of some key pieces.
- + The presentation is well structured and shows a deliberate attempt to analyse and review each sub-system.
- + The project is well planned and is progressing nicely towards the objective.
- + There are good dynamics between the student and the industry supervisor.

## Suggestions for improvement:

+ Overall, the work is nicely executed. I would just suggest preparing some quick-selling numbers to engage the viewers (e.g., “99.99% NMAC reduction”).