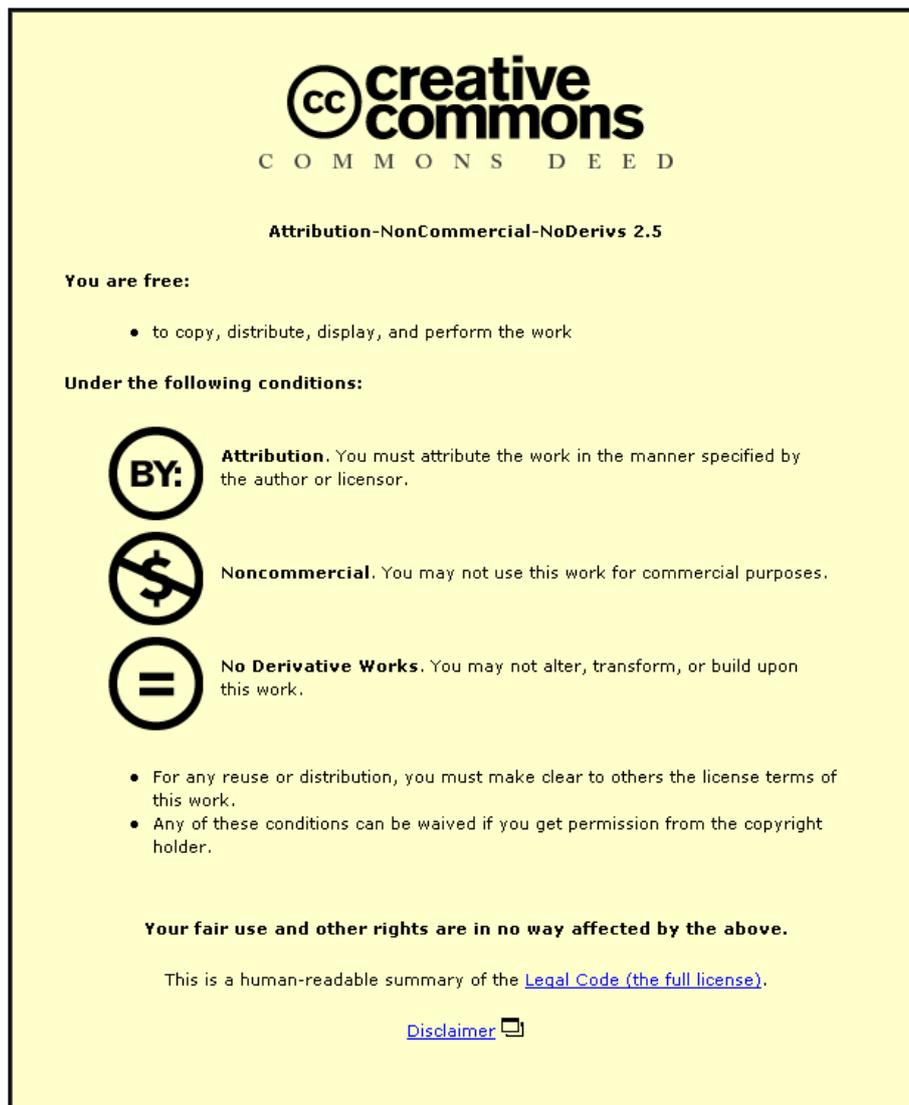


This item was submitted to Loughborough University as a PhD thesis by the author and is made available in the Institutional Repository (<https://dspace.lboro.ac.uk/>) under the following Creative Commons Licence conditions.



For the full text of this licence, please go to:  
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

# Autonomous Terminal Area Operations for Unmanned Aerial Systems



Owen McAree

Department of Aeronautical and Automotive Engineering

Loughborough University

A thesis submitted for the degree of

*Doctor of Philosophy*

June 2013

I would like to dedicate this thesis to my loving parents ...

## Acknowledgements

I would like to thank Professor Wen-Hua Chen for his guidance and support during the course of this work, and before that as a lecturer and project supervisor. I have the greatest appreciation for all his efforts, not only in support of myself, but in building up a highly successful research group which has inspired myself and others to pursue a research career.

I would also like to thank Charles Patchett for his guidance during the project. His aviation background has provided a wealth of information to steer this project, ensuring it has great relevance to industry.

I am grateful for the help and support I have received from all the members of the Loughborough University Centre for Autonomous Systems throughout the course of my research. Their knowledge and enthusiasm has been a constant source of inspiration and motivation.

My thanks go to BAE Systems who provided the funding for my work through an EPSRC CASE studentship and to Loughborough University and the Department of Aeronautical and Automotive Engineering whose facilities have been invaluable.

Finally, I would like thank my parents for their ongoing love and support.

## Abstract

After many years of successful operation in military domains, Unmanned Aerial Systems (UASs) are generating significant interest amongst civilian operators in sectors such as law enforcement, search and rescue, aerial photography and mapping. To maximise the benefits brought by UASs to sectors such as these, a high level of autonomy is desirable to reduce the need for highly skilled operators. Highly autonomous UASs require a high level of situation awareness in order to make appropriate decisions. This is of particular importance to civilian UASs where transparency and equivalence of operation to current manned aircraft is a requirement, particularly in the terminal area immediately surrounding an airfield.

This thesis presents an artificial situation awareness system for an autonomous UAS capable of comprehending both the current continuous and discrete states of traffic vehicles. This estimate forms the basis of the projection element of situation awareness, predicting the future states of traffic. Projection is subject to a large degree of uncertainty in both continuous state variables and in the execution of intent information by the pilot. Both of these sources of uncertainty are captured to fully quantify the future positions of traffic.

Based upon the projection of future traffic positions a self separation system is designed which allows an UAS to quantify its separation to traffic vehicles up to some future time and manoeuvre appropriately to minimise the potential for conflict. A high fidelity simulation environment has been developed to test the performance of the artificial situation awareness and self separation system. The system has demonstrated good performance under all situations, with an equivalent level of safety to that of a human pilot.

# Contents

<b>Contents</b>	<b>iv</b>
<b>List of Figures</b>	<b>viii</b>
<b>Nomenclature</b>	<b>xiv</b>
<b>List of Acronyms</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Overview . . . . .	1
1.2 Outline . . . . .	4
<b>2 Literature Review</b>	<b>6</b>
2.1 Introduction . . . . .	6
2.2 Autonomy . . . . .	6
2.2.1 Autonomous functions . . . . .	11
2.3 Situation awareness . . . . .	15
2.3.1 Level 1 SA - Perceptual encoding . . . . .	19
2.3.2 Level 2 SA - Comprehension of meaning . . . . .	22
2.3.3 Level 3 SA - Projection of future states . . . . .	23
2.3.4 Response selection . . . . .	24
2.4 Terminal area . . . . .	25
2.4.1 Visual approaches . . . . .	25
2.4.2 Instrument approaches . . . . .	28
2.4.3 Approach summary . . . . .	29
2.4.4 Situation awareness . . . . .	31

2.5	Summary . . . . .	33
<b>3</b>	<b>Comprehension</b>	<b>34</b>
3.1	Circuit definitions . . . . .	35
3.1.1	Circuit Coordinate System (CCS) . . . . .	35
3.1.2	Discrete modes . . . . .	37
3.2	Hybrid state estimation . . . . .	37
3.2.1	Aircraft model . . . . .	39
3.2.2	Flight Change Points . . . . .	44
3.2.3	Discrete mode transition . . . . .	47
3.2.4	State Dependent Transition Hybrid Estimation algorithm .	50
3.3	Results . . . . .	54
3.3.1	Simulation set up . . . . .	54
3.3.2	Test 1 - Nominal flight path . . . . .	56
3.3.3	Test 2 - FCP uncertainty . . . . .	61
3.3.4	Sensitivity study . . . . .	65
3.3.5	Discussion . . . . .	66
3.4	Summary . . . . .	66
<b>4</b>	<b>Projection</b>	<b>68</b>
4.1	Introduction . . . . .	68
4.2	Nominal flight path projection . . . . .	69
4.2.1	Short term prediction . . . . .	69
4.2.2	Long term prediction . . . . .	71
4.3	Projection uncertainty . . . . .	72
4.3.1	Continuous linear uncertainty . . . . .	72
4.3.2	Continuous curvilinear uncertainty . . . . .	75
4.3.3	Discrete uncertainty . . . . .	80
4.3.4	Bounded uncertainty . . . . .	87
4.3.5	Overview . . . . .	92
4.4	Results . . . . .	92
4.4.1	Stochastic traffic model . . . . .	93
4.4.2	Model parameters . . . . .	95

4.4.3	Error calculation . . . . .	95
4.4.4	Test 1 - Known parameters . . . . .	97
4.4.5	Test 2 - Unknown continuous parameters . . . . .	101
4.4.6	Test 3 - Unknown discrete variance . . . . .	103
4.5	Summary . . . . .	103
<b>5</b>	<b>Self Separation</b>	<b>106</b>
5.1	Introduction . . . . .	106
5.2	Self separation metrics . . . . .	107
5.2.1	Distance between vehicles . . . . .	108
5.2.2	Time separation . . . . .	115
5.2.3	Time to Point of Closest Approach (TPCA) . . . . .	117
5.2.4	Distance at TPCA . . . . .	120
5.3	Non-Gaussian projection . . . . .	120
5.3.1	Distance between vehicles . . . . .	120
5.3.2	Time separation . . . . .	124
5.3.3	TPCA and distance at TPCA . . . . .	124
5.4	Decision making . . . . .	124
5.4.1	Circuit constraints . . . . .	125
5.4.2	Self separation rules . . . . .	128
5.4.3	Practical considerations . . . . .	133
5.5	Summary . . . . .	133
<b>6</b>	<b>Test Environment</b>	<b>135</b>
6.1	Introduction . . . . .	135
6.2	Environment overview . . . . .	136
6.3	Autopilot system - ArduPilot Mega (APM) . . . . .	139
6.3.1	Communication . . . . .	139
6.3.2	Autopilot functions . . . . .	141
6.3.3	Prototyping features . . . . .	143
6.4	Micro Air Vehicle Communication Protocol (MAVLink) . . . . .	143
6.5	High fidelity flight model - X-Plane . . . . .	144
6.5.1	Software Development Kit (SDK) . . . . .	146

6.5.2	Multiplayer vehicles . . . . .	148
6.5.3	Communication . . . . .	148
6.6	MATLAB/Simulink interface . . . . .	149
6.7	Real world environment . . . . .	149
6.7.1	Multi vehicle and mixed environment testing . . . . .	151
6.8	Results . . . . .	152
6.8.1	Speed hold test . . . . .	152
6.8.2	Heading hold test . . . . .	153
6.8.3	Mixed mode multiple vehicle test . . . . .	154
6.9	Conclusion . . . . .	156
<b>7</b>	<b>Experimental Results</b>	<b>157</b>
7.1	Introduction . . . . .	157
7.2	Experimental set up . . . . .	158
7.3	Test scenarios . . . . .	158
7.4	Software In the Loop (SIL) Results . . . . .	164
7.4.1	Typical trajectories . . . . .	164
7.4.2	General trends . . . . .	166
7.5	Conclusion . . . . .	167
<b>8</b>	<b>Conclusions</b>	<b>169</b>
8.1	Summary . . . . .	169
8.2	Research challenges . . . . .	171
8.2.1	Multiple model projection . . . . .	171
8.2.2	Constrained projection . . . . .	172
8.2.3	Computational complexity . . . . .	172
8.2.4	Decision making fidelity . . . . .	173
8.2.5	Communications integrity . . . . .	173
<b>Appendix A Derivation of haversine formulae</b>		<b>174</b>
A.1	Distance between two points on Earth . . . . .	174
A.2	Initial heading between two points on Earth . . . . .	176
<b>References</b>		<b>179</b>

# List of Figures

1.1	An overview of the system presented in this thesis . . . . .	4
2.1	Three levels of Flight Control System (FCS) . . . . .	8
2.2	Reducing operator bandwidth . . . . .	9
2.3	Endsley’s model of situation awareness in dynamic decision making	16
2.4	Endsley’s model of situation awareness for a highly automated system	17
2.5	Human and Unmanned Aerial System (UAS) situation awareness	19
2.6	Adams model of situation awareness . . . . .	20
2.7	Standard overhead join . . . . .	27
2.8	Procedural approach example . . . . .	30
3.1	Coordinate System Transformation . . . . .	36
3.2	Discrete modes within a circuit pattern . . . . .	38
3.3	Discretisation of a continuous flight path . . . . .	41
3.4	Sub-mode transition Markov diagrams . . . . .	44
3.5	Single variable cumulative distribution function for mode transition	49
3.6	Execution logic of the State Dependent Transition Hybrid Estima- tion (SDTHE) algorithm . . . . .	52
3.7	Typical circuit dimensions used as nominal flight path . . . . .	55
3.8	Plot of the estimated flight path without Flight Change Point (FCP) uncertainty . . . . .	58
3.9	Plot of the discrete mode estimation without FCP uncertainty . .	59
3.10	Plot of the errors in the estimated continuous states without FCP uncertainty . . . . .	60
3.11	Plot of the estimated flight path with FCP uncertainty . . . . .	62

## LIST OF FIGURES

---

3.12	Plot of the discrete mode estimation with FCP uncertainty . . . . .	63
3.13	Plot of the errors in the estimated continuous states with FCP uncertainty . . . . .	64
4.1	Piecewise integration of continuous uncertainty . . . . .	74
4.2	Linearisation error during curved flight . . . . .	76
4.3	Curvilinear projection illustration . . . . .	77
4.4	Curvilinear coordinate system . . . . .	78
4.5	Transformation from curvilinear to CCS coordinates . . . . .	79
4.6	Discrete uncertainty in flight path projection . . . . .	81
4.7	Effect of discrete transition variance on projection . . . . .	82
4.8	Effect of crosstrack variance on projection . . . . .	83
4.9	Comparison of curvilinear projection with and without discrete uncertainty vs Monte Carlo simulation . . . . .	86
4.10	Curvilinear coordinates with bounded uncertainty . . . . .	87
4.11	The result of coordinate constriction . . . . .	88
4.12	The result of coordinate constriction for projection times close to Estimated Time of Arrival (ETA) . . . . .	89
4.13	The result of variance reduction . . . . .	90
4.14	The result of variance reduction for projection times close to ETA	91
4.15	Diagram of the Curvilinear Projection with Discrete Uncertainty (CPDU) algorithm . . . . .	93
4.16	RMS error vs Projection time for first test . . . . .	98
4.17	Worst case parameter combination - projection distributions . . . . .	99
4.18	Worst case parameter combination - marginal distributions . . . . .	100
4.19	Worst and best case error comparison . . . . .	102
4.20	Sensitivity of projection to discrete uncertainty parameter variation	104
5.1	Distance between and known and uncertain point . . . . .	108
5.2	Distance between and known and uncertain point, coordinate sys- tem transformed . . . . .	109
5.3	Comparison of simulated and analytical radial distribution . . . . .	112
5.4	Comparison of analytical and approximate radial distribution . . . . .	113
5.5	Illustration of large variance relative to mean . . . . .	114

## LIST OF FIGURES

---

5.6	Comparison of analytical and approximate radial distribution with large mean relative to variance . . . . .	114
5.7	Comparison of analytical and approximate temporal distribution with comparative mean relative to variance . . . . .	116
5.8	Comparison of analytical and approximate radial distribution with large mean relative to variance . . . . .	117
5.9	TPCA example . . . . .	118
5.10	Cumulative distribution plot of distance for converging aircraft . . . . .	119
5.11	Positions of UAS relative to traffic . . . . .	122
5.12	Comparison of numerical Cumulative Distribution Function (CDF) against Monte Carlo simulation (1000 executions) . . . . .	123
5.13	Typical circuit joins . . . . .	126
5.14	Typical circuit joins excluding the overhead . . . . .	127
5.15	Allowable circuit variations . . . . .	129
5.16	Decision making process for a UAS outside of the circuit. T: Traffic, U: UAS . . . . .	130
5.17	Example of UAS joining the circuit . . . . .	131
5.18	Decision making process for a UAS inside the circuit. T: Traffic, U: UAS . . . . .	132
6.1	Overview of the systems presented in this thesis within the test environment context . . . . .	136
6.2	X-Plane, APM, Simulink integration . . . . .	138
6.3	The APM autopilot . . . . .	140
6.4	The APM ground station . . . . .	140
6.5	APM auto mode execution logic . . . . .	142
6.6	MAVLink packet structure . . . . .	144
6.7	Custom Simulink block for MAVLink communication . . . . .	149
6.8	Ripmax Wot4 Foam-E with APM installed . . . . .	150
6.9	X-Plane model of the Wot4 airframe . . . . .	151
6.10	Real world interface between Simulink and APM . . . . .	152
6.11	Speed hold test for SIL, Hardware In the Loop (HIL) and real world . . . . .	153
6.12	Heading hold test for SIL, HIL and real world . . . . .	154

## LIST OF FIGURES

---

6.13	Mixed mode (SIL and real) formation flight . . . . .	155
7.1	APM SIL build executing in a VirtualBox virtual machine . . . . .	159
7.2	Network router to provide Transmission Control Protocol (TCP) link for VirtualBox . . . . .	160
7.3	Diagram of the SIL and real world test scenario . . . . .	162
7.4	Alternate routes for UAS from hold to circuit . . . . .	163
7.5	SIL results - Test 1 . . . . .	164
7.6	SIL results - Test 2 . . . . .	165
7.7	SIL results - Test 3 . . . . .	165
7.8	Minimum separation distance between UAS and traffic over 100 simulations . . . . .	167
A.1	Calculation of initial bearing from normal vectors . . . . .	178

# Nomenclature

## Roman Symbols

$\mathcal{N}$	Gaussian (Normal) distribution
$a$	Altitude
$C$	Guard condition
$c$	Crosswind axis
$d$	Distance
$dt$	Time step increment
$h$	Height
$k$	Time step
$L$	Coefficient matrix
$M$	Discrete mode history
$m$	Discrete mode
$p$	Probability
$Q$	Process noise covariance
$R$	Radius of the earth, $6378km$
$SA$	Situation awareness

## NOMENCLATURE

---

$t$	Prediction time
$u$	Upwind axis
$v$	Velocity
$x$	Continuous vehicle state
$Z$	Measurement history
$z$	Measurement

### Greek Symbols

$\alpha$	Mode probability
$\mu$	Mean
$\nu$	Measurement noise
$\omega$	Process noise
$\Phi$	Gaussian (Normal) cumulative distribution
$\phi, \lambda$	Latitude, Longitude
$\pi$	Mode transition probability
$\psi'$	Heading relative to runway direction
$\psi$	Heading
$\Sigma$	Covariance matrix
$\sigma$	Standard deviation
$\theta^*$	Flight Change Point state

### Subscripts

$CL$	Curvilinear
$d$	Distance

## NOMENCLATURE

---

$e$	Crosstrack error
$H$	Human
$i, j$	Index
$l$	Lateral
$r$	Runway
$S$	System
$t$	Traffic
$V$	Vehicle
$v$	Vertical

# List of Acronyms

<b>AAL</b>	Above Aerodrome Level
<b>ADS-B</b>	Automated Dependant Surveillance - Broadcast
<b>AFIS</b>	Aerodrome Flight Information Service
<b>AHRS</b>	Attitude Heading Reference System
<b>AI</b>	Artificial Intelligence
<b>AIP</b>	Aeronautical Information Publication
<b>AIS</b>	Aeronautical Information Service
<b>AMSL</b>	Above Mean Sea Level
<b>ANO</b>	Air Navigation Order
<b>API</b>	Application Program Interface
<b>APM</b>	ArduPilot Mega
<b>ASCII</b>	American Standard Code for Information Interchange
<b>ATC</b>	Air Traffic Control
<b>ATR</b>	Automated Target Recognition
<b>ATZ</b>	Aerodrome Traffic Zone
<b>BRS</b>	Ballistic Recovery Systems

## LIST OF ACRONYMS

---

<b>BVPDF</b>	Bivariate Probability Density Function
<b>CA</b>	Constant Acceleration
<b>CAA</b>	Civil Aviation Authority
<b>CAS</b>	Controlled Airspace
<b>CCS</b>	Circuit Coordinate System
<b>CD</b>	Constant Descent
<b>CDA</b>	Continuous Descent Approach
<b>CDF</b>	Cumulative Distribution Function
<b>CFH</b>	Carefree Handling
<b>CH</b>	Constant Height
<b>COTS</b>	Commercial Off The Shelf
<b>CPDU</b>	Curvilinear Projection with Discrete Uncertainty
<b>CT</b>	Constant Turn
<b>CTS</b>	Cumulative Temporal Separation
<b>CV</b>	Constant Velocity
<b>DLL</b>	Dynamically Linked Library
<b>DPCA</b>	Distance at Point of Closest Approach
<b>EPO</b>	Expanded Polyolefin
<b>ETA</b>	Estimated Time of Arrival
<b>FAA</b>	Federal Aviation Administration
<b>FBW</b>	Fly By Wire
<b>FCI</b>	Flight Control Interface

## LIST OF ACRONYMS

---

<b>FCP</b>	Flight Change Point
<b>FCS</b>	Flight Control System
<b>GA</b>	General Aviation
<b>GBSAA</b>	Ground Based Sense And Avoid
<b>GCS</b>	Global Coordinate System
<b>GNSS</b>	Global Navigation Satellite System
<b>GPS</b>	Global Positioning System
<b>GPS</b>	Global Positioning System
<b>GUI</b>	Graphical User Interface
<b>HIL</b>	Hardware In the Loop
<b>HUD</b>	Heads Up Display
<b>ICAO</b>	International Civil Aviation Organization
<b>IEEE</b>	Institute of Electrical and Electronics Engineers
<b>IFR</b>	Instrument Flight Rules
<b>ILS</b>	Instrument Landing System
<b>IMC</b>	Instrument Meteorological Conditions
<b>IMM</b>	Interacting Multiple Model
<b>IP</b>	Internet Protocol
<b>IQR</b>	Inter-Quartile Range
<b>ISR</b>	Intelligence, Surveillance and Reconnaissance
<b>JVM</b>	Java Virtual Machine
<b>KR</b>	Knowledge Representation

## LIST OF ACRONYMS

---

<b>LiPo</b>	Lithium Polymer
<b>MAV</b>	Micro Air Vehicle
<b>MAVLink</b>	Micro Air Vehicle Communication Protocol
<b>MEMS</b>	Micro Electromechanical Sensor
<b>MVCDF</b>	Multivariate Cumulative Distribution Function
<b>MVPDF</b>	Multivariate Probability Density Function
<b>NAS</b>	National Airspace System
<b>NATS</b>	National Air Traffic Services
<b>OHJ</b>	Overhead Join
<b>OODA</b>	Observe, Orient, Decide, Act
<b>OOP</b>	Object-oriented Programming
<b>PDF</b>	Probability Density Function
<b>PDR</b>	Perceive, Decide, React
<b>PID</b>	Proportional, Integral, Derivative
<b>PMF</b>	Probability Mass Function
<b>PNG</b>	Portable Network Graphic
<b>RC</b>	Radio Controlled
<b>RF</b>	Radio Frequency
<b>RMS</b>	Root Mean Square
<b>ROTA</b>	Rules Of The Air
<b>RT</b>	Radiotelephony
<b>SAA</b>	Sense and Avoid

## LIST OF ACRONYMS

---

<b>SAS</b>	Stability Augmentation System
<b>SDK</b>	Software Development Kit
<b>SDTHE</b>	State Dependent Transition Hybrid Estimation
<b>SI</b>	International System of Units
<b>SIL</b>	Software In the Loop
<b>TCA</b>	Terminal Control Area
<b>TCAS</b>	Traffic Collision Avoidance System
<b>TCP</b>	Transmission Control Protocol
<b>TMA</b>	Terminal Manoeuvring Area
<b>TPCA</b>	Time to Point of Closest Approach
<b>TSM</b>	Temporal Separation Metric
<b>TTS</b>	Text To Speech
<b>UAS</b>	Unmanned Aerial System
<b>UDP</b>	User Datagram Protocol
<b>URL</b>	Uniform Resource Locator
<b>USB</b>	Universal Serial Bus
<b>VFR</b>	Visual Flight Rules
<b>VMC</b>	Visual Meteorological Conditions

# Chapter 1

## Introduction

### 1.1 Overview

The operation of Unmanned Aerial Systems (UASs) to date has been largely by military forces wishing to gain a tactical advantage whilst removing human operators from danger. As these systems have matured and their component cost reduced, their use in civilian environments has become increasingly attractive. UASs lend themselves well to scenarios such as law enforcement, search and rescue, power and pipe line inspection, environmental monitoring, aerial photography and mapping, where the use of manned aircraft is costly and requires highly skilled pilots.

For UASs to be operated in civil National Airspace Systems (NASs) they must integrate safely and seamlessly with manned vehicles. The scenarios listed above occur largely in uncontrolled airspace, where aircraft do not require flight plans and Air Traffic Control (ATC) do not provide any separation assurance. To safely integrate an UAS in to uncontrolled airspace it must be capable of maintaining separation from manned vehicles which may be exhibiting complex, non-deterministic, behaviour. An UAS with a low level of autonomy is only able to meet these separation requirements to a certain degree, due to the limited and delayed information available to the remote pilot. In the extreme circumstance of total communications loss it would be impossible for the pilot to maintain separation to other vehicles. To mitigate this problem it is necessary for an UAS to

---

possess a higher level of autonomy with the ability to operate fully autonomously. Highly autonomous UASs have the additional benefit of requiring less skilled operators, further reducing the operating costs compared to manned aircraft.

To enable an UAS to autonomously separate from traffic it must possess a high level of situation awareness, that is an understanding of where traffic vehicles are and how they will behave in the future. With a high level of situation awareness an UAS is able to plan its own route so as to minimise conflicts with traffic. Determining the current position of traffic is aided by technologies such as Automated Dependant Surveillance - Broadcast (ADS-B), however in uncontrolled airspace there is often little information available about their future trajectories (such as a flight plan or ATC instructions) leading to a high degree of uncertainty in future positions. This thesis is focused on reducing the uncertainty in future positions by utilising additional knowledge such as the Rules Of The Air (ROTA) and heuristics governing traffic behaviour.

This thesis identifies the requirement for high situation awareness as being most critical within the terminal area, that is the airspace immediately surrounding an arrival or departure airfield. This is because many aircraft are operating in a small region of airspace, likely on converging courses, and often manoeuvre in quick succession. In free airspace, away from the terminal area, it may be assumed that over a period of a few minutes a traffic aircraft will not manoeuvre, but instead maintain its present velocity. In this scenario, uncertainties in the future position of traffic will accumulate due to inaccuracies in the velocity measurement or changing wind conditions. In the terminal area however, this assumption of linear trajectories does not hold, as manoeuvres are often performed in quick succession. Uncertainty in when a vehicle is intending to manoeuvre is a significant challenge in the attainment of terminal area situation awareness by an UAS.

This thesis presents a complete situation awareness system, with regards to traffic vehicles, for an autonomous UAS operating in the terminal area. Firstly, the levels of situation awareness are identified and the requirements at each level quantified. For the latter two situation awareness levels of comprehension and projection, a number of algorithms are designed to construct an artificial situation awareness system. Comprehension focusses on a novel use of a hybrid estimation

---

technique to provide estimates of both continuous and discrete states, allowing the **UAS** to infer the intent of the traffic based on the **ROTA**. Projection extends this comprehended state by estimating future states of the traffic, allowing the **UAS** to plan a conflict free route to maximise flight safety. The projection algorithm treats the traffic vehicles future position as a Probability Density Function (**PDF**) and estimates both the mean and covariance based on heuristics governing terminal area behaviour. The ability of traffic vehicles to make discrete changes to their flight paths at unknown future times (such as commencing a turn) presents a significant challenge to this projection algorithm. The effect of these discrete uncertainties on future traffic positions has been accounted for by treating the discrete transitions as probabilistic events and using a novel geometric technique to realise their impact.

To provide a demonstration of the utility of this artificial situation awareness system, a rule based decision maker is employed to allow the **UAS** to plan a conflict free path. The metrics which govern these decisions correspond to the separation (either time or distance) between vehicles, this is not immediately available from the statistical representation of position given by the projection system. A number of different methods for calculation of these metrics are derived, dependant on the complexity of the position **PDF**. Some of these require significant assumptions and approximations in order to make them computationally feasible, resulting in a number of novel techniques for quantifying vehicle separation being presented.

Figure 1.1 illustrates an overview of the situation awareness and self separation systems presented in this thesis.

Finally, to test the situation awareness, self separation and decision making systems in the most realistic circumstances, a high fidelity simulation and small scale test environment have been developed. This environment allows any system coded in MATLAB/Simulink to interact with a Commercial Off The Shelf (**COTS**) autopilot system which is controlling either a simulated vehicle or a real world aircraft. Tests of the system in this environment have shown good performance, with equivalent levels of safety to that of a human pilot.

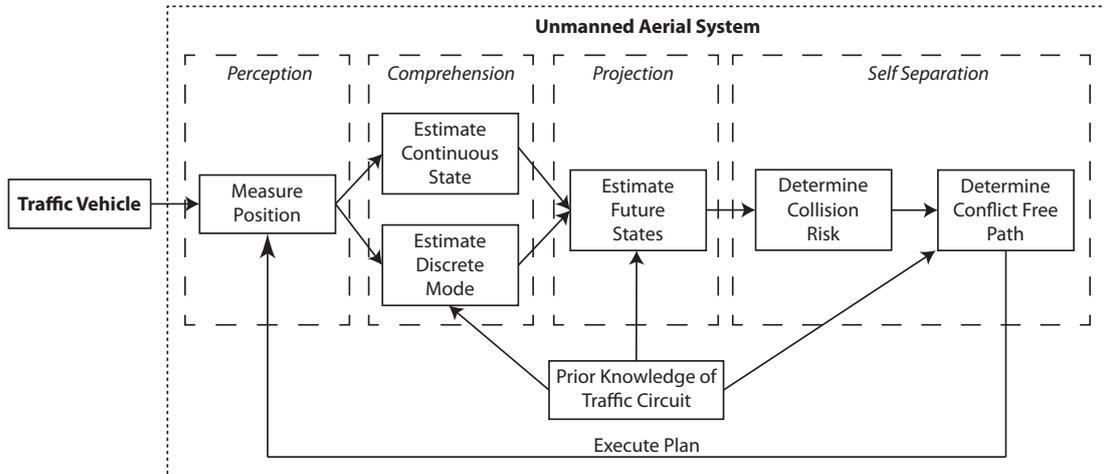


Figure 1.1: An overview of the system presented in this thesis

## 1.2 Outline

This thesis details the development of an artificial situation awareness system for an autonomous [UAS](#) operating in a crowded terminal area. The contents of each chapter are outlined below.

### Chapter 2 - Literature Review

An in depth review of literature associated with autonomy in [UASs](#), situation awareness and terminal area operations. Each subject is dealt with individually, culminating in the situation awareness requirements for an autonomous [UAS](#) operating in the terminal area being defined.

### Chapter 3 - Comprehension

The development of the comprehension element of situation awareness for [UASs](#) in the terminal area is detailed. Two distinct forms of comprehension are required for continuous and discrete states, these are implemented through a hybrid estimation technique.

---

## **Chapter 4 - Projection**

Projection of the future states of traffic vehicles by the [UAS](#) is detailed, this is the final element of situation awareness. The major focus of this chapter is dealing with the inevitable uncertainty which arises from predicting future states.

## **Chapter 5 - Self Separation**

Utilising the artificial situation awareness developed in the previous chapter, a means of the [UAS](#) separating itself from traffic is detailed. This utilises a simple ruled based decision making framework as it is intended as an illustrative example only, not a fully developed system.

## **Chapter 6 - Test Environment**

The virtual and real world test environment developed for this project is described. The integration of [COTS](#) systems is detailed and some initial results presented.

## **Chapter 7 - Experimental Results**

Results are presented for the entire situation awareness and self separation system, implemented in the test environment.

## **Chapter 8 - Conclusions**

The thesis is concluded with future research directions discussed.

# Chapter 2

## Literature Review

### 2.1 Introduction

This chapter presents a review of the literature relevant to the project area, with the discussion broken down into three major areas. Firstly, Section 2.2 considers the autonomy of UASs which highlights the requirement for an artificial situation awareness system. Section 2.3 then reviews this area, drawing particular parallels with human decision making. Finally, Section 2.4 discusses the challenges associated with the terminal area, paying particular attention to the associated problems with autonomy and situation awareness for an UAS.

### 2.2 Autonomy

The push for greater autonomy amongst UASs comes as a natural progression from early vehicles. A major selling point of early UASs was the removal of the human pilot from so-called Dull, Dirty and Dangerous missions. This function has proved highly successful but the push in recent years has been to break the one-to-one relationship between vehicles and operators. Current UASs differ very little from manned aircraft, the only distinction being that the pilot is in a remote, ground based location. Whilst this protects the pilot from any harm which may come to the vehicle, it does not reduce the costs associated with training them. An increase in the autonomy of the vehicle would allow the operator to perform

---

a more supervisory role, a manager rather than a pilot. Ultimately this will allow a single operator to command a number of vehicles in a similar way to the oversight exhibited over civil air transport aircraft by air traffic controllers [Cummings et al., 2010; Department of Defence, 2005]. This would result in a significant saving in manpower and training as well as allowing the vehicles to undertake more complex missions owing to their autonomy.

Operation of UASs in NASs also necessitate high levels of autonomy due to the potential for delayed or interrupted communications. If an UAS with low autonomy is required to respond to an emergency situation, such as a potential collision with another vehicle, but experiences a delay in communicating this information to the pilot a loss of separation or even a collision may occur before the pilot is able to respond. Furthermore, if a total loss of communication is experienced a degree of autonomy is required to safely recover the vehicle.

Developments in the field of low level control algorithms for stability augmentation systems and autopilots allow for even current generation UASs to conduct entire flights without pilot input [Cantwell, 2009]. This ability is aided by recent advances in path planning technologies which allow an autopilot to accept a set of waypoints and autonomously plan the most efficient route between them [Bellingham et al., 2002; Bottasso et al., 2008; Eele and Richards, 2009]. Such path planning methods are also able to incorporate basic collision avoidance strategies provided information about the conflicting object can be adequately perceived [Eele and Richards, 2009]. These systems represent a huge step towards autonomy, however an operator is still required to guide the vehicle (i.e. provide the waypoints). The current level of autonomous functions must be augmented with mission level autonomy before UASs can be considered fully autonomous.

Figure 2.1 illustrates the three tiers of a typical Flight Control System (FCS). Current production UASs utilise fully autonomous stabilisation and control with a degree of autonomy in path planning; mission planning, however is left wholly to the operator [Department of Defence, 2005]. The operator, who may still be termed the pilot, is responsible for all of the tasks attributed to mission planning shown in Figure 2.1. The pilot will receive a mission brief which must be translated into a number of discrete objectives. Many objectives may remain the same for all missions such as ‘Taxi to the runway’, ‘Take-off’, ‘Return to base

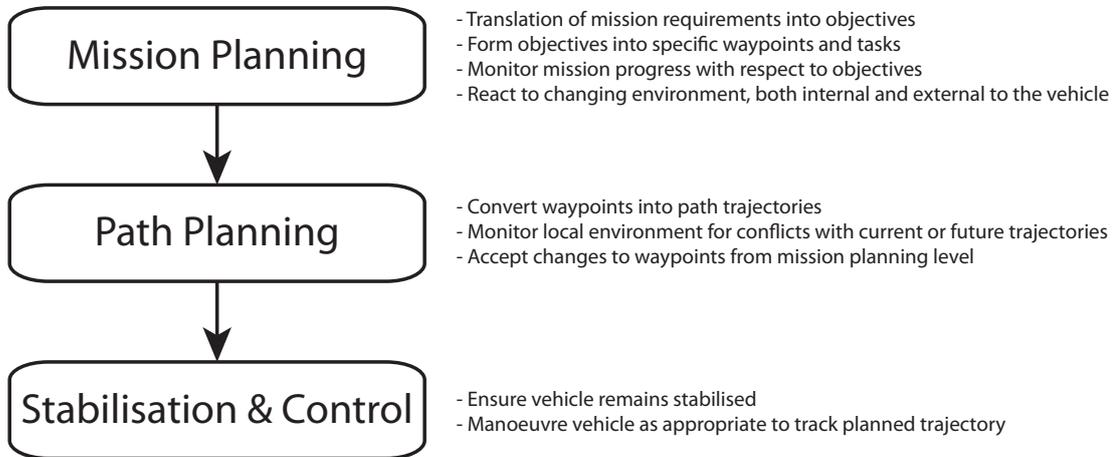


Figure 2.1: Three levels of FCS (modified from [Vachtsevanos et al., 2005])

and land’. Others will be specific to a particular mission such as ‘Follow convoy’ or ‘Monitor for enemy activity’. These objectives are linguistic in nature, that is they are made up from words of the English language. Before the automated path planning system can process the objectives, they must be turned into quantifiable waypoints or tasks. A waypoint is simply a location to which the vehicle must go. Waypoints can be purely spatial or spatiotemporal to represent time constraints. Tasks are jobs which the vehicle must perform without a specific location associated, such as tracking a target.

Breaking down a mission brief into objectives and then into waypoints and tasks represents part of the data flow represented on the left of Figure 2.1. These waypoints/tasks will then be processed by the path planning algorithms which will provide reference trajectories for the stabilisation and control system. Not shown in Figure 2.1 is the flow of data into the mission planning task from external sources. These include weather forecasts, surveillance information, and fault diagnosis systems. All of this data is considered with reference to the mission objectives and appropriate action is taken. Survival of the vehicle may override mission objectives, however if the mission is considered of high enough importance the vehicle may be sacrificed (owing to its unmanned nature).

Automating the processes currently performed by the pilot would greatly reduce the vehicles dependency on said pilot. In such a situation the pilot becomes a manager with reduced bandwidth requirement [Department of Defence, 2007].

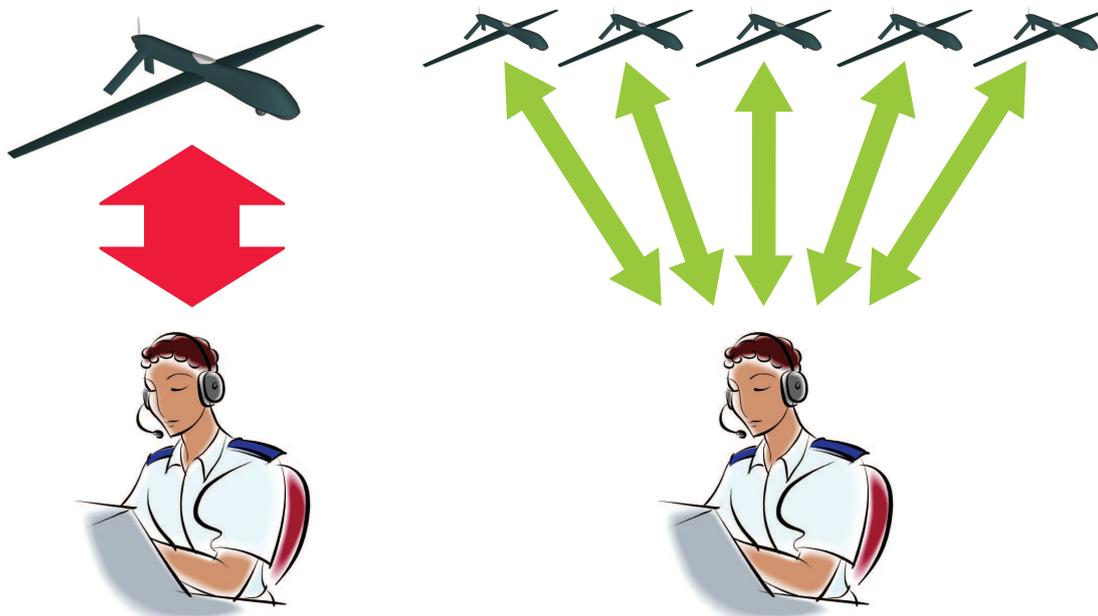


Figure 2.2: Reducing operator bandwidth

The greater the level of vehicle autonomy, the lower the manager bandwidth and consequently the greater the number of vehicles which can be managed, Figure 2.2. Whilst this task for a single vehicle may be complex enough, the next natural progression is for multiple vehicles to become autonomously aware of one another allowing coordination and cooperation without the need for intermediation by the manager.

Current **UASs** include a degree of automation in their lower level tasks therefore they can be termed autonomous systems. However they are not fully autonomous as they have only achieved a level of autonomy. In defining guidelines for the adoption of **UASs** into **NAS** the Civil Aviation Authority (**CAA**) provide a qualitative description of **UAS** Autonomy [**Civil Aviation Authority, 2010b**]. The key points are the ability of an autonomous system to gain situation awareness and to make rational decision based upon this. Both of these attributes are synonymous with some level of Artificial Intelligence (**AI**).

In a similar way to **Ippolito and Pisanich [2005]**, **Civil Aviation Authority [2010b]** generalises the classical Observe, Orient, Decide, Act (**OODA**) military pilot workload loop from **Department of Defence [2007]** into Perceive, Decide,

---

React (PDR). An autonomous UAS is tasked with closing the PDR loop without human interaction. Perception is the task of creating situation awareness from a set of sensory and user inputs<sup>1</sup>. This task is more complex than simply measuring parameters as they must each be put into context. Combinations of measurements relating to a particular object (e.g. another aircraft) must be grouped together to form an understanding of that object, termed perceptual grouping [Moore, 1996]. In addition to the perception of the external situation, internal situation awareness is also required. This represents a health management system capable of perceiving problems with the vehicle. In order for an awareness to be created from these individual perceptions, a means of computational Knowledge Representation (KR) is required [Negnevitsky, 2005].

Autonomous reaction is largely covered by the lower level autopilot, leaving only decision to be tackled from the PDR loop. Autonomous decision making systems come from within the field of AI. An introduction to the range of techniques available in addition to related fields such as KR and machine learning is given by Negnevitsky [2005]. An important consequence of an autonomous decision making system is a requirement for communication [Department of Defence, 2007; Negnevitsky, 2005]. For a fully autonomous UAS to be trusted by its operators it must inform them of any deviations from its flight path (such as for collision or weather avoidance). Failure to communicate intentions in this way will result in the operators losing confidence in the UAS ability to make decisions. This is analogous to the relationship which exists between a manned aircraft and ATC within Controlled Airspace (CAS) (Civil Aviation Authority [2009a] details the UK regulations). The controller is aware of the pilot's intentions as a result of a flight plan, should the pilot wish to deviate from this plan they are required to inform ATC. The decision is still made by the pilot; however it is then backed up by ATC who retain confidence in the pilots abilities. Should a fully autonomous UAS ever be operated within such CAS this communication ability becomes mandatory so as to abide by the same regulations as manned aircraft [Civil Aviation Authority, 2010b].

---

<sup>1</sup>Not to be confused with the perception element of situation awareness, this ambiguity is addressed later

---

### 2.2.1 Autonomous functions

It is generally accepted that a fully autonomous UAS is not likely to exist within the next 20 years [Department of Defence, 2005]. To further the current breed of UASs toward this goal, it is important to push for the removal of human interaction incrementally in as many areas as possible. Despite being oriented toward the battlefield the Department of Defence UAV Roadmap [Department of Defence, 2007] provides an insight into the functions currently undertaken by a pilot which would benefit from being automated.

Typical tasks currently undertaken by pilots include teaming (or swarming) of multiple vehicles, battlefield surveillance/reconnaissance, management of vehicle health, collision detection and avoidance, target recognition and prioritisation as well as generic intelligence in order to deal with the unexpected. Additional functions become important as the UAS autonomy level increases such as natural language processing to allow a vehicle to communicate its intentions intuitively and to accept commands. In addition to developing autonomous solutions to these functions for the current generation of UAS hardware, there is also a push for miniaturisation to allow Micro Air Vehicles (MAVs) to benefit from the same level of autonomy.

#### Swarming

With regards to swarming of UAS, a large amount of work in this field has been undertaken in recent years [Flint et al., 2002; Kuwata et al., 2007; Pack et al., 2009; Samek et al., 2007; Sinclair et al., 2008; Weitz et al., 2008]. The majority of this work has been focused on extending the mid level control problem of path planning to multi-vehicle platforms [Kuwata et al., 2007; Sinclair et al., 2008; Weitz et al., 2008] and the associated problems of intra-swarm collision avoidance [Samek et al., 2007]. Current work is focused on utilising UAS swarms to achieve decentralised sensing [Department of Defence, 2005] in order to further Intelligence, Surveillance and Reconnaissance (ISR) abilities [Pack et al., 2009]. It should be noted that the work in swarm collision avoidance is dedicated solely to avoiding collisions with other vehicles in the swarm. In the vast majority of cases the location of other vehicles within the swarm should be known, at least

---

approximately, by other swarm members making collision avoidance a dynamic path planning problem.

It has been noted that there is far more work in the field of swarms over single vehicle development. Whilst swarms present a major advantage to battlefield surveillance by providing multiple perspectives and redundancy, a battlefield scenario has minimal requirements for vehicle reliability. Due to the lack of onboard pilot and the minimal risk posed to civilians (relative to those already posed by a warzone), the loss of a battlefield UAS is seen as little more than an inconvenience [Manning et al., 2004]. Civilian UASs, however, are likely to operate within a NAS and are therefore required to meet far higher levels of safety. Whilst a swarm of vehicles may be invaluable in certain civil applications (law enforcement or search and rescue, for example) they cannot sacrifice safety as in a battlefield situation. A swarm of vehicles cannot be used within a NAS until the safety of each vehicle can be guaranteed. A result of this is that when considering civil UAS, the single vehicle is of far more interest.

### Collision avoidance

Collision avoidance of non-cooperative vehicles is a key requirement to enable flight within a NAS, this is often referred to as ‘sense and avoid’ after the legal requirement of human pilots to ‘see and avoid’ conflicting traffic within uncontrolled airspace [Civil Aviation Authority, 2010a]. If the conflicting vehicle is broadcasting its position with a system such as ADS-B [RTCA, 2002] and the UAS is suitably equipped to receive this signal then collision avoidance becomes, once more, a path planning problem. The CAA remind us in Civil Aviation Authority [2010b], however, that an automated UAS must abide by the ROTA [Civil Aviation Authority, 2010a] and therefore give way to objects such as gliders, hang-gliders, paragliders, microlights, balloons and parachutists<sup>1</sup> which are unable to carry a system such as ADS-B (due to cost, weight, and/or power requirements). This requires an UAS to emulate a humans ‘see and avoid’ action, leading to the use of vision processing techniques to detect conflicts [Carnie et al., 2006]. Vision is preferable to radar or laser systems as it is a passive system

---

<sup>1</sup>Assuming the UAS is powered and heavier than air

---

requiring far less power. The use of vision systems for collision avoidance also relaxes the requirement for the UAS to carry ADS-B hardware, this is especially important for MAVs. A high fidelity vision system is also able to aid vehicle control in the same way as a pilots eyes are used to augment their vestibular senses, this can allow smaller, less accurate inertial sensors to be used as in Fowers et al. [2007].

Midair collisions are presently a rare occurrence, however they are most likely to occur within the terminal area [Geyer et al., 2008; National Transportation Safety Board, 2006] due to the convergence of aircraft to one point in space. Effective collision avoidance within a terminal area must not be purely reactive because of the relatively low altitudes and airspeeds involved, severely limiting the ability of a vehicle to manoeuvre. A proactive method must be employed which utilises a high level of situation awareness to allow an UAS to plan it's route around any potentially conflicting traffic. This approach is closely aligned with the CAA guidelines requiring rational decision making based upon situation awareness [Civil Aviation Authority, 2010b].

### Automated Target Recognition (ATR)

An interesting extension to the collision avoidance requirements is laid out in Civil Aviation Authority [2010b]. For an UAS to operate within busy civil airspace it is required to comply with ATC instruction *“in the same way and within the same timeframe that the pilot of a manned aircraft would”* [Civil Aviation Authority, 2010b]. Such instructions may include being asked to follow another aircraft or confirm that the UAS has another aircraft in sight (prior to commencing an approach to land, for example). In order to achieve these functions the UAS requires not only the ability to detect (and avoid where necessary) other aircraft but it must be able to recognise them and broadly interpret their intentions. An example of such a request can be found in Civil Aviation Authority [2009a]:

✚ Pilot - *“G-CD, downwind”*

↑ ATC - *“G-CD, number two, follow the Cherokee on base”*

✚ Pilot - *“Number two, Cherokee in sight, G-CD”*

---

In this example the aircraft *G-CD*<sup>1</sup> is asked by *ATC* to follow a *Cherokee*<sup>2</sup> which is currently positioned on the base leg of an aerodrome traffic circuit (see Section 2.4). *G-CD* must confirm visual contact with the *Cherokee* and then proceed as *Number two* (to land).

An *UAS* in this situation may currently ‘see’ multiple aircraft and must determine which one the controller is referring to by using information from the controller (such as relative speed/position, type of aircraft, etc...) as well as interpretation of its intentions (i.e. it has been informed it is about to land so is likely to be at low altitude with undercarriage and flaps deployed). This level of processing is synonymous with the military requirements for *ATR* and extends it to allow the *UAS* to make decisions based upon its situation awareness. As well as being a requirement for civil applications as mentioned above, making decisions based upon *ATR* is as important for automated military *UAS* as it will allow vehicles to detect and track enemy vehicles in real time.

### **Forced landing**

Whilst collision avoidance and *ATR* are important factors for civilian *UAS*s. It should be recognised that whilst the consequences of a midair collision are often catastrophic they make up a very small number of accidents amongst General Aviation (*GA*), who are the current *NAS* users largely reliant on ‘see and avoid’. The majority of accidents are caused by mismanagement of the aircraft systems resulting in a failure, typically of the engine. 45% of the *GA* accidents in the US in 2003 were caused by either loss of engine power or other system failure whilst only 18% were caused by some form of midair collision and only 10% of these midair collisions were with other aircraft [[National Transportation Safety Board, 2006](#)]. Whilst every effort can be made to minimise the chance of powerplant or other system failure for an *UAS* it must be capable of dealing with such an event in order to be certified for use in a *NAS*. Broadly, this is the field of vehicle health management [[Reichard, 2004](#)]. Following a powerplant failure (of a single engine vehicle) there is little time to decide which course of action to take. Inevitably a descent must be made, terminating in a landing at what is most

---

<sup>1</sup>*G-xx* being the standard, shortened, form of UK aircraft registrations

<sup>2</sup>Referring to a Piper PA-28 Cherokee, a small 4 seat fixed wing aircraft

---

likely to be an undesirable location, usually termed a ‘forced landing’ [Fitzgerald et al., 2005]. Alternatives to forced landings are Ballistic Recovery Systems (BRS) which enable near vertical landings to be made under a parachute. BRS systems are undesirable in civil applications as all control of the vehicle is relinquished to the parachute therefore the landing location is governed solely by the prevailing wind.

## 2.3 Situation awareness

Situation awareness is the ability of a system to both perceive its environment and make inferences about its future states. A number of different definitions of what constitutes situation awareness exist but that given by Endsley [1988] is most suited to the conceptual shift from human to artificial situation awareness:

*“the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning and the projection of their status in the near future”* [Endsley, 1988]

Figure 2.3 illustrates the parallels between Endsley’s model and the PDR loop mentioned previously. It can be seen that Endsley’s model is closely aligned with the CAA guidelines which base decision making on situation awareness. To reconcile the PDR loop with this model, it is necessary to encapsulate situation awareness inside the perception task of PDR. The perceptual task of PDR must therefore refer to ‘perception’ of elements, ‘perception’ of meaning and ‘perception’ of future states. This introduces a great deal of ambiguity in the use of the term perception, therefore the traditional PDR loop will be abandoned in favour of Endsley’s model. Figure 2.3 also illustrates the factors, both system and individual, which play a part in gaining and maintaining situation awareness.

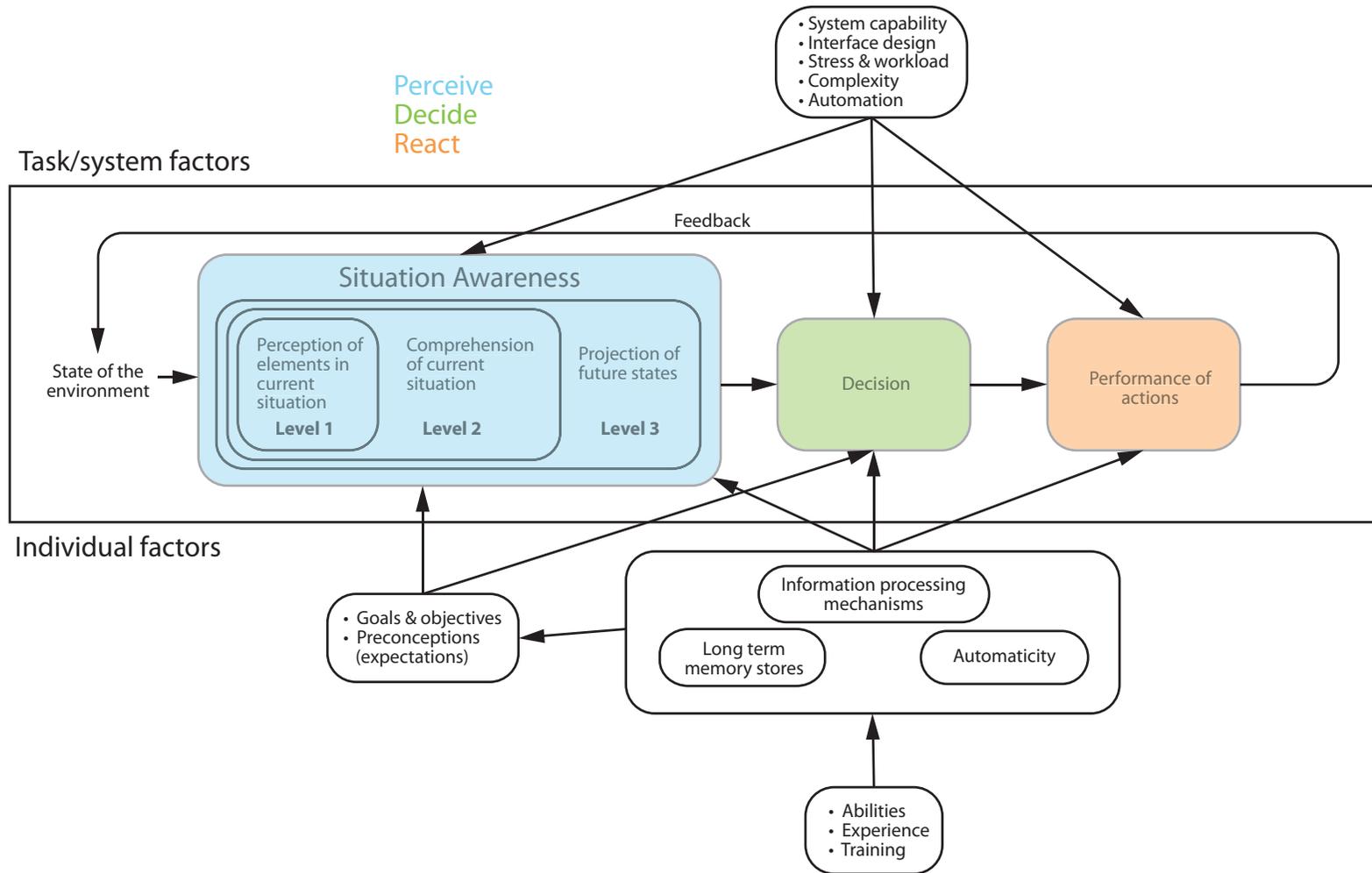


Figure 2.3: Endsley's model of situation awareness in dynamic decision making (modified from [Endsley, 1995])

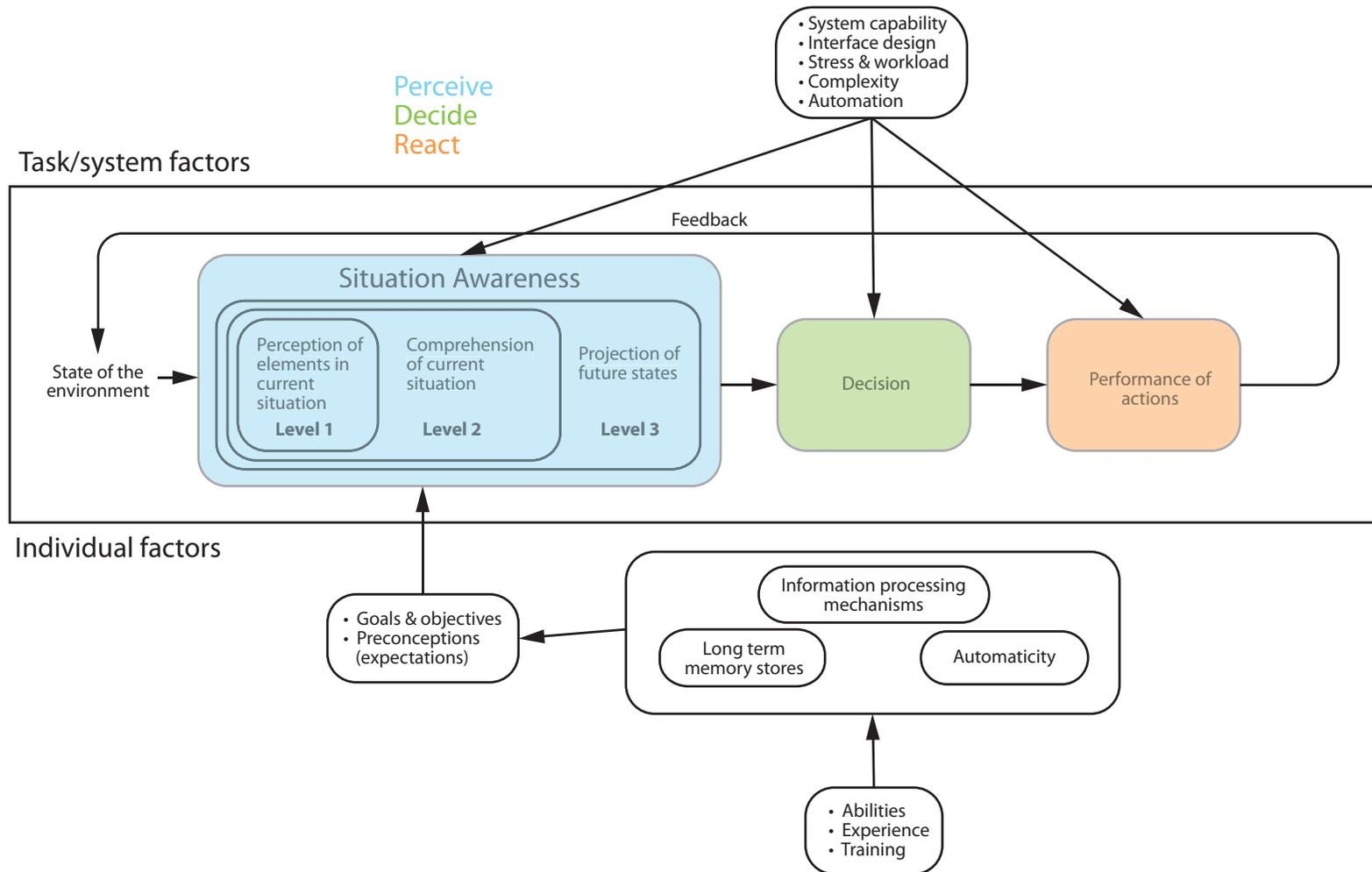


Figure 2.4: Endsley's model of situation awareness for a highly automated system. Links directly from individual to system removed (modified from [Endsley, 1995])

---

It is important to realise that remotely piloting a vehicle causes a great loss in situation awareness due to the loss of tactile and vestibular sensory information leading to a reduction of the perceptual abilities of the pilot. Whilst automation of functions within the UAS may be seen as an aid to the pilot's situation awareness, it is in fact detrimental [Wickens, 2002]. This is due to the removal of the pilot from the decision making process of onboard systems. Figure 2.4 illustrates the disconnection of the pilot from a highly automated UAS. The individual (in this case the pilot) only has an input to the system at the perceptual level with decision and action being entirely autonomous. The system, however, maintains the same level of control over all aspects of the process and, as a result, the pilot loses awareness of what the system is doing.

This compounded loss of situation awareness amongst UAS pilots is unacceptable within civil airspace as the ability to maintain equivalent safety standards to manned aircraft is lost. A means of artificially recovering the pilots situation awareness can be obtained by analysing the extreme case of a fully autonomous UAS. In order to maintain human equivalent safety a fully autonomous UAS must possess artificial situation awareness. It stands to reason that a semi-autonomous UAS must possess a degree of situation awareness in order to recover that lost by its operator [Adams, 2007]. Adams provides an introduction to situation awareness for unmanned vehicles, including a formalised definition given below.

$$SA_S = SA_H \cup SA_V \quad (2.1)$$

$$SA_V = (A \times C) \cup E \cup X \quad (2.2)$$

$$SA_{S_{shared}} = SA_H \cap SA_V \quad (2.3)$$

where  $SA_S$ ,  $SA_H$ ,  $SA_V$  are the situation awareness of the system, human and vehicle respectively.  $A \times C$  represents the contribution to vehicle situation awareness brought about by autonomy, with  $C$  being the coefficient between this and the autonomy level  $A$ .  $E$  represents the environmental characteristics of situation awareness and  $X$  those which are neither autonomous nor environmental.

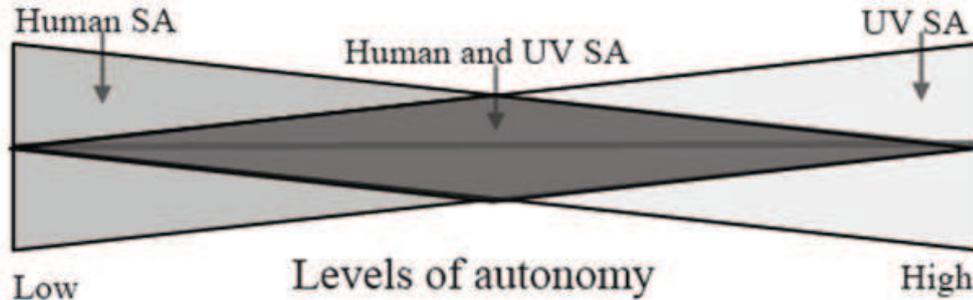


Figure 2.5: Human and UAS situation awareness [Adams, 2007]

Finally, the shared awareness of both vehicle and operator, representing common understanding, is given by  $SA_{shared}$ . Analysing this definition reveals that the situation awareness of the operator is reduced by increased vehicle autonomy, therefore the awareness of the vehicle must be increased to maintain the systems awareness, illustrated by Figure 2.5

Before artificial situation awareness may be considered, a more detailed model is needed which identifies the tasks performed at each level. Such a model was produced by Adams, overlaying Endsley’s three situation awareness levels (Figure 2.3) onto the human information processing model from Wickens et al. [2004], reproduced in Figure 2.6. Each task identified in Figure 2.6 is described in detail in Wickens et al. [2004] for information processing by the human brain, a description pertaining to artificial systems is introduced by Adams. The tasks of perception and thought decision making are of particular interest to autonomous UAS and are discussed in more detail below.

### 2.3.1 Level 1 SA - Perceptual encoding

Artificial systems have a wide range of sensing abilities far above those of human beings, such as infrared imaging and radar. Whilst limited, human sensor ability benefits from thousands of years of evolution to closely integrate the available information. Integration of artificial sensory information into a consistent stream of data is known as sensor fusion. The most widely used sensor fusion method is that of a Kalman Filter [Welch and Bishop, 2001]. Once a consistent data stream

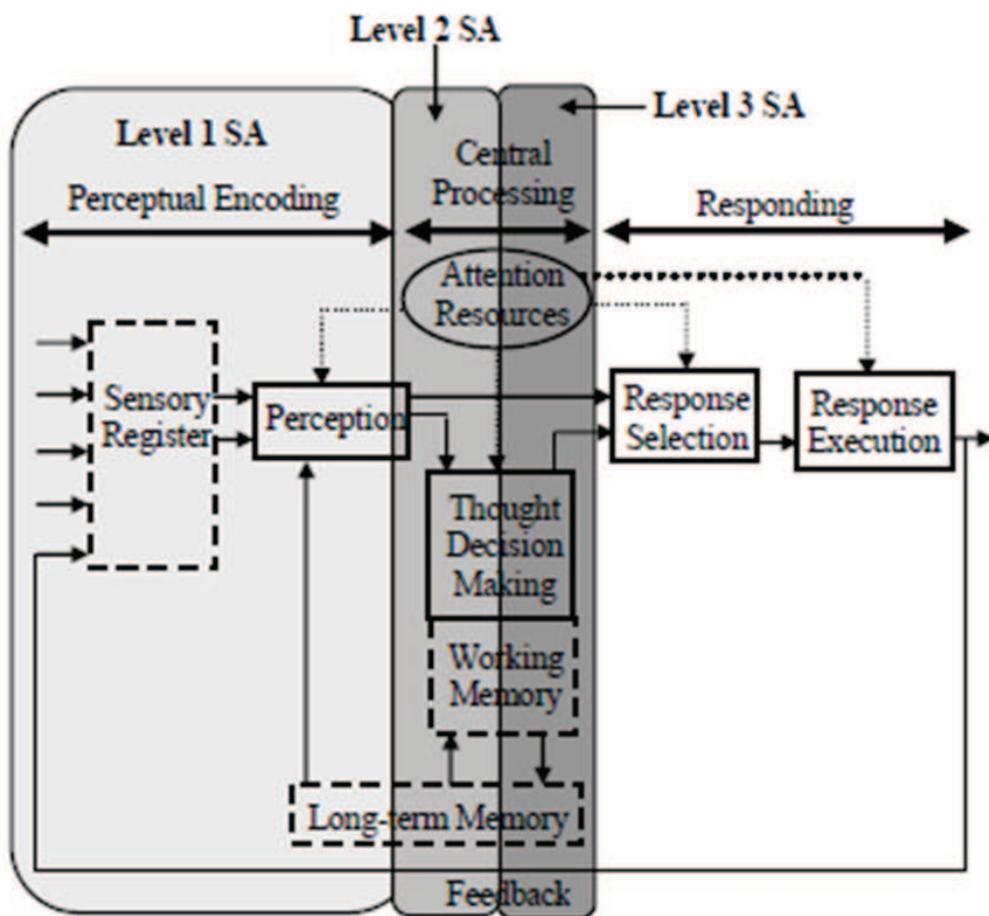


Figure 2.6: Adams model of situation awareness [Adams, 2007]

---

is available perception can begin. Perception is more than simply detecting an object within the environment, it also involves generating an understanding of that object. [Wickens et al. \[2004\]](#) suggest that this occurs through three distinct processes which may occur in parallel:

1. Bottom-up feature analysis
2. Unitisation
3. Top-down processing

Bottom-up processing starts with the data received from the sensory register and begins making sense of it. Assessment of individual features in this way starts to build up a representation of the object being perceived. Unitisation takes this representation and creates one element from it, a unit. This is done by referring to a stored database of units, within the long term memory. For example an object sensed by an [UAS](#) vision system will have features such as size, shape, relative position and velocity, colour. If these features are consistent with an aircraft they can be unitised as such and the [UAS](#) can be said to have perceived an aircraft. If no such unit exists to explain the features then the [UAS](#) has perceived an unknown object. Unitisation in this way is synonymous with the computational field of fuzzy logic which enables numerical data to be represented and processed linguistically [[Negnevitsky, 2005](#)]. In the case of an [UAS](#) sensing an aircraft fuzzy logic can be employed to characterise the vehicle by speed. For example, a vehicle travelling at a similar speed to the [UAS](#) is likely to be of a similar type (size, power, etc...), however a vehicle travelling over the speed of sound is likely to be a military fast jet.

Top down processing is the ability to fill in gaps in the sensory information based upon expectations present in the long term memory. A classic example of top down perception in aviation is radio communication. Pilots perception of information in radio messages is largely aided by knowledge of the message contents prior to reception. For example a pilot asking for permission to take off will receive either a clearance or a request to hold position and not some other message. As a result the pilot is able to jump straight to a unitised perception of the message without any bottom up processing. If a message is received which

---

was unexpected then bottom up processing must be used to make sense of the individual parts and construct a meaningful unit.

Within the context of autonomous UAS a great deal of research exists in the areas of sensing and perception of other vehicles as reviewed in Geyer et al. [2008]. This work has largely stemmed from the collision avoidance requirements discussed in Section 2.2.1 and incorporates a number of heuristics such as “*big vehicles are fast*” to aid with unitisation and top down processing.

### 2.3.2 Level 2 SA - Comprehension of meaning

In Figure 2.6 the comprehension element of situation awareness is shown as being where the outputs of perception are fed into a decision making system. Within the context of situation awareness decision making refers to making sense of the currently perceived situation and using this to predict future states. In order to make sense of the current situation an autonomous UAS which has perceived another vehicle must make inferences about its intentions. Inference is based upon experience in a similar way to perceptual unitisation, using stored information from the long term memory. It may be based upon rules (heuristics) such as “*an aircraft in that position is about to turn right*” or social norms such as “*an aircraft in that position normally turns right*”. Both heuristics and norms must be taught to human pilots before they can operate safely within a NAS. UASs operating within the same NAS must also be made aware of these heuristics and norms [Civil Aviation Authority, 2010b].

A great deal of work exists in the area of intent inference for large civil aircraft operating under a ‘free flight’ regime [Hoekstra et al., 2001; Krozel, 2000]. Free flight involves the derestriction of the NAS systems to allow civil aircraft to fly optimal paths between departure and arrival airfields in an effort to reduce flight times with subsequent saving in fuel and reduction in emissions. The importance of situation awareness within a free flight system has been highlighted in Endsley [1997] with the important conclusion that not all automation strategies (such as optimised path generation [Menon et al., 1999]) are beneficial to situation awareness. Recent work has looked at the implementation of free flight intent inference methods within a terminal area with the focus being to aid the situation

---

awareness of ATC [Seah and Hwang, 2009].

### 2.3.3 Level 3 SA - Projection of future states

With an observation of the current state of a system and an inference about its evolution over time it is possible to predict future states. For example, if a vehicle currently has 100 litres of fuel which it is using at a rate of 20 litres per hour it is expected that after an hour it will have 80 litres remaining. In this example the currently observed state is both the fuel quantity and rate of usage and the inferred evolution is that the rate will remain constant. In reality, the observations will be subject to error which accumulates quickly as projection time increases. If the error can be quantified it may be accounted for during projection to give a predicted state as a probability distribution, rather than an absolute value [Pfoser and Tryfona, 2001].

The accumulation of observation errors is a problem when projecting the state of any system. The Traffic Collision Avoidance System (TCAS) system performs a basic linear extrapolation of vehicle positions in much the same way as the fuel example above [Kuchar and Drumm, 2007]. When considering vehicles in the terminal area, however, the ability of the vehicle to manoeuvre must be considered due to their frequency (more detail on terminal area manoeuvres is given in Section 2.4). This requires an autonomous UAS to infer the future intentions of the vehicles, as described in the comprehension section above. These intentions can then be used to determine the vehicles future path. The task of predicting future positions of aircraft along arbitrary paths is routinely performed by ATC professionals, algorithms which quantify the uncertainty present in these predictions have been developed as an aid to safe operation [Crisostomi et al., 2007; Liu and Hwang, 2011].

In a similar way to the observation of present position, the inference of intentions will be subject to error which must be quantified during projection to give a representative prediction. For example, a procedure may dictate that a pilot performs a turn when five miles from the airfield. An UAS has no additional information so must use this procedure to infer future states of the vehicle. However, there is a possibility that the pilot will not manoeuvre at precisely five

---

miles, leading to a different set of future positions.

The accumulation of knowledge from perception through comprehension to projection results in information which is of higher utility to an autonomous **UAS**, that is the future states of traffic allow replanning of the **UASs** path. This increased utility is balanced by an increase in uncertainty associated with the errors introduced by each assumption, such as the assumption of the future path of a vehicle.

### 2.3.4 Response selection

Once an **UAS** possesses an acceptable level of situation awareness, it should be capable of making decisions based upon this information. Knowledge of the future positions of traffic vehicles allows an **UAS** to alter its path in order to minimise potential conflicts. This could become a highly complex optimisation problem if a large number of traffic vehicles are present resulting in no single best solution. However, manned aviation has been subject to this same problem in the terminal area for many years. The result is a number of regulations and heuristics which aid pilots in making safe decisions with respect to other vehicles, these are discussed in the next section. These regulations are not only an aid to decision making by an autonomous **UAS**, they must be followed to ensure transparency and equivalence [[Civil Aviation Authority, 2010b](#)].

A rule based decision making system is best suited to this highly regulated framework as it is easy to construct (one rule per regulation) and computationally simple to implement [[Negnevitsky, 2005](#)]. A major downside to a rule based system (over a neural network, for example), is its inability to deal with unexpected situations. As it is not anticipated that fully autonomous **UAS** will operate entirely without human oversight, it is possible to hand the decision making authority back to a human operator in the case of an unexpected situation for which the **UAS** has no rules governing its decision.

---

## 2.4 Terminal area

A terminal area is the airspace immediately surrounding an airfield which is predominantly used by aircraft arriving at or departing from that airfield. Terminal areas are often within controlled airspace such as Aerodrome Traffic Zones (ATZs) or Terminal Manoeuvring Areas (TMAs) to provide protection from aircraft not in communication with the airfield [Civil Aviation Authority, 2010a,c]. A TMA (also known internationally as Terminal Control Area (TCA)) is defined by the International Civil Aviation Organization (ICAO) as being “established at the confluence of airways in the vicinity of one or more major aerodromes” [Civil Aviation Authority, 2009a]. This definition alludes to a problem facing terminal area operations, the convergence of a number of aircraft on a small region of airspace. It is for this reason that the majority of midair collisions occur within the terminal area [National Transportation Safety Board, 2006]. It should be noted that only arriving aircraft are to be considered from this point forward due to the problems faced by their convergence.

Two primary means of approaching an airfield exist depending on the available facilities and prevailing weather conditions; visual and instrument approaches. A visual approach is used at smaller airfield which do not possess instrument landing facilities (such as radar or Instrument Landing System (ILS) [Thom, 2003]) when weather conditions permit arriving aircraft to see the airfield and other arriving traffic, known as Visual Meteorological Conditions (VMC) [Civil Aviation Authority, 2008]. When weather conditions prohibit visual acquisition (Instrument Meteorological Conditions (IMC) [Civil Aviation Authority, 2008]), airfields without instrument facilities do not operate. Larger airfield with instrument facilities continue to operate under Instrument Flight Rules (IFR) [Civil Aviation Authority, 2008], where radio navigation equipment is used to guide aircraft and ATC oversee separation between traffic. The following sections cover the standard procedures employed for both approach types.

### 2.4.1 Visual approaches

To facilitate the safe operation of aircraft approaching an airfield under Visual Flight Rules (VFR) a number of standard procedures have been developed. For

---

an airfield which possesses no [ATC](#) facilities (an uncontrolled airfield) a typical procedure is the standard Overhead Join ([OHJ](#)), [Figure 2.7](#). An [OHJ](#) is a means of joining an airfield traffic circuit by first flying overhead to observe the conditions (wind sock and runway surface) and any traffic in the circuit. Once an overflight has been conducted, a descent to circuit height (usually 1000ft Above Aerodrome Level ([AAL](#))) is made and the aircraft joins the circuit via the crosswind leg. The circuit is a rectangular flight path flown by all arriving aircraft which ensures that the runway and all other traffic remain in sight at all times, shown in white in [Figure 2.7](#).

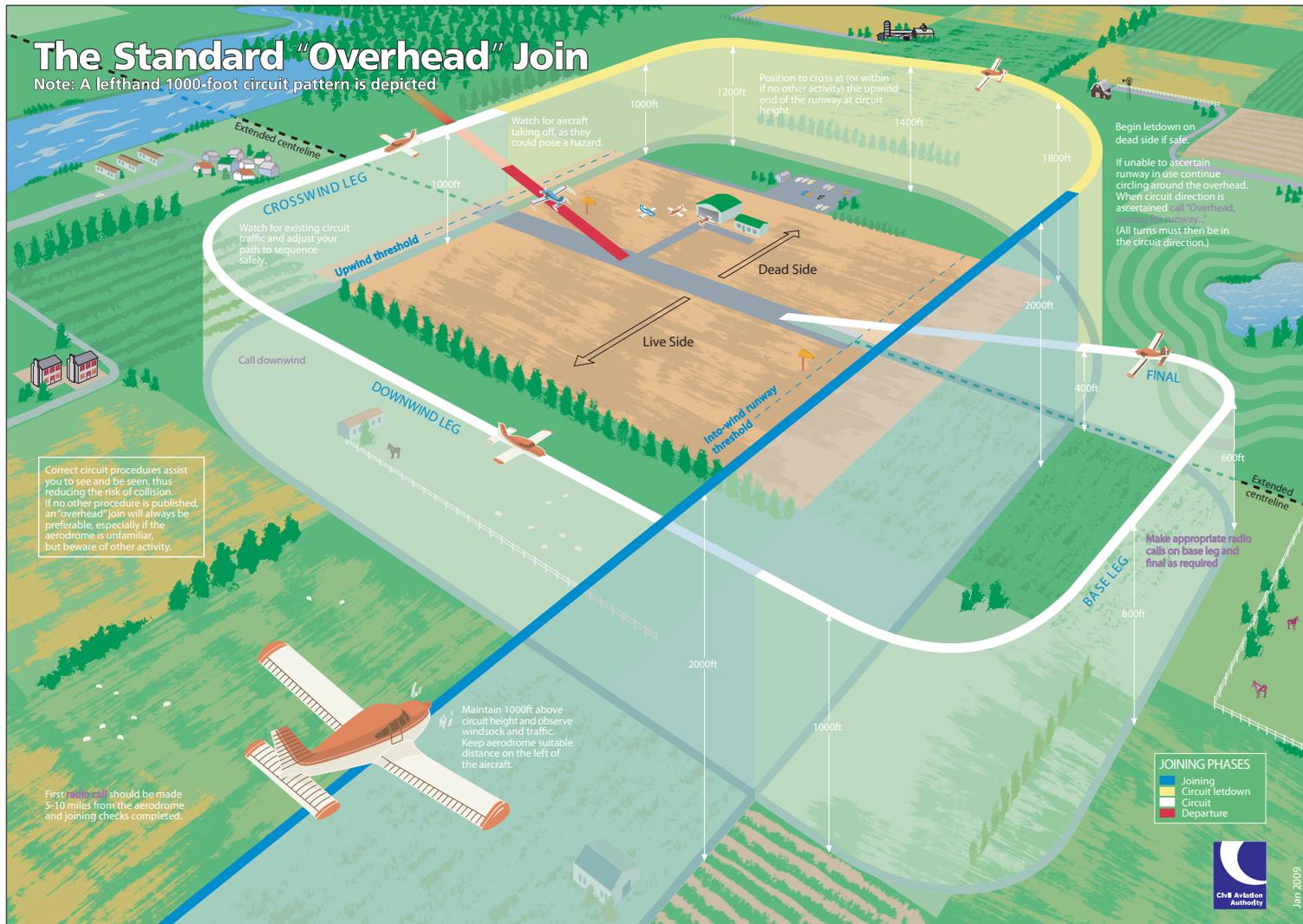


Figure 2.7: Standard overhead join [Civil Aviation Authority, 2009b]

---

If an airfield is equipped with [ATC](#) facilities, it is common for arriving aircraft not to perform an [OHJ](#) but instead join the circuit directly via one of the legs. If a non-authoritative [ATC](#) service is provided such as an Aerodrome Flight Information Service ([AFIS](#)) [[Civil Aviation Authority, 2009a](#)], an arriving aircraft would announce its current position and intention to join to which [ATC](#) would provide traffic information allowing the pilot to sequence appropriately<sup>1</sup>:

✚ Pilot - *“G-CD, five miles south east, joining left base for runway 23”*

✚ ATC - *“G-CD roger, one turning base, one mid-downwind”*

At airfields equipped with fully authoritative [ATC](#), an arriving aircraft will likely be instructed how to join and told how to sequence<sup>1</sup>:

✚ Pilot - *“G-CD, eight miles south west, request joining instructions”*

✚ ATC - *“G-CD join downwind left hand for runway 27, number 3 to the Cessna 172 joining base”*

✚ Pilot - *“Join left downwind for 27, number 3, G-CD”*

In this example a joining aircraft is informed that it is the third in the queue to land, which implies that [ATC](#) expect the pilot to obtain visual contact with the aircraft second in the queue and maintain adequate separation. To aid the pilot in visually acquiring the traffic, [ATC](#) provide information about the aircraft type and its position.

## 2.4.2 Instrument approaches

The previous section has covered airfields which are operating under [VFR](#), that is the weather conditions are defined as [VMC](#) and aircraft are expected to navigate and separate from traffic based on visual acquisition [[Civil Aviation Authority, 2008, 2011a](#)]. An airfield operating in weather conditions defined as [IMC](#) require aircraft to operate under [IFR](#) as visual acquisition may be impossible. Terminal area operations under [IFR](#) fall in to two distinct categories:

---

<sup>1</sup>These examples are shortened to highlight the key points, actual communication would contain additional information

- 
- Procedural
    - Separation assurance is provided by a set of standard procedures, similar to the [OHJ](#) and circuit but without the need for visual acquisition.
  - Vectored
    - Separation assurance is provided [ATC](#) using radar vectoring

Figure 2.8 illustrates a typical procedural instrument approach chart used to communicate the standard procedure to pilots. These charts provide horizontal and vertical guidance with respect to fixed ground based beacons or augmented Global Navigation Satellite System ([GNSS](#)) [[Civil Aviation Authority, 2007](#)]. Pilots provide [ATC](#) with position reports at critical points (such as when passing over a beacon) enabling [ATC](#) to ensure separation is maintained between all vehicles. Procedural approaches use holding patterns to separate aircraft vertically prior to commencement of the final approach, this makes them popular at airfields without a radar installation as [ATC](#) oversight is needed less often.

An airfield equipped with a radar system will primarily use vectored approaches, that is [ATC](#) provide continuous guidance to the pilot. This enables [ATC](#) to position aircraft at the minimum safe separation to ensure maximum runway utilisation. Vectored approaches don't require holding patterns to be used as [ATC](#) can simply vector an aircraft on a longer route to provide increased separation, this has enabled the proliferation of Continuous Descent Approaches ([CDAs](#)) to help reduce the environmental impact at airfields [[Civil Aviation Authority, 2001](#)].

### 2.4.3 Approach summary

The previous sections have discussed the two primary means of an aircraft arriving at an airfield. Operation of [UASs](#) under [IFR](#) has a lower situation awareness requirement with regards to traffic vehicles because separation assurance is provided by [ATC](#). It is unlikely, however, that [UASs](#) will operate from larger airfields (equipped with instrument facilities) in the first instance. This is due to the high volumes of traffic already present and the potential for disruption should an [UAS](#)

INSTRUMENT APPROACH CHART - ICAO

**CRANFIELD  
ILS/DME/NDB(L)  
RWY 21  
(ACFT CAT A,B,C)**

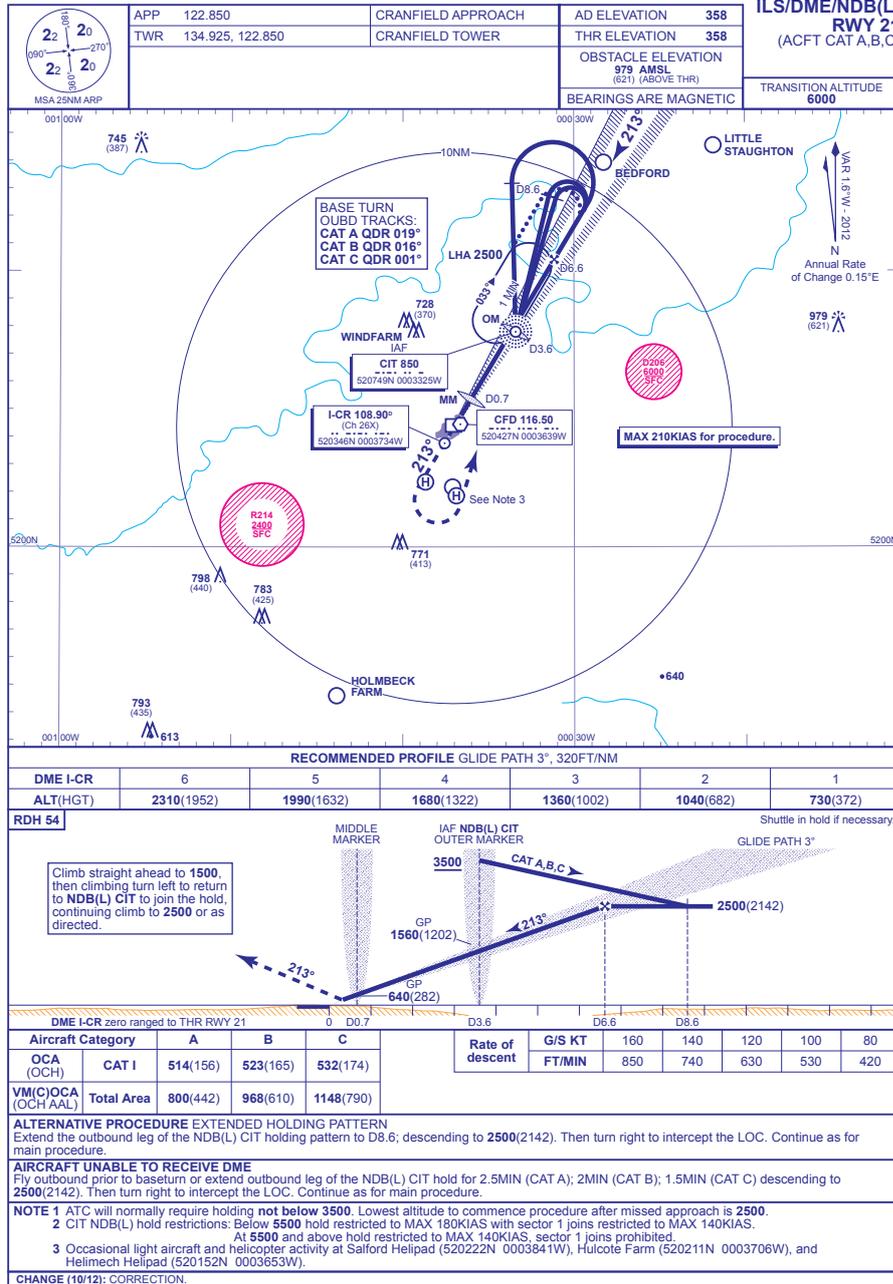


Figure 2.8: Procedural approach example [Civil Aviation Authority, 2012]

---

malfunction. Additionally, a number of potential **UAS** users currently operate from smaller airfields as a means of reducing operating costs, such as UK law enforcement [Hayles, 2006]. For these reasons, this thesis will focus on the integration of **UAS** into a visual terminal area at a small uncontrolled airfield.

In the near term it is possible to envisage civilian **UAS** operations being entirely segregated from that of manned aircraft. For example prior to arrival of an **UAS**, an airfield may become the subject of temporary restricted airspace preventing manned aircraft from operating while the **UAS** is airborne. This is the model presently employed by the British Army at Camp Bastion, Afghanistan [Button, 2009]. This approach will enable **UASs** to begin operating in civil applications more quickly, however it does not represent a permanent solution as it impedes the operation of other aircraft [Civil Aviation Authority, 2010b]. Therefore, to enable autonomous **UASs** to operate routinely at small airfields a large degree of situation awareness is required.

#### 2.4.4 Situation awareness

To facilitate the quantification of situation awareness in aviation professionals, requirements have been identified for a number of different operators including fighter pilots [Endsley, 1993], **ATC** [Endsley and Rodgers, 1994] and commercial pilots [Endsley et al., 1998]. These requirements detail the information which a pilot must possess in order to be considered to have good situation awareness. The information requirements for commercial pilots provide a comprehensive breakdown of what knowledge a pilot must possess to achieve each level of situation awareness. Of particular interest to **UAS** operation in a visual terminal area is knowledge of other vehicles in the vicinity and the impact of their presence on the desired flight path of the **UAS**. With this key requirement in mind, the requirements for an artificial system to maintain situation awareness are laid out below.

- Level 1 - Perception
  - Positions of other vehicles
  - Types of vehicles (e.g. large jet, small piston, **UAS**)

- 
- Clearances issued to ownship and other vehicles
  - Level 2 - Comprehension
    - Separation from other vehicles
    - Relative trajectory of other vehicles
    - Intentions of other vehicles
    - Hazards currently posed by other aircraft
  - Level 3 - Projection
    - Trajectory of ownship and other vehicles
    - Separation from other vehicles

Referring back to Figure 2.6 it can be seen that a vehicle in possession of the knowledge listed above is capable of moving beyond situation awareness and into the response selection and execution, with regards to traffic. The above list can therefore be said to represent the external situation awareness requirement for a UAS operating within a terminal area. Additional requirements are present for internal situation awareness, such as information regarding aircraft systems operability, but these are beyond the scope of this thesis.

The perception of traffic vehicles by an UAS is broadly similar to the task of collision avoidance, however in the terminal area an additional Ground Based Sense And Avoid (GBSAA) system may be employed at the airfield [Lacher et al., 2010]. It has been noted that a large body of work exists in the field of traffic detection for collision avoidance purposes. In this thesis it will be assumed that a means of detecting the positions of traffic aircraft in the vicinity exists, sufficient to provide the perceptual information to the UAS.

Knowledge of vehicle type is required for the UAS to predict its kinematic behaviour. This could be obtained via ADS-B [RTCA, 2002], by voice communication between the UAS [Civil Aviation Authority, 2010a] or visual acquisition. For this thesis it is assumed the UAS has precise knowledge of the vehicle type via one of these methods.

Finally, the assumption that the UAS is operating at an uncontrolled airfield implies that no clearances can be issued by ATC, that is all decisions are the

---

responsibility of the individual vehicles [Civil Aviation Authority, 2010a]. This simplifies the analysis by removing the need to consider interaction with ATC by the UAS.

Utilising these assumptions, this thesis will focus on the development of comprehension and projection methodologies for an autonomous UAS operating in a visual terminal area at an uncontrolled airfield.

## 2.5 Summary

This chapter has reviewed the literature in the field of autonomous UASs, with particular focus on civilian terminal area operations. The benefits and challenges of autonomy for UASs have been discussed and a number of autonomous functions already present on production UASs introduced. The requirement for civilian UASs to possess equivalent levels of situation awareness to human pilots lead to a detailed discussion on human and artificial situation awareness functions.

Operations in the terminal area were introduced, focusing on the procedures currently in use by manned aircraft. These procedures allow an UAS to operate readily at a large airfield utilising ATC as a means of separation assurance, however operation at smaller airfields requires high levels of situation awareness by the UAS. The procedures in place at small airfields provide an UAS a great deal of information about the intentions and future states of traffic vehicles. This allows for a situation awareness system to be developed, the requirements of which have been discussed.

The three elements of situation awareness have been discussed and their meaning with regards to the terminal area introduced. The perception element has seen a lot of development over recent years due to the collision avoidance requirements for UASs, for this thesis it will be assumed adequate perceptual information is available from such a system. Some work exists in the field of comprehension and projection as aids to the operation of ATC, none of which has been applied to the field of autonomous UAS.

The following chapters detail the development of an artificial situation awareness system for a fully autonomous UAS operating within the terminal region at a small airfield with some degree of ATC oversight, but no radar facilities.

# Chapter 3

## Comprehension

The previous chapter introduced the situation awareness requirement for an autonomous [UAS](#) operating in the terminal area of a small uncontrolled airfield. Of primary concern are the current and future states of traffic aircraft in the vicinity, which an [UAS](#) may then use to plan a conflict free path. Due to the large body of work in the field of vehicle detection by [UASs](#) it is assumed that perception of traffic position is provided by on-board sensors (such as cameras or radar) or other systems (such as [ADS-B](#)). This chapter details the use of these position observations within a comprehension system to estimate additional state information and infer the intent of the vehicle, the requirements laid out in [Section 2.3.2](#). Future chapters use this information to predict future states of the vehicle and replan a route as appropriate.

Small uncontrolled airfields do not benefit from precisely defined paths which arriving aircraft follow. Instead a traffic circuit is used, the dimensions of which are dependant on the type of aircraft and defined with respect to the runway. This traffic circuit forms the basis of the Circuit Coordinate System ([CCS](#)) which is used to represent perceived traffic positions. Additional state information, such as the velocity vector of a vehicle, can be estimated from sequential position estimates. This state information is continuous in nature, that is it can take on an infinite number of values. However, intent information is inferred based on a set of discrete modes, or circuit legs. It is necessary, therefore, to estimate the discrete mode along with the continuous states to enable projection to take place, detailed in the next chapter.

---

The State Dependent Transition Hybrid Estimation (**SDTHE**) algorithm for estimating the continuous state and discrete mode of aircraft has been developed in [Seah and Hwang \[2009\]](#), which forms the basis of this chapter. The **SDTHE** algorithm was developed to provide more accurate continuous state measurement for **ATC** by incorporating knowledge of its discrete mode from published arrival procedures at large airfields. Hybrid estimation provides improved continuous state estimates by tailoring the vehicle models used to specific modes, for example straight or turning flight. It is well suited to terminal area tracking as each discrete mode corresponds to a particular manoeuvre whose dynamics can be captured by the associated model. This chapter extends the **SDTHE** technique to an **UAS** operating at a small uncontrolled airfield where no published routes exist and discrete modes are defined based on **CAA** guidelines, airfield layout, aircraft performance and heuristics (see Section 2.4.1).

The next section defines the **CCS** and generalised circuit path which will be used throughout. Section 3.2 introduces the principle of hybrid state estimation and details its application to the comprehension problem. Finally, Section 3.3 demonstrates the performance of the comprehension element in isolation with a simple simulation.

## 3.1 Circuit definitions

### 3.1.1 Circuit Coordinate System (**CCS**)

Visual operation within the terminal area is conducted with reference to the active runway, determined by the current wind direction. The first comprehension task is to transform the positions of traffic from the Global Coordinate System (**GCS**) of latitude ( $\phi_t$ ), longitude ( $\lambda_t$ ) and altitude ( $a_t$ ) to a runway based **CCS**, illustrated in Figure 3.1. The upwind axis ( $u$ ) is defined as along the centreline of the active runway in the direction of landing and crosswind ( $c$ ) is defined as pointing toward the ‘live’ side of the runway which is determined by circuit direction. Height is not shown in Figure 3.1 but is defined as positive up. The origin of the **CCS** is the runway threshold.

Transformation from **GCS** to **CCS** is a two stage process. First the position

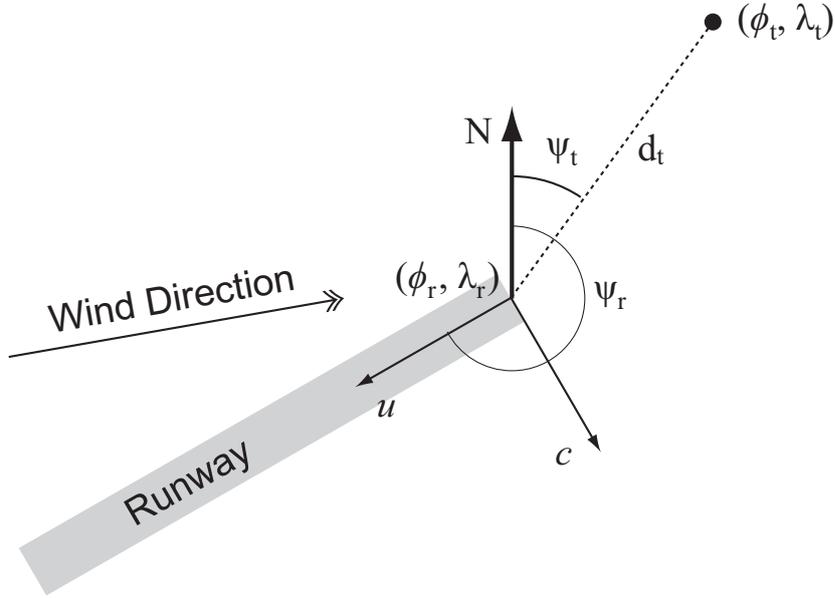


Figure 3.1: Coordinate System Transformation

of the traffic  $(\phi_t, \lambda_t)$  is transformed to a distance and bearing  $(d_t, \psi_t)$  from the runway threshold  $(\phi_r, \lambda_r)$ . This is done using the formulae given in (3.1) and (3.2), where  $R$  is the radius of the Earth ( $6378km$ ), see Appendix A for derivation. This polar representation is then resolved into the Cartesian CCS with (3.3) which also includes the height translation. In addition to performing these translations for all traffic vehicles, the position of the UAS is also represented in the CCS.

$$d_t = 2R \arctan \left( \frac{\sqrt{\sin^2 \left( \frac{\phi_t - \phi_r}{2} \right) + \cos \phi_r \cos \phi_t \sin^2 \left( \frac{\lambda_t - \lambda_r}{2} \right)}}{\sqrt{1 - \sin^2 \left( \frac{\phi_t - \phi_r}{2} \right) - \cos \phi_r \cos \phi_t \sin^2 \left( \frac{\lambda_t - \lambda_r}{2} \right)}} \right) \quad (3.1)$$

$$\psi_t = \arctan \left( \frac{\sin(\lambda_t - \lambda_r) \cos(\phi_t)}{\cos \phi_t \sin \phi_r - \sin \phi_r \cos \phi_t \cos(\lambda_t - \lambda_r)} \right) \quad (3.2)$$

$$\begin{pmatrix} u_t \\ c_t \\ h_t \end{pmatrix} = d_t \begin{pmatrix} \cos(\psi_r - \psi_t) \\ \sin(\psi_r - \psi_t) \\ a_t - e_r \end{pmatrix} \quad (3.3)$$

---

where  $h_t$  is the height of the traffic above the runway threshold and  $e_r$  is the elevation of the runway threshold Above Mean Sea Level (AMSL).

### 3.1.2 Discrete modes

The CCS is used to represent the continuous states of traffic vehicles, such as position and velocity. To enable intent inference, a number of discrete modes must also be defined relating to the phase of the circuit in which the traffic resides. Figure 3.2 illustrates these discrete modes in addition to a number of Flight Change Points (FCPs) which govern the transition from one state to the next. For example, FCP4 represents the point at which a vehicle transitions from the ‘Late Downwind’ state to the ‘Turning on to Base’ state.

## 3.2 Hybrid state estimation

The perceived position information of traffic, represented in the CCS, will be imperfect due to sensor inaccuracies and time delays associated with communication. A popular means of providing a state estimate based on imperfect measurements is the Kalman filter, a recursive state estimation technique [Bar-Shalom et al., 2001]. In addition to uncertainties in traffic state, it is impossible to accurately define the FCPs a priori as the precise point at which a vehicle is going to manoeuvre is hidden from the UAS. The problem of estimating both aircraft state and discrete locations is well suited to an Interacting Multiple Model (IMM) estimator. Such an estimator contains a number of individual state estimators optimised for each discrete mode, the covariance of each estimator being used to determine the likelihood of that discrete mode being correct [Bar-Shalom et al., 2001].

Typical implementations of IMM estimators assume that transitions between discrete modes occur with a constant probability, this approach is not well suited to traffic state estimation within the terminal area. For example a vehicle positioned early on the downwind leg has a very small (zero) probability of transitioning to the base leg, however an IMM estimator would not make this distinction. An extension to the IMM principle is the SDTHE algorithm which conditions

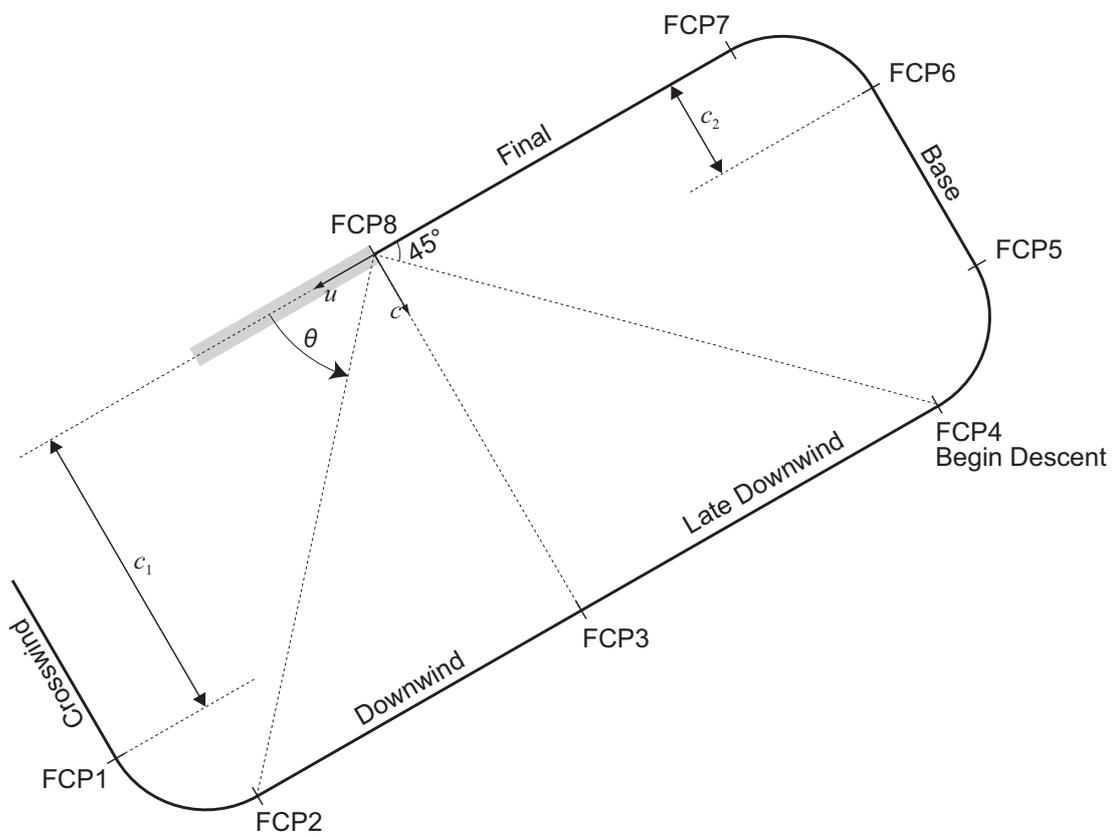


Figure 3.2: Discrete modes within a circuit pattern

---

Table 3.1: Terminal area discrete modes

Mode	Terminating FCP	Sub modes	Bearing
1. Crosswind	FCP1	CV CH	$0 < \theta \leq 45$
2. Downwind Turn	FCP2	CT CH	$0 < \theta \leq 45$
3. Downwind	FCP3	CV CH	$45 < \theta \leq 90$
4. Late Downwind	FCP4	CV CH	$90 < \theta \leq 135$
5. Base Turn	FCP5	CA CD	$135 < \theta \leq 180$
6. Base	FCP6	CV CD	$135 < \theta \leq 180$
7. Final Turn	FCP7	CT CD	$135 < \theta \leq 180$
8. Final	FCP8	CV CD	$\theta = 180$
9. Runway	-	-	-

the transition probabilities on the currently estimated state [Seah and Hwang, 2009]. The SDTHE algorithm breaks down each discrete mode into lateral and vertical planes. In each state a vehicle may be in one of three lateral sub-modes: Constant Velocity (CV), Constant Turn (CT) or Constant Acceleration (CA) and one of two vertical sub-modes: Constant Height (CH) or Constant Descent (CD). Table 3.1 shows the sub-mode combinations for the discrete modes illustrated in Figure 3.2.

This section details the operation of the SDTHE algorithm as applied to the circuit pattern in Figure 3.2. The next subsection defines the aircraft models used for each of the lateral and longitudinal sub-modes. The subsequent subsections detail the mode transitions and algorithm structure.

### 3.2.1 Aircraft model

Each of the sub-modes requires an associated flight model to estimate the continuous states. This model takes the form of a linear discrete time state space model which forms the basis of a Kalman filter for each mode. The form of the state update equation is

$$x(k+1) = A_i x(k) + W_i \omega(k) \quad (3.4)$$

where  $x(k)$  is the state at time step  $k$ ,  $A_i$  is a coefficient matrix and  $W_i \omega(k)$  is the process noise of the  $i$ th mode. The process noise is modified from the standard

---

form to include the coefficient matrix  $W$ , this allow an intuitive and consistent definition of noise to be used across all sub-modes as will be seen later.

The complete state of a traffic vehicle is given as

$$x(k) = \begin{bmatrix} u & \dot{u} & \ddot{u} & c & \dot{c} & \ddot{c} & h & \dot{h} \end{bmatrix}^T \quad (3.5)$$

where  $T$  denotes the transpose of the vector.

Vertical acceleration can be ignored as changes in vertical velocity occur quickly when transitioning in to or out of the [CD](#) sub-mode, however lateral accelerations persist during turns (i.e. the [CT](#) sub-mode).

Each flight model has the same measurement equation as it is assumed the same perceptual information is available from [\(3.3\)](#) in each state.

$$z(k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} x(k) + \nu(k) \quad (3.6)$$

where  $\nu(k)$  is the measurement noise.

For clarity, each sub-mode will only refer to the relevant elements of the state vector, either lateral ( $x_l(k)$ ) or vertical ( $x_v(k)$ )

$$x_l(k) = \begin{bmatrix} u & \dot{u} & \ddot{u} & c & \dot{c} & \ddot{c} \end{bmatrix}^T \quad x_v(k) = \begin{bmatrix} h & \dot{h} \end{bmatrix}^T \quad (3.7)$$

[Figure 3.3](#) illustrates the lateral model. The actual path is shown as a dashed line which has been discretised at two points,  $k$  and  $k + 1$ . At each points the velocity and acceleration vectors are decomposed in to their respective components in the [CCS](#). For illustration purposes the time step is assumed to be unity. The position of the second point has advanced by slightly more than the velocity vector of the first point due to the acceleration between points. The velocity at the second point is modified precisely by the acceleration at the first point.

It is clear from [\(3.6\)](#) that neither the velocities nor accelerations of the vehicle are observed, therefore these must be modelled as random variables. To enable the same structure of model to be used for each sub-mode the random variables will represent accelerations in the system, therefore the general form of the lateral

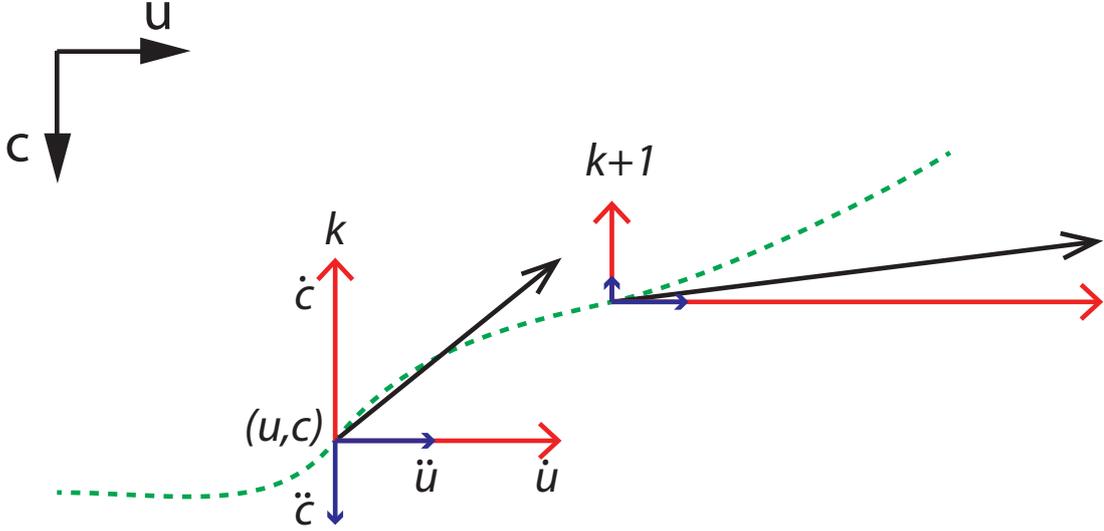


Figure 3.3: Discretisation of a continuous flight path

model is

$$x_l(k+1) = \begin{bmatrix} 1 & dt & dt^2/2 & 0 & 0 & 0 \\ 0 & 1 & dt & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & dt & dt^2/2 \\ 0 & 0 & 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} x_l(k) + \begin{bmatrix} dt^2/2 & 0 \\ dt & 0 \\ 1 & 0 \\ 0 & dt^2/2 \\ 0 & dt \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \omega_u \\ \omega_c \end{bmatrix} \quad (3.8)$$

where  $dt$  is the time-step,  $\omega_u$  and  $\omega_c$  are the random perturbations in accelerations in each axis which are assumed to be Gaussian with zero mean.

To improve the accuracy of state estimation in each sub-mode, whilst maintaining the simplicity of a linear model, it is possible to modify (3.8) to account for the characteristics of that sub-mode. The modified models used for each sub-mode are detailed in the following sections.

### Constant Velocity sub-mode

The **CV** sub-mode assumes that a vehicle is travelling at a constant speed on a fixed heading but is subject to random variations in its flight path. Equation

---

(3.8) can be modified for this sub-mode by removing the steady state accelerations (the third and sixth rows of the  $A$  matrix). The  $W$  matrix remains unmodified as short term accelerations (such as erroneous measurements or wind disturbances) do effect velocity and position.

$$x_l(k+1) = \begin{bmatrix} 1 & dt & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & dt & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} x_l(k) + \begin{bmatrix} dt^2/2 & 0 \\ dt & 0 \\ 1 & 0 \\ 0 & dt^2/2 \\ 0 & dt \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \omega_{u_{CV}} \\ \omega_{c_{CV}} \end{bmatrix} \quad (3.9)$$

where  $\omega_{u_{CV}}$  and  $\omega_{c_{CV}}$  represent the acceleration noise for the CV sub-mode whose covariance matrix is given as

$$Q_{CV} = \begin{bmatrix} \sigma_{u_{CV}}^2 & 0 \\ 0 & \sigma_{c_{CV}}^2 \end{bmatrix} \quad (3.10)$$

The variance terms  $\sigma_{u_{CV}}^2$  and  $\sigma_{c_{CV}}^2$  correspond to how accurately a vehicle is assumed to hold speed.

### Constant Turn sub-mode

The CT sub-mode assumes the vehicle is performing a turn whilst subject to random disturbances. To maintain a steady turn a constant acceleration is required (toward the centre of the turn) despite the magnitude of velocity remaining constant. To allow this steady state acceleration to persist, no modification is made to (3.8) for the CT sub-mode. The process noise for the CT sub-mode is similar to that for the CV sub-mode with covariance matrix

$$Q_{CT} = \begin{bmatrix} \sigma_{u_{CT}}^2 & 0 \\ 0 & \sigma_{c_{CT}}^2 \end{bmatrix} \quad (3.11)$$

However, the variance terms now represent both the speed and rate of turn accuracy.

---

### Constant Acceleration sub-mode

The **CA** sub-mode assumes the vehicle is accelerating whilst flying on a constant heading. This sub-mode uses exactly the same state equation as the **CT** sub-mode but features a noise covariance matrix which is constrained to allow accelerations only in the direction of flight, shown below

$$Q_{CA} = \begin{bmatrix} (\sigma_{CA} \cos(\psi_l(k)))^2 & 0 \\ 0 & (\sigma_{CA} \sin(\psi_l(k)))^2 \end{bmatrix} \quad (3.12)$$

where  $\psi_l(k)$  is the heading of the vehicle relative to the **CCS** and is given by

$$\psi_l(k) = \arctan \left( \frac{\dot{c}(k)}{\dot{u}(k)} \right) \quad (3.13)$$

Equation (3.12) couples the process noise variances in each axis by assuming the magnitude is constant ( $\sigma_{CA}^2$ ). For example a vehicle heading directly along the upwind axis will have a covariance matrix of

$$Q_{CA} = \begin{bmatrix} \sigma_{CA}^2 & 0 \\ 0 & 0 \end{bmatrix} \quad (3.14)$$

allowing accelerations in this direction only.

### Constant Height sub-mode

The **CH** sub-mode assumes that the vehicle is maintaining a constant height but is subject to random disturbances. The state equation for the **CH** sub-mode is given below

$$x_v(k+1) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} x_v(k) + \begin{bmatrix} dt \\ 0 \end{bmatrix} \omega_{CH} \quad (3.15)$$

As accelerations in height are usually small, these are not considered. Instead the process noise is given as a velocity which is not propagated forward (much like the accelerations in the **CV** sub-mode). Once again the noise covariance may be tweaked to improve Kalman filter performance.

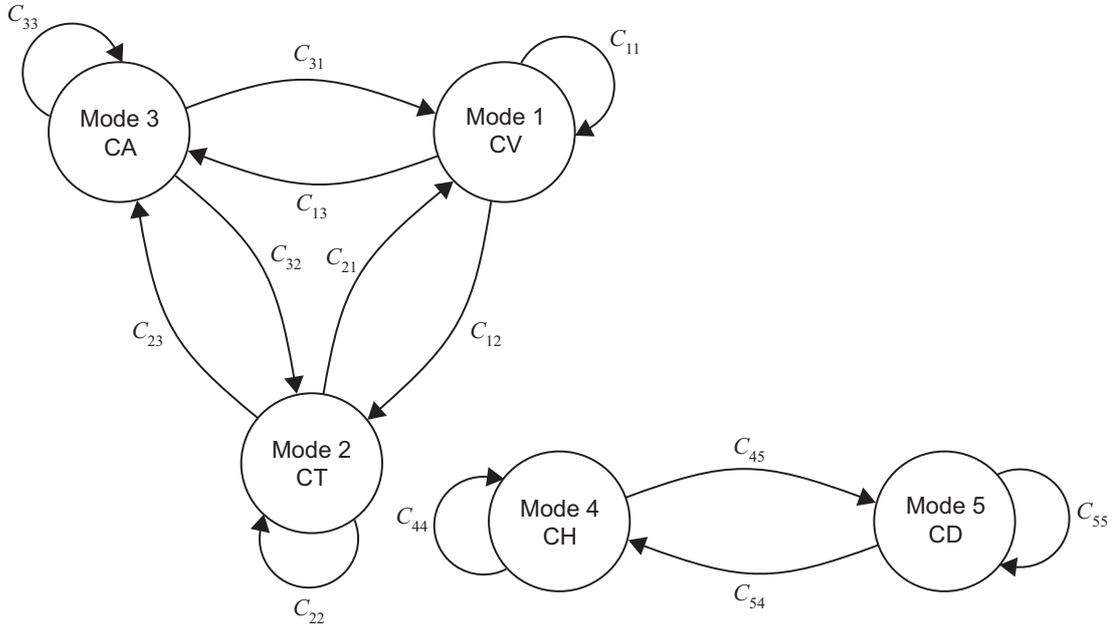


Figure 3.4: Sub-mode transition Markov diagrams

### Constant Descent sub-mode

The **CD** sub-mode assumes that a vehicle is descending at a constant rate but subject to random disturbances. The state equation, given below, is similar to that of the **CH** sub-mode, but velocity disturbances are propagated forward.

$$x_v(k+1) = \begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix} x_v(k) + \begin{bmatrix} dt \\ 1 \end{bmatrix} \omega_{CD} \quad (3.16)$$

As with the **CH** sub-mode, the process noise is given as a velocity and may be tweaked as appropriate.

### 3.2.2 Flight Change Points

Transition between modes (sub-mode pairs) occurs at the **FCPs** and is assumed to be Markovian in that it is only conditioned on the previous mode, as shown in Figure 3.4. For example, referring to Table 3.1, **FCP2** represents the transition from the Downwind Turn to the Downwind Leg and is a transition from the **CT** to **CV** sub-mode ( $C_{21}$ ) whilst remaining in the **CH** sub-mode ( $C_{44}$ ).

---

The transition guard conditions ( $C_{ij}$ ) are dependant on the current state and are represented in the form

$$C_{ij} = L_x x + L_\theta \theta^* \leq 0 \quad (3.17)$$

where  $L_x$  is the coefficient matrix of the continuous state ( $x$ ) and  $L_\theta$  is the coefficient matrix of the state of the FCP ( $\theta^*$ ). It is assumed that the FCP state ( $\theta^*$ ) is not known exactly but instead will be modelled as a Gaussian variable.

The guard condition represents the conditions under which the vehicle remains in the current mode, such that if these conditions are no longer met the transition will occur. Referring to Figure 3.2 the guard conditions for each of the FCPs can be determined.

### FCP1 - Crosswind

The crosswind guard condition is

$$c \leq c_1 \quad (3.18)$$

which may be represented in the form (3.17) by setting

$$L_x = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad L_\theta = -1 \quad \theta^* = c_1 \quad (3.19)$$

### FCP2 - Turning Downwind

The guard conditions for turning downwind is based upon vehicle heading which cannot be used directly as it is not a state variable. Instead, velocities can be used

$$\dot{c} \geq 0 \quad \text{and} \quad \dot{u} \geq -v \quad (3.20)$$

where  $v$  is the nominal velocity. Therefore, the coefficient matrices are

$$L_x = \begin{bmatrix} 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad L_\theta = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad \theta^* = \begin{bmatrix} 0 \\ v \end{bmatrix} \quad (3.21)$$

---

### FCP3 - Downwind

The downwind guard condition is

$$u \leq 0 \quad (3.22)$$

Therefore

$$L_x = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad L_\theta = 1 \quad \theta^* = 0 \quad (3.23)$$

### FCP4 - Late Downwind

The guard condition for leaving the downwind leg is conditioned on the relative position of the runway threshold as indicated in Figure 3.2 and the commencement of a descent

$$u \geq \frac{-c}{\tan(\pi/4)} \quad \text{and} \quad h \geq h_c \quad (3.24)$$

where  $h_c$  is the typical circuit height. The coefficient matrices are

$$L_x = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix} \quad L_\theta = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \quad \theta^* = \begin{bmatrix} \frac{c}{\tan(\pi/4)} \\ h_c \end{bmatrix} \quad (3.25)$$

### FCP5 - Turning Base

This guard condition is heading based similar to FCP2, the same velocity approach is used

$$\dot{u} \leq 0 \quad \text{and} \quad \dot{c} \geq -v \quad (3.26)$$

where  $v$  is the nominal velocity. The coefficient matrices are

$$L_x = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \quad L_\theta = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \quad \theta^* = \begin{bmatrix} 0 \\ v \end{bmatrix} \quad (3.27)$$

### FCP6 - Base

The guard condition for base is

$$c \geq c_2 \quad (3.28)$$

---

Therefore

$$L_x = \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad L_\theta = 1 \quad \theta^* = c_2 \quad (3.29)$$

### FCP7 - Turning Final

This is the final heading based [FCP](#)

$$\dot{u} \geq -v \quad \text{and} \quad \dot{c} \leq 0 \quad (3.30)$$

where  $v$  is the nominal velocity. The coefficient matrices are

$$L_x = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad L_\theta = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \quad \theta^* = \begin{bmatrix} v \\ 0 \end{bmatrix} \quad (3.31)$$

### FCP8 - Final

This is the final [FCP](#) which occurs when the vehicle touches down on the runway. The guard conditions are

$$u \leq 0 \quad \text{and} \quad h \geq 0 \quad (3.32)$$

Therefore

$$L_x = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix} \quad L_\theta = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \theta^* = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (3.33)$$

## 3.2.3 Discrete mode transition

To capture uncertainty in the continuous state and [FCP](#) guard conditions, it is necessary to treat transitions as probabilistic events. The mode transition probability to mode  $j$  at time  $k + 1$  can be written as the 8-dimensional integral

$$p[m(k+1) = j \mid M^k, Z^k] = \int_{\mathbb{R}^8} p[m(k+1) = j \mid M^k, Z^k, x] p[x(k) = x \mid M^k, Z^k] dx \quad (3.34)$$

---

where  $p[m(k+1) = j | M^k, Z^k]$  is the probability of being in mode  $j$  at time  $k+1$  given the mode and measurement histories up to  $k$ ,  $M^k$  and  $Z^k$  respectively, and

$$\int_{\mathbb{R}^8} \bullet dx = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \bullet dx_1 \dots dx_8 \quad (3.35)$$

which represents integration over the entire (8-dimensional) continuous state vector  $x$ .

Equation (3.34) states that the mode transition is conditioned on the previous mode and measurement history in addition to the continuous state history. It is derived by considering a continuous form of the law of total probability which relates the marginal probability of mode transition to the conditional probability of mode transition given a continuous state. To simplify the analysis it is assumed that the mode transition is a Markovian process, therefore the conditional mode transition probability  $p[m(k+1) | M^k, Z^k, x]$  depends only on the mode and continuous state of the previous time step and not their entire history, therefore if the mode at time  $k$  is  $i$

$$p[m(k+1) = j | M^k, Z^k] = \int_{\mathbb{R}^8} \pi_{ij}(x) p[x(k) = x | M^k, Z^k] dx \quad (3.36)$$

where

$$\pi_{ij}(x) = p[m(k+1) = j | m(k) = i, x(k)] \quad (3.37)$$

and is the probability of transition from state  $i$  to state  $j$  conditioned on the continuous state. This is assumed to be a Gaussian Multivariate Cumulative Distribution Function (MVCDF) with mean and covariance evaluated from the FCP guard condition from mode  $i$  to  $j$

$$\pi_{ij}(x) = Pr[L_{x,ij}x + L_{\theta,ij}\theta^* \leq 0 | x] \quad (3.38)$$

It is possible to model the probability of this inequality being satisfied as a Gaussian MVCDF but first the associated Multivariate Probability Density Function (MVPDF) must be found. The random variable in (3.38) is  $\theta^*$  which is the uncertain guard condition state and this may be considered as a Gaussian

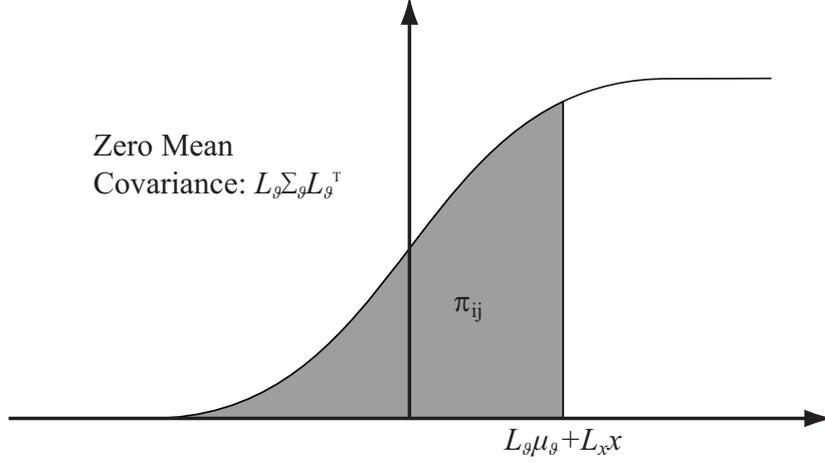


Figure 3.5: Single variable cumulative distribution function for mode transition

**MVPDF** with mean  $\mu_\theta$  and covariance  $\Sigma_\theta$ . Therefore

$$p[L_{x,ij}x + L_{\theta,ij}\theta^* | x] = L_{x,ij}x + L_{\theta,ij}\mathcal{N}_q(\theta^*; \mu_\theta, \Sigma_\theta) \quad (3.39)$$

where  $\mathcal{N}_q$  represents a  $q$ -dimensional **MVPDF** for  $\theta^*$  with mean  $\mu_\theta$  and covariance matrix  $\Sigma_\theta$ . By Theorem 2.4.1 from [Anderson \[1984\]](#), (3.39) can be written

$$p[L_{x,ij}x + L_{\theta,ij}\theta^* | x] = \mathcal{N}_q(\theta^*; L_{x,ij}x + L_{\theta,ij}\mu_\theta, L_{\theta,ij}\Sigma_\theta L_{\theta,ij}^T) \quad (3.40)$$

This defines a distribution with mean  $L_{x,ij}x(k) + L_{\theta,ij}\mu_\theta$  and covariance  $L_{\theta,ij}\Sigma_\theta L_{\theta,ij}^T$ . Figure 3.5 illustrates the one dimensional case of how the transition probability can be calculated with these parameters in a **MVCDF**. Using  $\Phi_q$  as the  $q$ -dimensional zero mean **MVCDF** operator, this gives

$$\pi_{ij}(x) = \Phi_q(L_{x,ij}x + L_{\theta,ij}\mu_\theta, L_{\theta,ij}\Sigma_\theta L_{\theta,ij}^T) \quad (3.41)$$

The second term in the integral of (3.36) is the probability of being in state  $x$  at time  $k$  which can be assumed to be a Gaussian **MVPDF** with mean  $\hat{x}_i(k)$  and covariance  $P_i(k)$  where the subscript  $i$  represents the estimate made by the

---

$i$ th mode Kalman filter. Equation (3.36) can now be written

$$p[m(k+1) = j \mid M^k, Z^k] = \int_{\mathbb{R}^8} \pi_{ij}(x) \mathcal{N}_8(x; \hat{x}_i(k), P_i(k)) dx \quad (3.42)$$

It can be shown (lemmas 1 and 2 in Seah and Hwang [2009]) that this integral represents a Gaussian MVCDF. Therefore the mode transition probability from mode  $i$  to mode  $j$  at time  $k$ , termed  $\gamma_{ij}(k)$ , can be written as

$$\gamma_{ij}(k) = \Phi_q(L_{x,ij}\hat{x}_i(k) + L_{\theta,ij}\mu_\theta, L_{x,ij}P_i(k)L_{x,ij}^T + L_{\theta,ij}\Sigma_\theta L_{\theta,ij}^T) \quad (3.43)$$

### 3.2.4 State Dependent Transition Hybrid Estimation algorithm

Figure 3.6 illustrates how the SDTHE algorithm utilises the mode transition probabilities discussed in the previous section to estimate mode probabilities and continuous states. The following sections explain the functionality of each of the components of Figure 3.6 with more detail being found in Seah and Hwang [2009].

The mode probability ( $\alpha(k)$ ) is used throughout the algorithm and represents the probability of residing in each mode at time  $k$ . In steady state, the elements of  $\alpha$  consists of single unity elements such as

$$\alpha(k) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.44)$$

which represents the 4th mode (Late Downwind). In this condition it acts simply to select which Kalman filter is used at various stages as discussed below. During mode transitions, however a typical value of  $\alpha$  may be

$$\alpha(k) = \begin{bmatrix} 0 & 0.4 & 0.6 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.45)$$

which represents a point during transition between modes 2 and 3 (Downwind Turn and Downwind). In this condition the Kalman filters for these modes are mixed together according to the mode probabilities.

---

## Mixing

At each time step, each of the Kalman filters requires an initial state  $\hat{x}_{i0}(k)$  and covariance matrix  $P_{i0}(k)$ . These are found by combining the estimates made at the previous time step, weighted by the mode probability vector.

$$\hat{x}_{i0}(k) = \sum_{j=1}^8 \hat{x}_j(k) \bar{\alpha}_{ij}(k) \quad (3.46)$$

$$P_{i0}(k) = \sum_{j=1}^8 (P_j(k) + (\hat{x}_j(k) - \hat{x}_{i0}(k))(\hat{x}_j(k) - \hat{x}_{i0}(k))^T) \bar{\alpha}_{ij}(k) \quad (3.47)$$

where

$$\bar{\alpha}_{ij}(k) = \frac{1}{c_i} \gamma_{ji}(k) \alpha_j(k) \quad (3.48)$$

and  $c_i$  is a normalising constant.

These equations have the effect of weighting each Kalman filters contribution based upon the probability of residing in the associated mode.

## Kalman Filters

For each of the discrete modes listed in Table 3.1 the governing equations are made up from a lateral and vertical model defined by their sub-modes. These sub-mode models can be written as one single model of the form

$$x(k+1) = Ax(k) + Ww(k) \quad (3.49)$$

$$z(k+1) = Cx(k) + v(k) \quad (3.50)$$

where  $w(k)$  and  $v(k)$  are the process and measurement noise respectively.

These equations are well suited to direct implementation of a typical discrete time linear Kalman filter as described in Welch and Bishop [2001]. The process noise covariance  $Q$  is modified slightly to accommodate the coefficient matrix  $W$ .

$$Q' = WQW^T \quad (3.51)$$

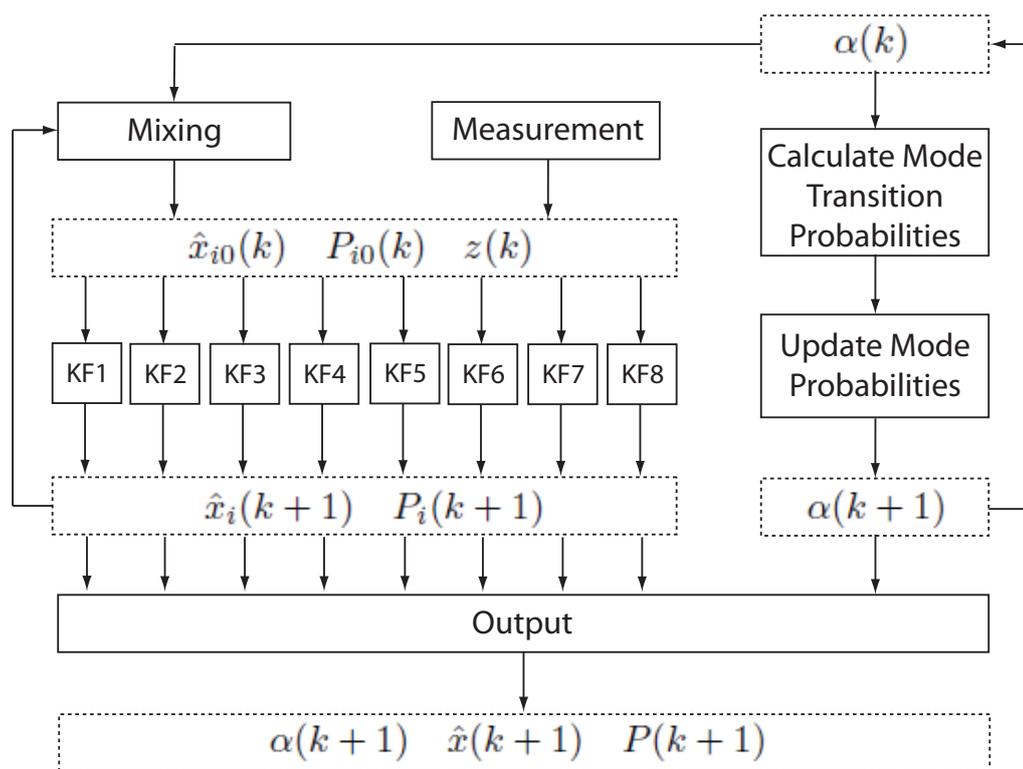


Figure 3.6: Execution logic of the [SDTHE](#) algorithm

---

This modification is made simply to allow for a more intuitive definition of process noise as described in Section 3.2.1.

### Mode probability update

To update the mode probabilities the Kalman filter likelihoods and transitions probabilities are used. The likelihood of the  $i$ th Kalman filter is the probability of the measurement at time  $k + 1$  given the  $i$ th mode is correct at that time. This can be calculated from the distributions of the residuals of the Kalman filter

$$\Lambda_i(k + 1) = \mathcal{N}_3(r_i(k + 1); 0, S_i(k + 1)) \quad (3.52)$$

where  $r$  and  $S$  are the residuals and their covariances respectively.

In addition to the Kalman filter likelihood, the  $i$ th mode can only be correct if a transition has occurred, therefore the current mode and transitions probabilities must be included

$$\alpha'_i(k + 1) = \Lambda_i(k + 1) \sum_{j=1}^8 \gamma_{ji}(k) \alpha_j(k) \quad (3.53)$$

$\alpha'_i(k + 1)$  is the non-normalised probability of the  $i$ th mode at time  $k + 1$ . To obtain the normalised vector  $\alpha$  simply divide by the sum of  $\alpha'$

$$\alpha(k + 1) = \frac{\alpha'(k + 1)}{\sum_{i=1}^8 \alpha'_i(k + 1)} \quad (3.54)$$

### Output

The output state estimate  $\hat{x}(k + 1)$  is calculated as the weighted sum of the Kalman filter estimates.

$$\hat{x}(k + 1) = \sum_{i=1}^8 \hat{x}_i(k + 1) \alpha_i(k + 1) \quad (3.55)$$

---

and its covariance

$$P(k+1) = \sum_{i=1}^8 (P_i(k+1) + (\hat{x}_i(k+1) - \hat{x}(k))(\hat{x}_i(k+1) - \hat{x}(k))^T) \alpha_i(k+1) \quad (3.56)$$

The mode probabilities are not directly useful as outputs from the comprehension system, therefore the most likely mode is output instead. This is simply the index of the maximum value in  $\alpha$ , for example if

$$\alpha(k+1) = \begin{bmatrix} 0 & 0 & 0 & 0.7 & 0.3 & 0 & 0 & 0 \end{bmatrix} \quad (3.57)$$

the mode output would be 4 (Late Downwind).

## 3.3 Results

In order to test the [SDTHE](#) algorithm a MATLAB/Simulink simulation was performed. The algorithm was coded within MATLAB and a Simulink model was created to simulate the measurement of noisy data from a single traffic vehicle.

### 3.3.1 Simulation set up

The nominal flight path of the vehicle is illustrated in [Figure 3.7](#). The vehicle begins abeam the runway beginning the crosswind leg at a nominal velocity of  $40ms^{-1}$  and a height of  $350m$ . The [FCPs](#) illustrated in [Figure 3.7](#) represent the nominal conditions, however as discussed in the previous section these must be modelled as [MVPDFs](#) to take account of variability in the flight path of the vehicle. The nominal [FCPs](#) are the means of the distribution ( $\mu_\theta$ ) and the covariances ( $\Sigma_\theta$ ) can be determined by observation of the typical variation in flight paths.

[Table 3.2](#) shows the mean and covariance information for each of the [FCPs](#). The mean vectors take the form of the [FCP](#) state vector  $\theta^*$  detailed in [Section 3.2.2](#) and the distributions are considered to be decoupled so only the diagonal terms are present in the covariance matrices. The variances for each parameter were tuned to achieve the desired performance based upon the variations in the

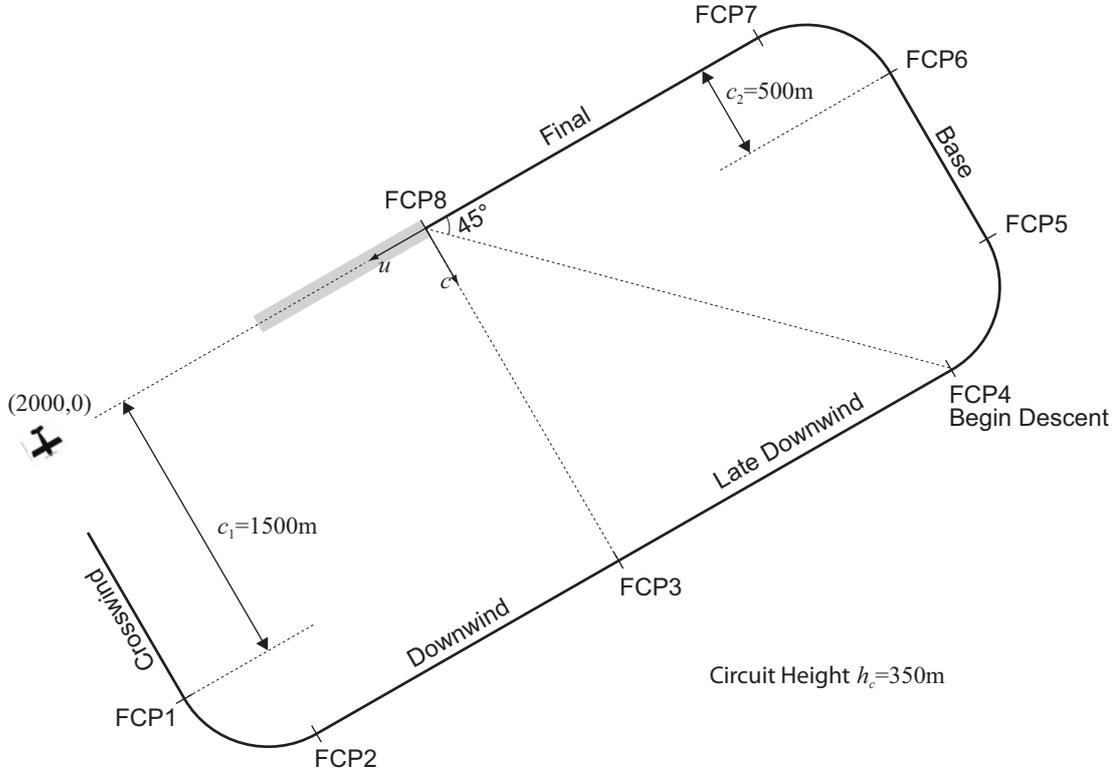


Figure 3.7: Typical circuit dimensions used as nominal flight path

simulated flight path.

A more complex model is used for the vehicle than the linear models used in the [SDTHE](#) algorithm so as to allow the vehicles turn rate to be used to characterise manoeuvres (not required by the filtering process of the [SDTHE](#)).

$$\begin{bmatrix} \hat{u} \\ \hat{c} \\ \hat{h} \\ \psi \end{bmatrix}_{k+1} = \begin{bmatrix} \hat{u} \\ \hat{c} \\ \hat{h} \\ \psi \end{bmatrix}_k + \begin{bmatrix} dt \cos \psi'_k & 0 & 0 \\ dt \sin \psi'_k & 0 & 0 \\ 0 & dt & 0 \\ 0 & 0 & dt \end{bmatrix} \begin{bmatrix} v \\ v_h \\ v'_\psi \end{bmatrix}_k \quad (3.58)$$

where the velocity, vertical velocity and rate of change of heading ( $\begin{bmatrix} v & v_h & v'_\psi \end{bmatrix}_k^T$ ) are set by means of a predefined flight plan. The output of the model which is fed into the [SDTHE](#) includes a small amount of noise which is typically associated

Table 3.2: FCP means and covariances

FCP	Means	Covariances
1	$\mu_\theta = c_1 = 1500m$	$\Sigma_\theta = (100m)^2$
2	$\mu_\theta = \begin{bmatrix} 0 \\ v \end{bmatrix} = \begin{bmatrix} 0ms^{-1} \\ 40ms^{-1} \end{bmatrix}$	$\Sigma_\theta = \begin{bmatrix} (3ms^{-1})^2 & 0 \\ 0 & (2ms^{-1})^2 \end{bmatrix}$
3	$\mu_\theta = 0m$	$\Sigma_\theta = (100m)^2$
4	$\mu_\theta = \begin{bmatrix} \frac{c}{\tan(\pi/4)} \\ h_c \end{bmatrix} = \begin{bmatrix} \frac{c}{350m} \\ \tan(\pi/4) \end{bmatrix}$	$\Sigma_\theta = \begin{bmatrix} (100m)^2 & 0 \\ 0 & (5m)^2 \end{bmatrix}$
5	$\mu_\theta = \begin{bmatrix} 0 \\ v \end{bmatrix} = \begin{bmatrix} 0ms^{-1} \\ 35ms^{-1} \end{bmatrix}$	$\Sigma_\theta = \begin{bmatrix} (2ms^{-1})^2 & 0 \\ 0 & (2ms^{-1})^2 \end{bmatrix}$
6	$\mu_\theta = c_2 = 500m$	$\Sigma_\theta = (100m)^2$
7	$\mu_\theta = \begin{bmatrix} v \\ 0 \end{bmatrix} = \begin{bmatrix} 30ms^{-1} \\ 0ms^{-1} \end{bmatrix}$	$\Sigma_\theta = \begin{bmatrix} (2ms^{-1})^2 & 0 \\ 0 & (2ms^{-1})^2 \end{bmatrix}$
8	$\mu_\theta = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0m \\ 0m \end{bmatrix}$	$\Sigma_\theta = \begin{bmatrix} (10m)^2 & 0 \\ 0 & (10m)^2 \end{bmatrix}$

with Global Positioning Systems (GPSs)

$$\begin{bmatrix} u \\ c \\ h \end{bmatrix}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \hat{u} \\ \hat{c} \\ \hat{h} \\ \psi \end{bmatrix}_k + \mathcal{N}_3 \left( \begin{bmatrix} u \\ c \\ h \end{bmatrix}; 0, \begin{bmatrix} (10m)^2 & 0 & 0 \\ 0 & (10m)^2 & 0 \\ 0 & 0 & (5m)^2 \end{bmatrix} \right) \quad (3.59)$$

### 3.3.2 Test 1 - Nominal flight path

In the first test the traffic vehicle is flown along the nominal path, manoeuvring at each of the FCPs precisely. A plot of the estimated position against the nominal path can be seen in Figure 3.8. Figure 3.9 shows the estimated discrete modes as time progresses. It should be noted that the ninth discrete mode is not actively estimated by the SDTHE, it is simply used to denote that FCP8 has been reached (the vehicle has landed). It can be seen from Figure 3.9 that all mode transitions were detected within 10s with most taking less than 5s to detect. Finally, Figure 3.10 shows the errors in the continuous state estimate verses true state. The measured speed in Figure 3.10 is calculated directly from GPS position by first order differencing to provide an indication of the improved accuracy. The discrete

---

mode changes are marked on Figure 3.10 to highlight the performance differences between the Kalman filters.

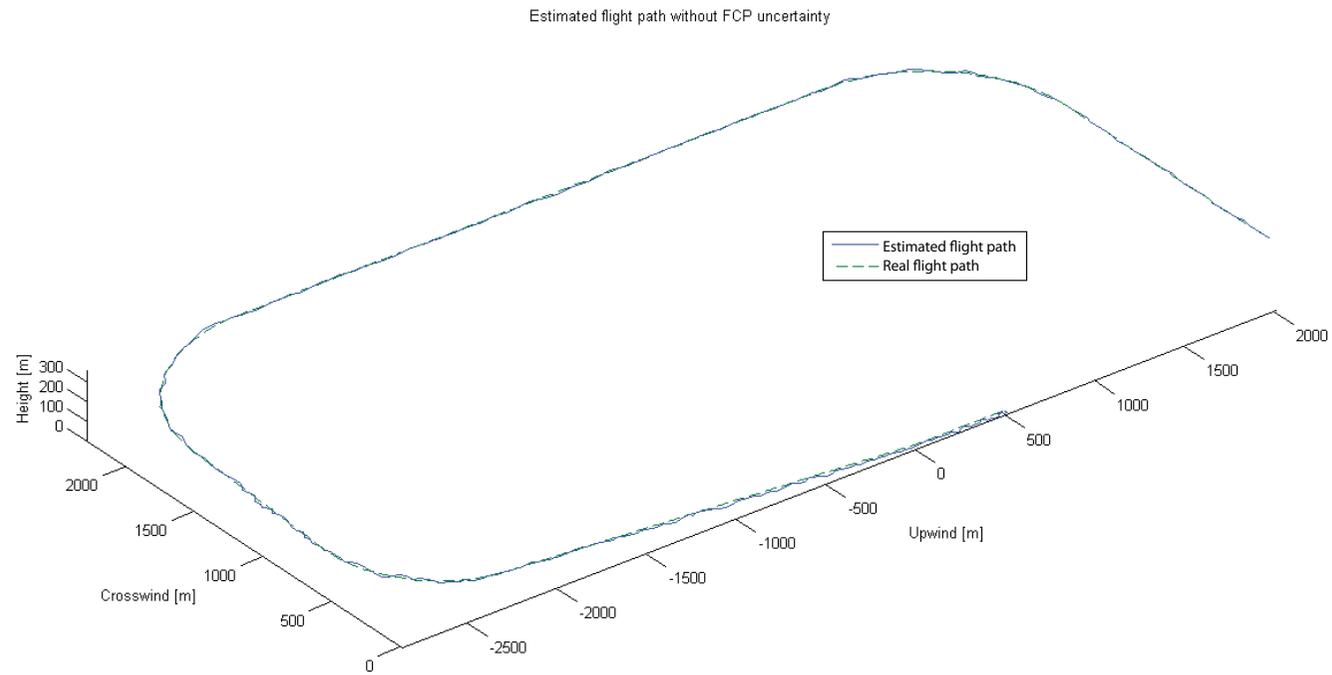


Figure 3.8: Plot of the estimated flight path without FCP uncertainty

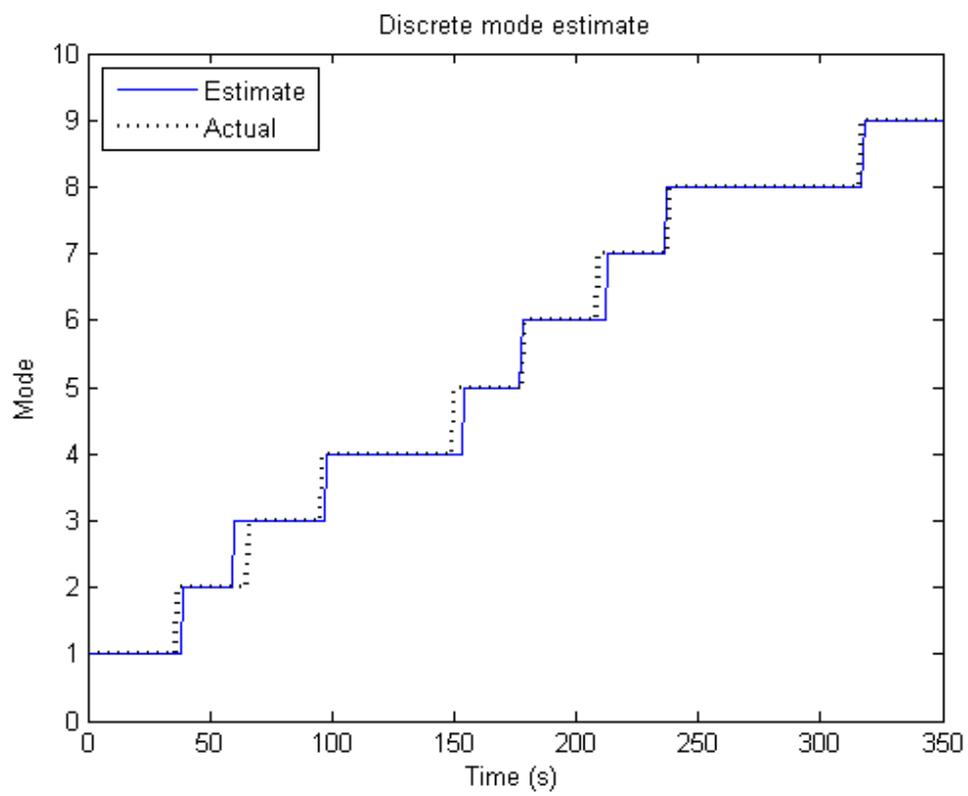


Figure 3.9: Plot of the discrete mode estimation without FCP uncertainty

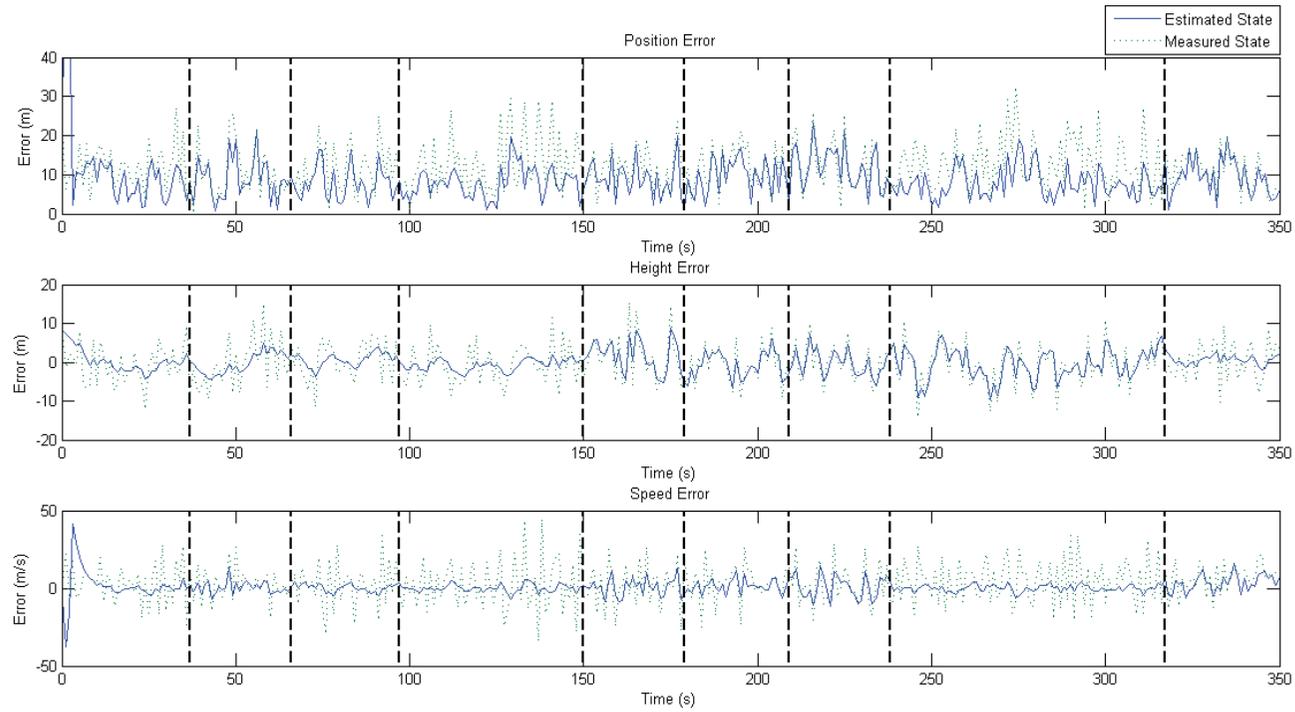


Figure 3.10: Plot of the errors in the estimated continuous states without [FCP](#) uncertainty

---

### 3.3.3 Test 2 - FCP uncertainty

In this test the traffic vehicle flies a path which exhibits a typical level of uncertainty in its FCPs, associated with a pilot not flying precisely as expected (e.g. turning later than expected). Random variations in heading and FCP position were used, with variances of  $5^\circ$  and  $100m$  respectively. The heading variance is representative of the typical accuracy with which a human pilot is able to maintain a constant heading. Position variance is of the order typically experienced at a small uncontrolled airfield, this is around 10% of the circuit dimensions. It can be seen from Figure 3.11 that the vehicle begins on the correct path but after initiating its turn onto the downwind leg too early continues to exhibit an abnormal path. Figures 3.12 and 3.13 show the discrete mode transitions and continuous state errors.

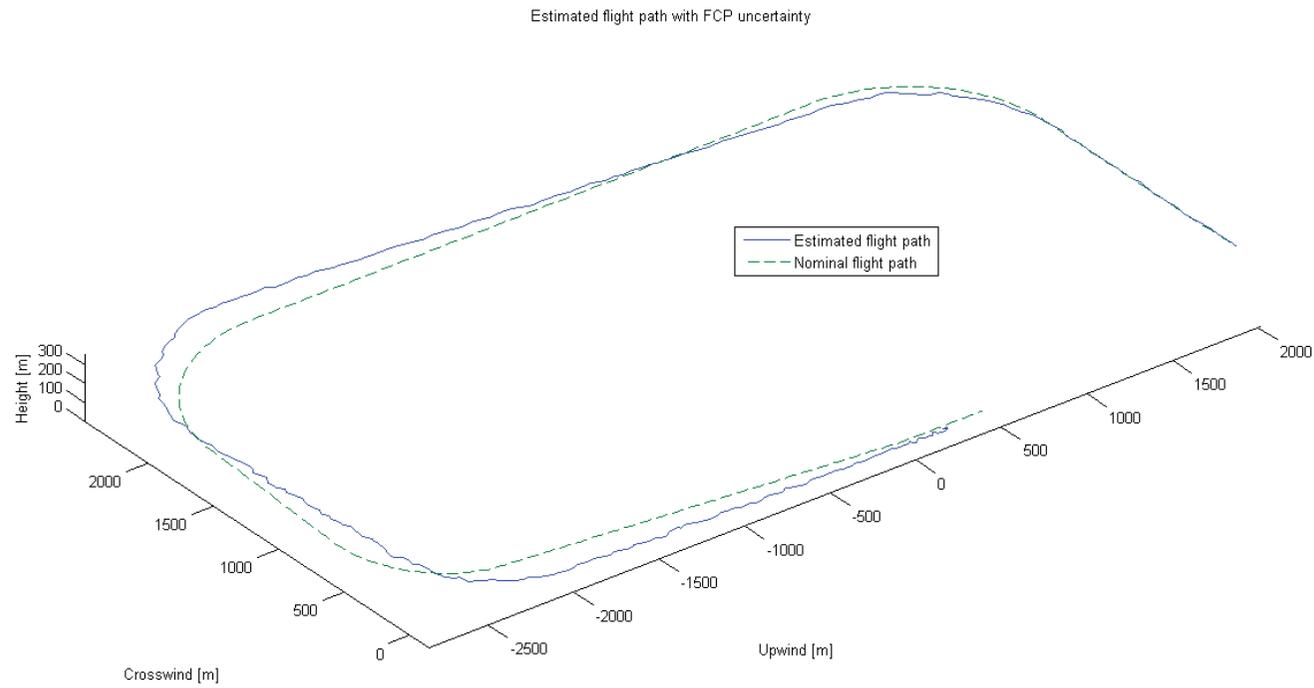


Figure 3.11: Plot of the estimated flight path with **FCP** uncertainty

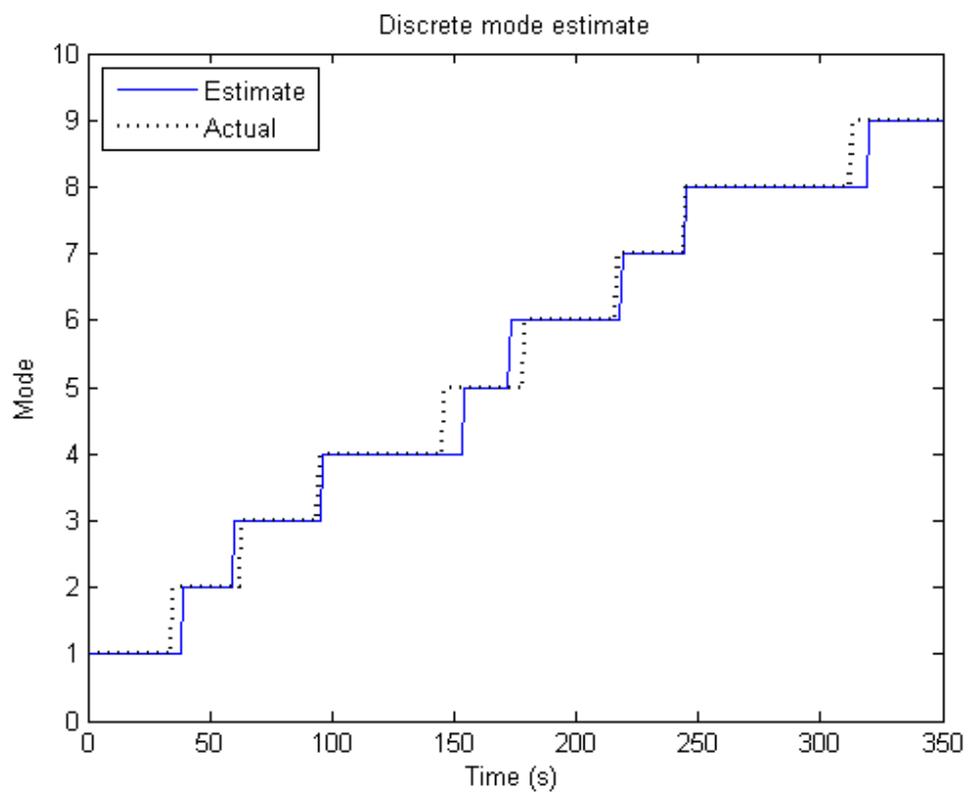


Figure 3.12: Plot of the discrete mode estimation with [FCP](#) uncertainty

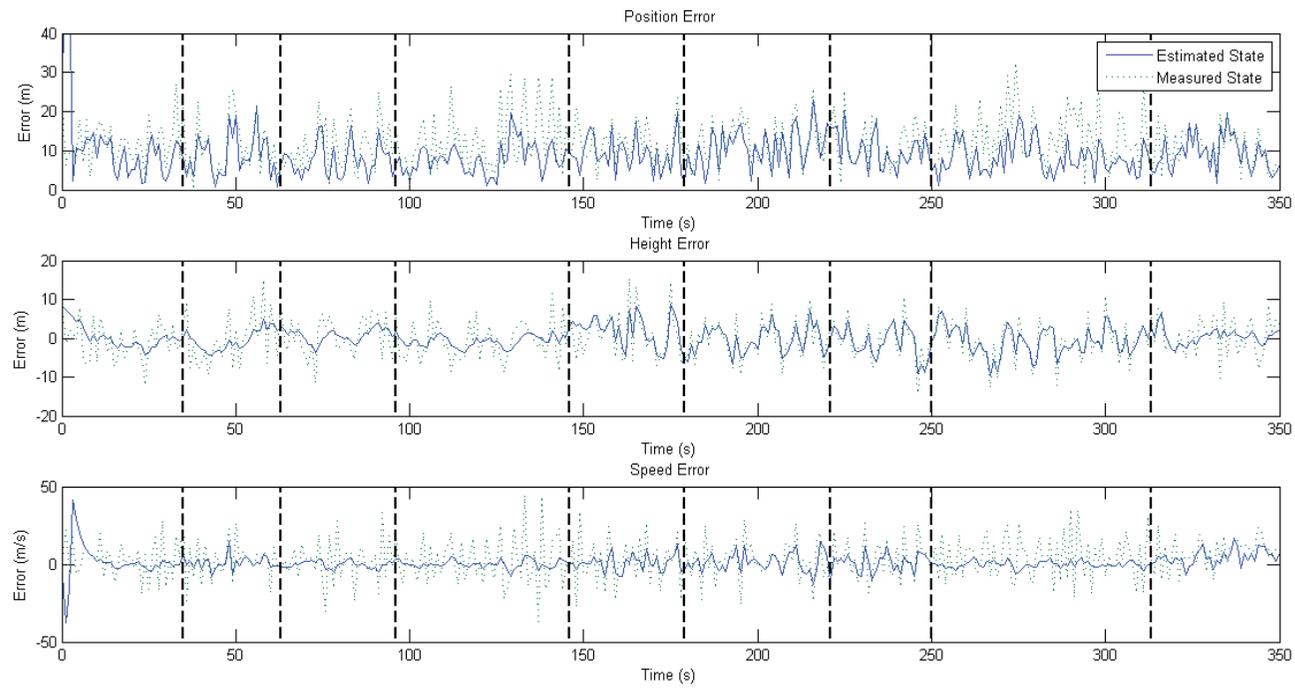


Figure 3.13: Plot of the errors in the estimated continuous states with FCP uncertainty

Table 3.3: Sensitivity of **SDTHE** position estimation to parameter variation

$\mu \backslash \Sigma$	90%	100%	110%
90%	9.17m	9.18m	9.18m
100%	9.43m	9m	9.14m
110%	9.47m	9.48m	9.47m

Table 3.4: Sensitivity of **SDTHE** discrete mode estimation to parameter variation

$\mu \backslash \Sigma$	90%	100%	110%
90%	11.1%	11.4%	11.4%
100%	5.7%	5.98%	5.98%
110%	5.7%	5.98%	5.98%

### 3.3.4 Sensitivity study

The parameters listed in Table 3.2 would likely be estimated from recorded data, a process which is subject to its own sources of error. To test the robustness of the **SDTHE** algorithm to errors in these parameters a sensitivity study was performed. This study varied the means and covariances listed in Table 3.2 from 90% to 110% whilst repeating Test 1 described above.

Table 3.3 illustrates the effect of this variation on the Root Mean Square (**RMS**) position estimation error. It is clear that **SDTHE** position estimation is relatively insensitive to variation in parameters, with variation in **FCP** means having the greatest effect.

Table 3.4 illustrates the effect of parameter variation on discrete mode estimation. The metric used to measure discrete mode estimation performance is the percentage of the test that the incorrect mode is estimated. The **SDTHE** algorithm with unmodified parameters estimates the incorrect mode for 5.98% of the flight, this is visible as a discrepancy between the two lines in Figure 3.9. As with position estimation, the discrete mode estimation is insensitive to variation in **FCP** covariance, however more sensitive to variations in **FCP** mean.

---

### 3.3.5 Discussion

The previous sections have presented results for nominal and typical off-nominal flight paths, in addition to a parameter sensitivity study. Analysis of Figures 3.9 and 3.12 shows the power of the SDTHE algorithm for predicting the discrete mode transitions. All transitions are detected within 10s of the actual transition occurring, with many detections being significantly faster. Occasionally, transitions are detected before they actually occur, for example the second transition in figure 3.12, this suggests that the variances governing these transitions may be too large. It is believed, however, that the variances used give the best performance over a wide range of uncertain flight paths.

Figures 3.10 and 3.13 show the benefits of weighting Kalman filters based upon mode probabilities. When the vehicle is in the CV and CH modes a more accurate estimate of speed and height is possible by using Kalman filters which reject changes in these variables. It can be seen that there is little improvement in the position estimation over the GPS measurement, this is due to the fact that the noise power of a GPS signal is small and therefore little improvement can be made. Speed estimation, however, is greatly improved over that obtained by first order differencing of position. An accurate estimation of speed is important for both discrete mode estimation and to enable initial flight path projection.

A sensitivity study has shown the SDTHE algorithm to be insensitive to variations in the FCP covariance parameters. Discrete mode estimation, however, is relatively sensitive to variations in FCP means. When estimating the FCP parameters from data, therefore, greater care should be taken to accurately quantify the mean values so as to ensure best performance from the algorithm.

## 3.4 Summary

This chapter has considered the problem of traffic state comprehension by an autonomous UAS operating in the terminal area. Assuming imperfect position information is available from on-board sensors, the continuous state and discrete mode of the vehicle are estimated. Referring to Section 2.4.4 it is clear that the requirements for comprehension have been met as both the separation and relative

---

trajectory of traffic are directly available from the continuous state estimate. The intentions of the traffic and the hazard posed may both be inferred from the discrete mode information, this is explained in the next chapter.

A **CCS** has been defined to provide continuous state information for all vehicles in a convenient reference frame. The problem of coupled continuous state and discrete mode estimation has been discussed and overcome by the use of the **SDTHE** algorithm. This algorithm was designed as an aid to **ATC** tracking vehicles on well defined routes, utilising discrete mode estimation to improve tracking accuracy. This chapter has extended the **SDTHE** algorithm to a loosely defined path governed by standards and heuristics which is dependant on the performance of the vehicle and is represented in the **CCS** (rather than global coordinates). The discrete modes associated with this visual circuit pattern have been defined with respect to aircraft performance parameters and the associated **FCP**s guard conditions have been derived. The discrete mode estimate provides an **UAS** with the ability to infer future intent of the vehicle based upon the standard procedures, further extending the algorithm from the **ATC** environment.

The ability of the **SDTHE** algorithm to provide filtered continuous state information and correctly determine discrete mode transitions based solely on noisy position measurements has been demonstrated. This allows an autonomous **UAS** to fully comprehend the present state of traffic aircraft allowing the projection element of situation awareness to be undertaken.

The next chapter builds on this comprehension of traffic to provide a projection system for an autonomous **UAS** in the terminal area.

# Chapter 4

## Projection

### 4.1 Introduction

The previous chapter discussed the comprehension of the current state of a traffic vehicle as required in Section 2.4.4. This chapter extends this by performing projection of the comprehended state to some future time horizon. Traditional collision avoidance algorithms perform a basic level of projection by assuming short term continuation of the currently observed state. For example, TCAS estimates the time to collision between two vehicles under the assumption that the observed relative velocity remains constant [Kuchar and Drumm, 2007]. This approach is sufficient for emergency collision avoidance in free airspace (away from the terminal area) where time to collision is low and the potential for vehicles to manoeuvre is small.

If this projection method is used for vehicles which do not pose an imminent risk of collision, small errors in the relative velocity measurement lead to large prediction errors due to the increased projection times involved. Additionally, if the vehicle is operating in the terminal area it is unlikely to maintain a constant velocity for any length of time, making the constant velocity assumption invalid.

This chapter details a novel projection methodology for an autonomous UAS based upon the hybrid comprehension method discussed in the previous chapter. With knowledge of the current discrete mode (from comprehension) it is possible to infer the future trajectory of the vehicle based on the traffic circuit discussed

---

in Section 3.1, including all future manoeuvres. It can not be guaranteed, however, that the vehicle will accurately track the nominal path. The uncertainties associated with both navigation errors and the ability of the vehicle to conduct manoeuvres earlier or later than expected must be quantified so as to allow the worst case conditions to be considered.

The next section discusses the projection along the nominal flight path, assuming the traffic behaves exactly as expected, to determine the mean position at some future time. Section 4.3 covers the sources of uncertainty in projection and how they are used to calculate a covariance matrix which quantifies both continuous and discrete uncertainties. Finally, Section 4.4 presents results of the projection system in isolation to demonstrate its performance.

## 4.2 Nominal flight path projection

Projection of the nominal future states of traffic vehicles may be abstracted into two distinct stages, short and long term. Short term projection assumes a continuation of the current state until the next FCP is reached in a similar way to that used by TCAS. When the projected path reaches a FCP, the current vehicle state can provide very little information about the future path. At this point long term projection takes over by using a simple vehicle model to capture its dynamics and assuming the nominal circuit path will be flown. The assumption holds, provided the vehicle has been positively identified as being in the circuit at the comprehension stage. Once established in the circuit (or more generally, any published arrivals procedure), a vehicle must follow the associated procedures in order to present predictable behaviour to any other vehicles [Civil Aviation Authority, 2009a].

### 4.2.1 Short term prediction

Short term projection is similar to the function of traditional collision avoidance systems such as TCAS. The current state of the traffic is known and the velocity vector is used to predict future positions. For the unaccelerated (CV, CH and

---

CD) flight modes, this prediction is given as

$$\begin{bmatrix} u \\ c \\ v \\ h \\ v_h \end{bmatrix}_{t_{ST}} = \begin{bmatrix} 1 & 0 & t \cos \psi'_0 & 0 & 0 \\ 0 & 1 & t \sin \psi'_0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & t \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ c \\ v \\ h \\ v_h \end{bmatrix}_0 \quad (4.1)$$

where  $t$  is the prediction time measured in seconds,  $\psi'_0 = \arctan\left(\frac{\dot{c}_0}{\dot{u}_0}\right)$  and  $v_0 = \sqrt{\dot{u}_0^2 + \dot{c}_0^2}$ . The subscript  $ST$  indicates that this is a short term prediction equation.

For the accelerated CA and CT flight modes, it is possible to estimate the acceleration, or rate of turn, and include this in the prediction; (4.2) and (4.3) respectively

$$\begin{bmatrix} u \\ c \\ v \\ h \\ v_h \end{bmatrix}_{t_{ST}} = \begin{bmatrix} 1 & 0 & t \cos \psi'_0 & 0 & 0 & 0 \\ 0 & 1 & t \sin \psi'_0 & 0 & 0 & 0 \\ 0 & 0 & 1 & t & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & t \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ c \\ v \\ a \\ h \\ v_h \end{bmatrix}_0 \quad (4.2)$$

where  $a_0 = \sqrt{\ddot{u}_0^2 + \ddot{c}_0^2}$ .

$$\begin{bmatrix} u \\ c \\ v \\ \psi' \\ h \\ v_h \end{bmatrix}_{t_{ST}} = \begin{bmatrix} 1 & 0 & t \cos \psi'_{t-1_{ST}} & 0 & 0 & 0 & 0 \\ 0 & 1 & t \sin \psi'_{t-1_{ST}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & t & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & t \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ c \\ v \\ \psi' \\ v_{\psi'} \\ h \\ v_h \end{bmatrix}_0 \quad (4.3)$$

where

$$v_{\psi'_0} = \frac{\ddot{c}_0 \dot{u}_0 - \dot{c}_0 \ddot{u}_0}{\dot{u}_0^2} \sec^2\left(\frac{\dot{c}_0}{\dot{u}_0}\right) \quad (4.4)$$

It is important to use separate prediction equations for different flight modes

---

as the continuous state estimate will contain small errors in acceleration terms even in unaccelerated modes. Since projection of the current state does not benefit from any further measurements, these errors would be magnified as the prediction horizon increases. Using (4.1) when the current mode is unaccelerated helps mitigate this problem.

The prediction time  $t$  may be increased until the next FCP is reached. Determination of the point when the FCP is reached is more complex than during comprehension due to the inherent uncertainties in the projection process, this will be discussed later.

## 4.2.2 Long term prediction

Once the projection time  $t$  places the vehicle past its next FCP, little if any of its current state information is of use to aid further prediction. At this point the only means of continuing projection forward is to make the assumption that the vehicle will follow the nominal circuit pattern. Vehicles observed to be in the circuit must follow these standardised procedures to enable safe operation to occur [Civil Aviation Authority, 2009a]. Vehicles in the vicinity which are not in the circuit have an obligation to remain clear and if they fail to do so this becomes an emergency collision avoidance situation which is beyond the scope of this thesis, therefore only circuit traffic is considered.

To perform this long term projection, it is necessary to have a rudimentary model of the vehicle dynamics, given in (4.5)

$$\begin{bmatrix} u \\ c \\ v \\ \psi' \\ h \end{bmatrix}_{t_{LT}} = \begin{bmatrix} 1 & 0 & dt \cos \psi'_{t-1_{LT}} & 0 & 0 \\ 0 & 1 & dt \sin \psi'_{t-1_{LT}} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ c \\ v \\ \psi' \\ h \end{bmatrix}_{t-1_{LT}} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ dt & 0 & 0 \\ 0 & dt & 0 \\ 0 & 0 & dt \end{bmatrix} \begin{bmatrix} a \\ v_{\psi'} \\ v_h \end{bmatrix}_{DSn} \quad (4.5)$$

where  $dt$  is the time step, the subscript  $LT$  represents long term projection.

The control vector  $\begin{bmatrix} a & v_{\psi'} & v_h \end{bmatrix}_{DSn}^T$  represents the acceleration and velocities of the vehicle in the  $n$ th discrete state. For example in flight modes CA and CD

---

it may take the value  $\begin{bmatrix} (-0.5ms^{-2}) & 0 & (-1ms^{-1}) \end{bmatrix}^T$ . To determine appropriate values for this vector in each future discrete state it is necessary for the **UAS** to have some knowledge of the aircraft type it is observing. With this information it is possible to select which model is most suited. Typical models may be ‘microlight’, ‘twin-engined piston’ or ‘large civil’. These models may be refined in real-time by observation of manoeuvres as they occur at the comprehension level.

During long term projection the transition between discrete states is trivial as the vehicle is following the nominal circuit path and will therefore intersect the **FCP**s exactly.

### 4.3 Projection uncertainty

The previous section introduced projection of the traffic vehicle along the nominal flight path, providing the mean position at some future time. This is subject to a large degree of uncertainty, however, as a vehicle may not maintain the exact speeds and heading estimated by the **UAS** during the comprehension stage and may transition between discrete states away from the **FCP**s nominal positions. To account for these uncertainties, the position of the vehicle can be thought of as a **MVPDF** with mean as defined by the nominal flight path and covariance defined by the uncertainties.

Throughout the majority of section it is assumed that the accumulation of errors is not bounded by any form of guidance system, therefore they may grow indefinitely. Bounding uncertainty is considered, however, with regard to assuring the vehicle reaches the runway and a similar approach can be used to include guidance laws if required.

#### 4.3.1 Continuous linear uncertainty

Uncertainties arising from the vehicles deviations from the nominal flight path are continuous in nature. They can be characterised as uncorrelated variances in

---

the aircrafts velocity

$$\Sigma_v = \begin{bmatrix} \sigma_{v_x}^2 & 0 & 0 \\ 0 & \sigma_{v_y}^2 & 0 \\ 0 & 0 & \sigma_h^2 \end{bmatrix} \quad (4.6)$$

where  $\sigma_{v_x}^2$ ,  $\sigma_{v_y}^2$ ,  $\sigma_h^2$  are the variances in velocity along the longitudinal, lateral and vertical aircraft axes respectively. Lateral velocity uncertainty may be difficult to define and can therefore be written in terms of a heading uncertainty

$$\sigma_{v_y} = \hat{v} \tan \sigma_\psi \quad (4.7)$$

where  $\hat{v}$  is the nominal velocity and  $\sigma_\psi$  is the variance in heading. This assumes that  $\sigma_{v_x}$  is small in proportion to  $\hat{v}$ . Also for small values of  $\sigma_\psi$ ,  $\sigma_{v_x} = \sigma_v$ , the velocity variance along the nominal flight path; and  $\tan \sigma_\psi = \sigma_\psi$ . Therefore (4.6) may be written

$$\Sigma_v = \begin{bmatrix} \sigma_v^2 & 0 & 0 \\ 0 & (\hat{v}\sigma_\psi)^2 & 0 \\ 0 & 0 & \sigma_h^2 \end{bmatrix} \quad (4.8)$$

To calculate positional uncertainty, it is necessary to integrate the velocity uncertainties along the flight path and then transform into the CCS. It can be assumed that the velocity uncertainties remain constant throughout any given discrete state but they may change between states. Due to the uncorrelated nature of the continuous uncertainty in each discrete state it is possible to perform this integration as a piecewise summation. Figure 4.1 illustrates a path which has been deconstructed in to its constituent discrete states. The traffic is currently in discrete state  $i$ , and at projection time  $t$  is nominally in discrete state  $j$ , the positional covariance, aligned with the aircraft axis is

$$\Sigma'(t) = \begin{bmatrix} \sigma_x(t)^2 & 0 & 0 \\ 0 & \sigma_y(t)^2 & 0 \\ 0 & 0 & \sigma_z(t)^2 \end{bmatrix} = \Sigma_{v_i} t_{i+1}^2 + \left( \sum_{n=i+1}^j \Sigma_{v_n} (t_{n+1} - t_n)^2 \right) + \Sigma_{v_j} (t - t_j)^2 \quad (4.9)$$

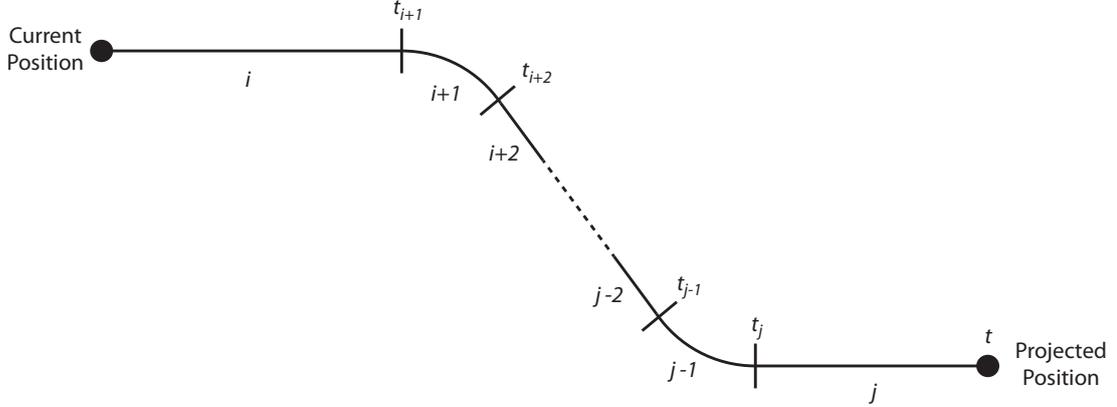


Figure 4.1: Piecewise integration of continuous uncertainty

where  $\Sigma_{v_n}$  is the  $n$ th discrete state velocity covariance defined in (4.8). The first term represents the accumulation of error between the current position and the first discrete transition (at  $t_{i+1}$ ), the summation represents the accumulation of errors for all intermediate discrete states and the final term represents the contribution from the state at projection time  $t$  (from  $t_j$  to  $t$ ).

In order to transform (4.9) into the CCS, it is necessary to rotate the lateral and longitudinal terms by the heading of the vehicle (relative to the runway direction),  $\psi'(t) = \arctan\left(\frac{\dot{c}(t)}{\dot{u}(t)}\right)$ . This approach implicitly linearises the nominal flight path at each projection time  $t$ , which may be an invalid assumption under some circumstances and a method will be presented in the next section to overcome this problem. To rotate a MVPDF, its covariance must be transformed by

$$\Sigma(t) = R\Sigma'(t)R^T \quad (4.10)$$

where  $R$  is the rotation matrix

$$R = \begin{bmatrix} \cos \psi'(t) & -\sin \psi'(t) & 0 \\ \sin \psi'(t) & \cos \psi'(t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.11)$$

This transformation is analogous to multiplication by the square of the rotation matrix so as to maintain the correct dimensionality of  $\Sigma$ , see Theorem 2.4.1

---

from [Anderson \[1984\]](#). Performing the transformation in (4.10) yields

$$\Sigma(t) = \begin{bmatrix} \sigma_u(t)^2 & \sigma_{uc}(t)^2 & 0 \\ \sigma_{uc}(t)^2 & \sigma_c(t)^2 & 0 \\ 0 & 0 & \sigma_h(t)^2 \end{bmatrix} \quad (4.12)$$

where  $\sigma_u$  and  $\sigma_c$  are the upwind and crosswind positional variances respectively and  $\sigma_{uc}$  is their covariance, given by

$$\begin{aligned} \sigma_u(t)^2 &= (\sigma_x(t) \cos \psi'(t))^2 + (\sigma_y(t) \sin \psi'(t))^2 \\ \sigma_c(t)^2 &= (\sigma_x(t) \sin \psi'(t))^2 + (\sigma_y(t) \cos \psi'(t))^2 \\ \sigma_{uc}(t)^2 &= (\sigma_y(t)^2 - \sigma_x(t)^2) \sin \psi'(t) \cos \psi'(t) \end{aligned} \quad (4.13)$$

Equation (4.12) together with the nominal flight path projection equations given in (4.2) provide a means of projecting the continuous states of a traffic vehicle under the assumption that the path is locally linear. The following sections will remove this linear restriction and introduce discrete uncertainties such as in the location of [FCPs](#).

### 4.3.2 Continuous curvilinear uncertainty

The rotated covariance matrix given in (4.12) aligns the [MVPDF](#) with the nominal flight path at projection time  $t$ . This assumes that the radius of curvature of the flight path is large when compared with the longitudinal variance  $\sigma_x(t)$ . A typical turn may have a radius in the order of  $1000m$ , therefore linearised projection will perform well for longitudinal uncertainties below  $100m$ . Since  $\sigma_x(t)$  is an increasing function with  $t$  this implies that linear projection is appropriate at small projection times and for all straight (or near straight) paths. [Figure 4.2](#) illustrates the error that arises if linear projection is applied to a curved nominal flight path at a high projection time. Intuitively, this error occurs because a vehicle traveling at a higher than nominal speed travels further along the nominal trajectory, not along its tangent. This error is mitigated by introducing curvilinear projection, [Figure 4.3](#).

Curvilinear projection uses the aircraft axis aligned positional uncertainties

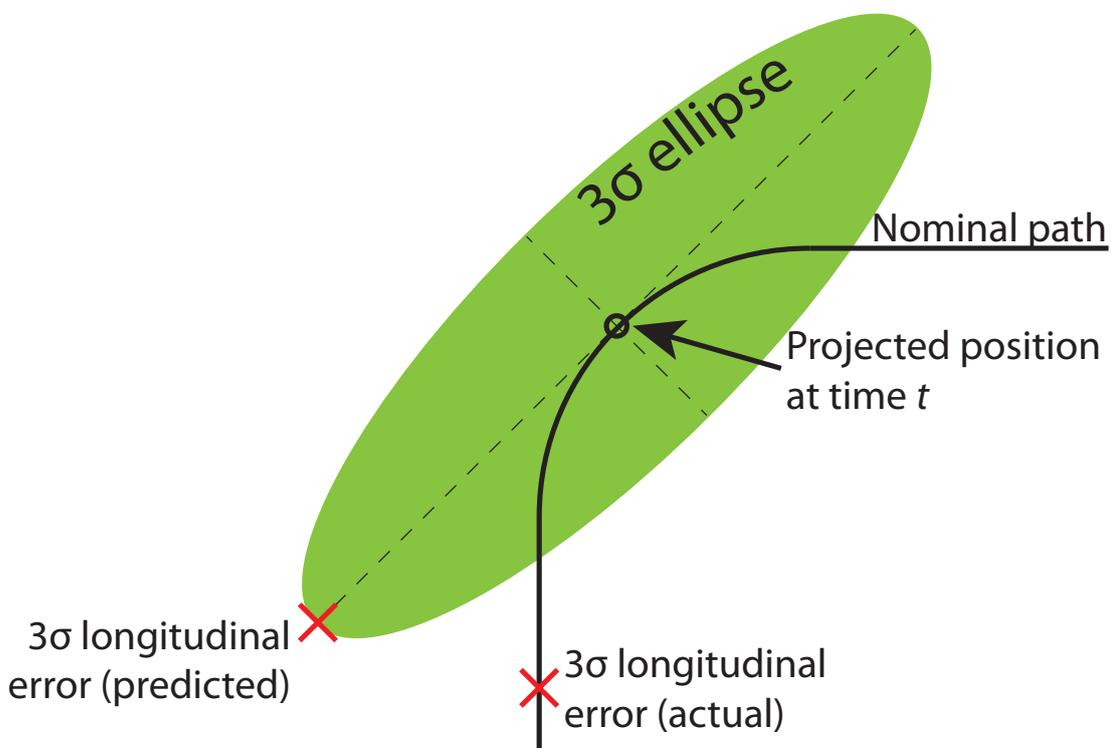


Figure 4.2: Linearisation error during curved flight

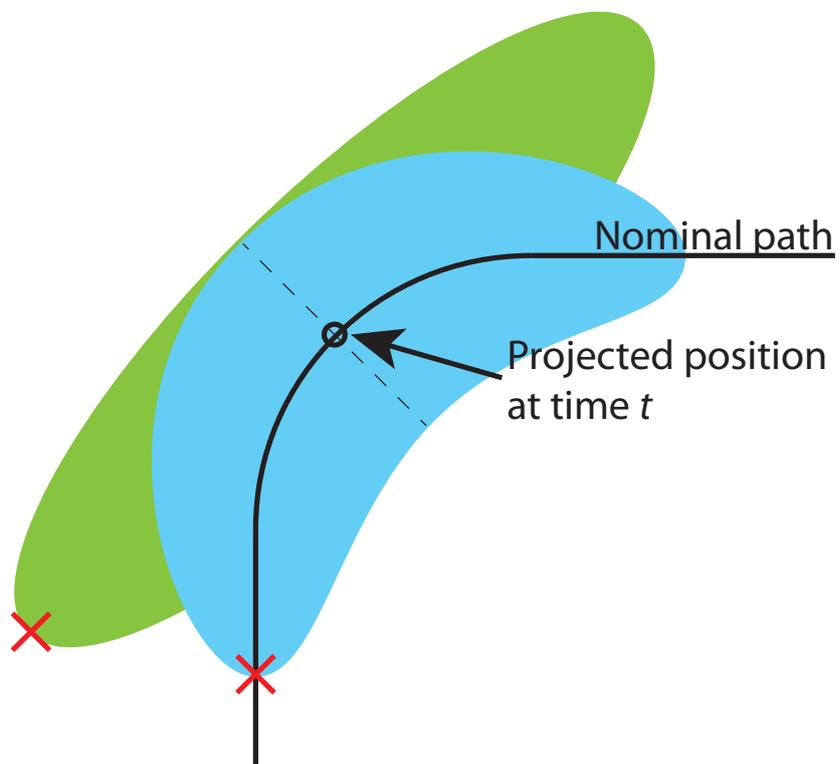


Figure 4.3: Curvilinear projection illustration

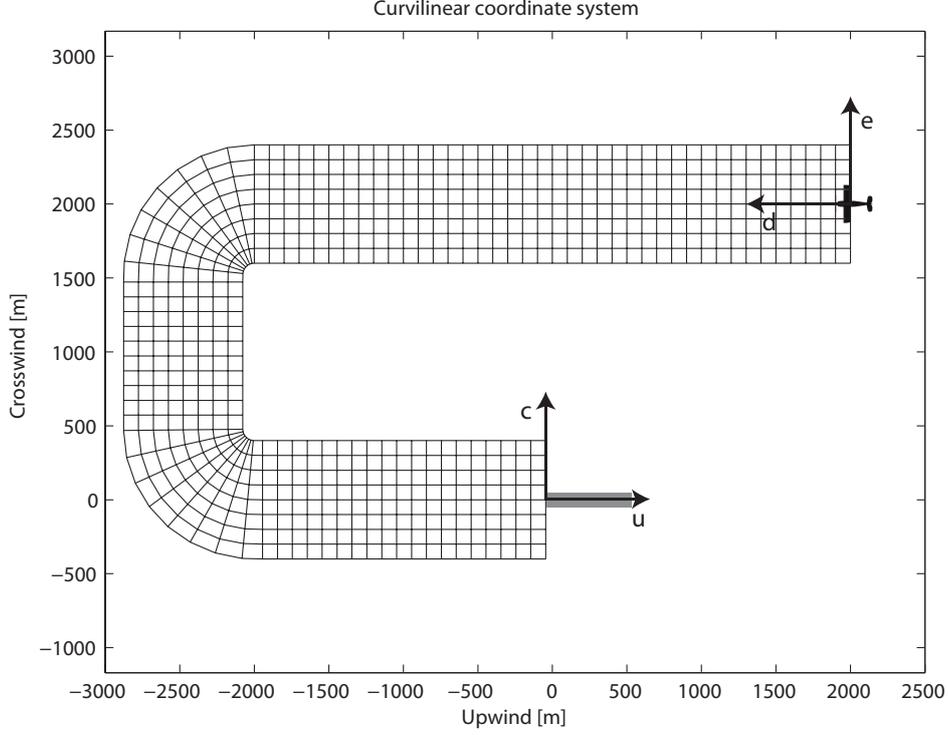


Figure 4.4: Curvilinear coordinate system

given in (4.9) and assigns them to coordinates along and normal to the nominal path (distance and crosstrack error axes) as illustrated in Figure 4.4.

$$\Sigma_{CL}(t) = \begin{bmatrix} \sigma_d(t)^2 & 0 & 0 \\ 0 & \sigma_e(t)^2 & 0 \\ 0 & 0 & \sigma_h(t)^2 \end{bmatrix} = \Sigma'(t) \quad (4.14)$$

where  $\sigma_d(t)^2$  and  $\sigma_e(t)^2$  are the distance and crosstrack error variances respectively and the subscript  $CL$  represents curvilinear coordinates.

To represent the nominal path in curvilinear coordinates, an additional parameter is required, the nominal distance travelled is calculated by

$$d(t) = \int_0^t v(\tau) d\tau \quad (4.15)$$

where  $t$  is the projection time. No calculation of nominal crosstrack error is

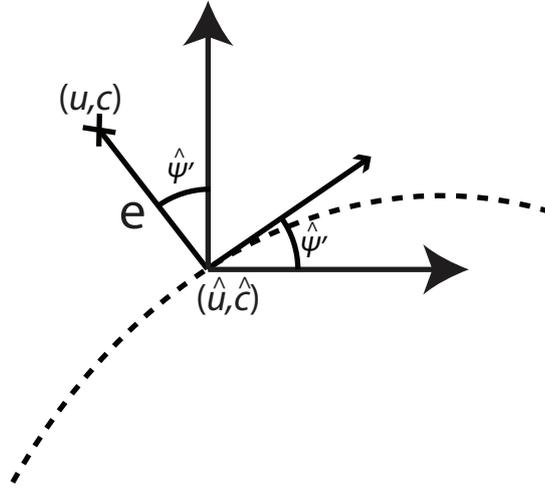


Figure 4.5: Transformation from curvilinear to [CCS](#) coordinates

required as this is zero by definition.

Equation (4.15) has the effect of defining the origin of the curvilinear coordinate system to be the vehicle's present position and extending the distance axis out along the nominal path for all future projection times. This introduces a problem when conducting further analysis on the projected position (such as path planning for the [UAS](#)) because each traffic vehicle possesses a unique coordinate system defined by its present position and there is no way to characterise the position of the [UAS](#) in the same frame. To overcome this, the projected position must be transformed back to the [CCS](#), illustrated in Figure 4.5 and given by

$$\begin{bmatrix} u(t) \\ c(t) \end{bmatrix} = \begin{bmatrix} \hat{u}(d(t)) \\ \hat{c}(d(t)) \end{bmatrix} + e(t) \begin{bmatrix} -\sin(\hat{\psi}'(d(t))) \\ \cos(\hat{\psi}'(d(t))) \end{bmatrix} \quad (4.16)$$

where  $\hat{u}(d(t))$ ,  $\hat{c}(d(t))$  and  $\hat{\psi}'(d(t))$  are the nominal upwind and crosswind positions and runway relative heading at the distance  $d(t)$ .

It is clear from Figure 4.3 that in the [CCS](#) the curvilinear projection distribution is non-Gaussian as it has been mapped to a curved path. This presents an added challenge to further analysis as operations such as determination of the cumulative distribution (which are trivial for Gaussians) are much more complex. However the following sections introduce both discrete and bounded uncertainty which further increase the non-Gaussian nature of the distribution whilst greatly

---

increasing its accuracy, therefore the added complexity is necessary.

### 4.3.3 Discrete uncertainty

The uncertainties discussed in the previous sections pertain to the build up of small navigational errors over time which is continuous in nature as it occurs at all projection times. Discrete actions are also subject to uncertainty during projection as the exact time at which they will occur is unknown to the UAS. Within the terminal area the most critical example of these discrete uncertainties is at the FCPs, when the aircraft will be manoeuvred by the pilot. For example, the decision of a pilot to commence a turn 5s earlier than anticipated by the UAS would result in significantly different future trajectories. During terminal area operations separation of traffic by height is rarely performed therefore discrete transitions in height are ignored and the problem is treated as two dimensional.

Figure 4.6 illustrates the problem this poses for projection; for clarity the continuous uncertainties discussed in the previous sections are ignored. In this example, a vehicle positioned at  $(0, 500)m$  heading in the crosswind direction would be expected to turn after reaching  $(1300, 500)m$ , the nominal path in figure 4.6. However a worst case variation of  $\pm 800m$  in the FCP may be observed in circuit traffic, represented by the early and late turns. The dashed lines represent contours of constant projection time and show how, should be vehicle turn early, it will progress further in the upwind direction than the nominal and late turn cases.

To capture discrete uncertainty, it is first characterised as a univariate Gaussian PDF

$$d_i \sim \mathcal{N}(\hat{d}_i, \tilde{d}_i^2) \quad (4.17)$$

where  $d_i$  is the distance along the nominal path that the  $i$ th discrete transition occurs,  $\hat{d}_i$  is its mean value and  $\tilde{d}_i^2$  its variance. Figures 4.7 and 4.8 illustrate two effects caused by the inclusion of discrete uncertainty in this way. Figure 4.7 shows how a turn made later than the mean will increase the vehicles crosstrack error and also delay its progress along the nominal path. These errors in distance

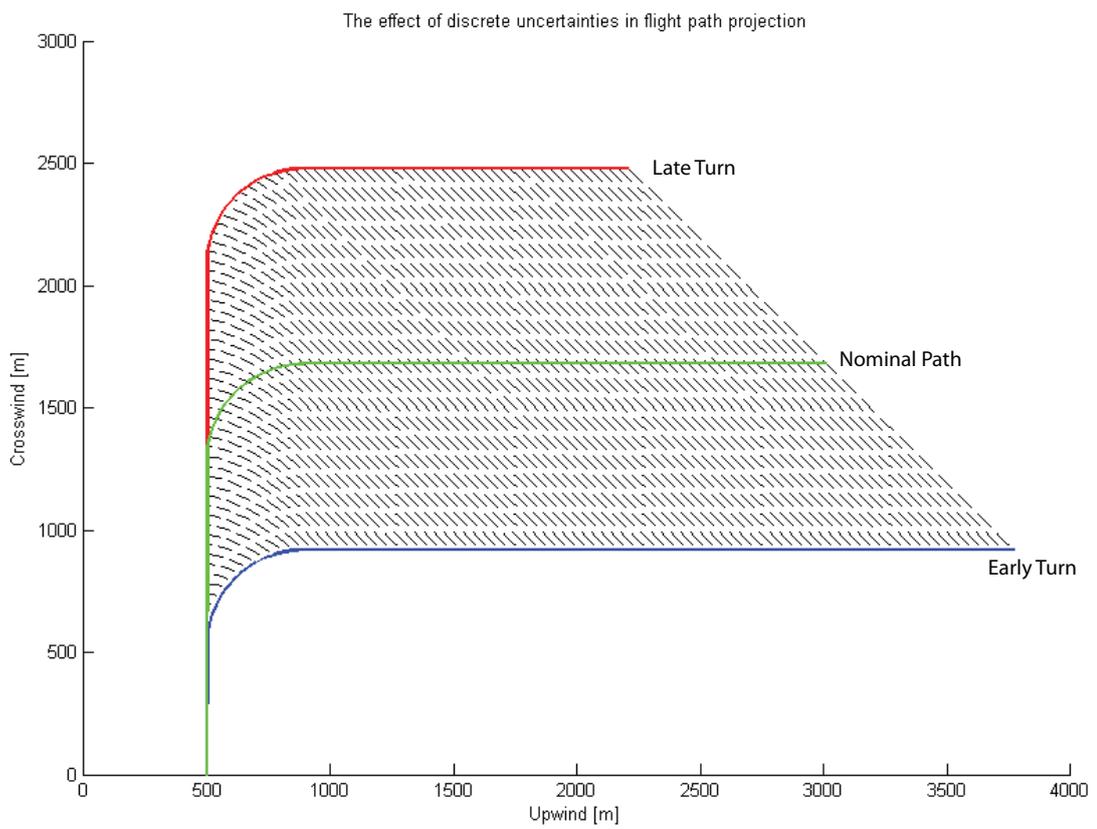


Figure 4.6: Discrete uncertainty in flight path projection

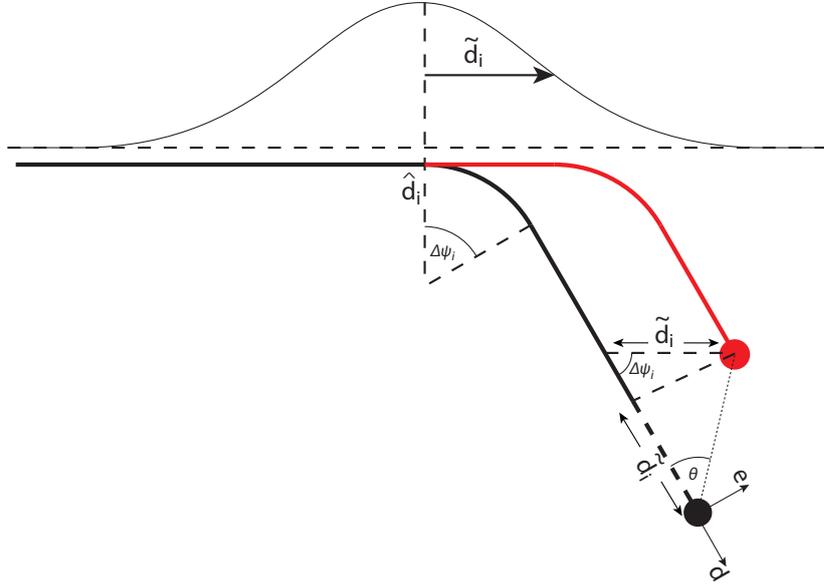


Figure 4.7: Effect of discrete transition variance on projection

and crosstrack error are given as

$$\epsilon_{i_d} = \begin{bmatrix} -\tilde{d}_i \\ \tilde{d}_i \sin(\Delta\psi_i) \end{bmatrix} \quad (4.18)$$

This leads to the covariance matrix for discrete transitions

$$\Sigma_{i_d} = \epsilon_{i_d} \epsilon_{i_d}^T = \begin{bmatrix} \tilde{d}_i^2 & -\tilde{d}_i^2 \sin(\Delta\psi_i) \\ -\tilde{d}_i^2 \sin(\Delta\psi_i) & \tilde{d}_i^2 \sin^2(\Delta\psi_i) \end{bmatrix} \quad (4.19)$$

Figure 4.8 shows the transformation of crosstrack error which occurs during a discrete transition. The subscript  $i$  on the crosstrack error term refers to the error accumulated between the  $(i-1)$ th and  $i$ th discrete transitions which can be written

$$\tilde{e}_i(t) = \begin{cases} \tilde{\psi}(\hat{d}_i - \hat{d}_{i-1}) & \text{if } i < n \\ \tilde{\psi}(d(t) - \hat{d}_{i-1}) & \text{if } i = n \end{cases} \quad (4.20)$$

where  $n$  is the next discrete transition at time  $t$ . The first equation in (4.20) accounts for the accumulation of crosstrack error from all discrete states up to the previous transition ( $i = n-1$ ). The second equation represents the continuous uncertainty in crosstrack error since the last discrete transition was encountered.

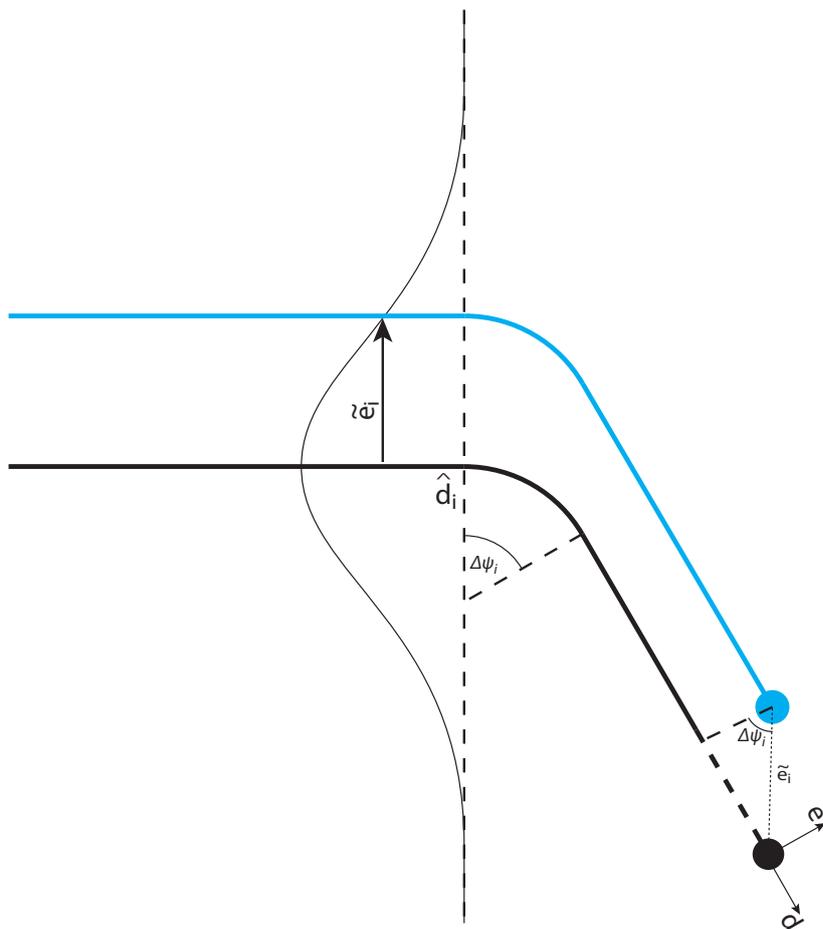


Figure 4.8: Effect of cross-track variance on projection

---

This piecewise continuous representation of crosstrack error is required to capture the transformation which occurs as each discrete transition is encountered. The crosstrack error term is then given as

$$\epsilon_{i_c} = \begin{bmatrix} -\tilde{e}_i \sin(\Delta\psi_i) \\ \tilde{e}_i \cos(\Delta\psi_i) \end{bmatrix} \quad (4.21)$$

This leads to a crosstrack error covariance matrix of

$$\Sigma_{i_e} = \epsilon_{i_c} \epsilon_{i_c}^T = \begin{bmatrix} \tilde{e}_i^2 \sin^2(\Delta\psi_i) & -\tilde{e}_i^2 \sin(\Delta\psi_i) \cos(\Delta\psi_i) \\ -\tilde{e}_i^2 \sin(\Delta\psi_i) \cos(\Delta\psi_i) & \tilde{e}_i^2 \cos^2(\Delta\psi_i) \end{bmatrix} \quad (4.22)$$

The covariance matrices given in (4.19) and (4.22) completely represent the distance uncertainty introduced and transformation of crosstrack error caused by discrete transitions. If the  $n$ th discrete transition is next at time  $t$  the projection of a traffic vehicles position is simply the sum of all the discrete transitions to  $n$ .

$$\Sigma_{tot}(t) = \Sigma_0(t) + \sum_{i=1}^n P_i(d(t))(\Sigma_{i_d} + \Sigma_{i_e}(t)) \quad (4.23)$$

where  $P_i(d(t))$  is the probability of the  $i$ th transition having occurred at distance  $d(t)$  along the path given by the Gaussian Cumulative Distribution Function (CDF)

$$P_i(d(t)) = \int_{-\infty}^{d(t)} p(x; \hat{d}_i, \tilde{d}_i) dx \quad (4.24)$$

where  $p(x; \hat{d}_i, \tilde{d}_i)$  is the value of the Gaussian PDF at  $x$  with mean  $\hat{d}_i$  and standard deviation  $\tilde{d}_i$ . The  $i$ th transition probability  $P_i(d(t))$  has the effect of switching on the uncertainties imposed by a transition as it is reached.

The  $\Sigma_0(t)$  term in (4.23) is a modified form of the continuous covariance matrix given in (4.14) with the crosstrack error variance removed (and vertical dimension ignored).

$$\Sigma_0(t) = \begin{bmatrix} \sigma_d(t)^2 & 0 \\ 0 & 0 \end{bmatrix} \quad (4.25)$$

Despite being continuous in nature, the crosstrack error must not be accounted for in the continuous covariance matrix as it is included in a piecewise fashion by

---

the discrete transitions in order to preserve the required transformations.

The covariance matrix (4.23) describes the uncertainty in projection of a traffic vehicle subject to discrete uncertainty in curvilinear coordinates. It accounts for the continuous uncertainty arising from errors in velocity estimation and discrete uncertainties arising from the commencement of manoeuvres at uncertainty times. Transformation to the CCS may be performed by applying (4.16) as before.

### Test results

To test the performance of discrete uncertainty projection, a typical terminal area path simulation was used. A traffic vehicle is positioned at  $[2000m \ 2000m]$  with a speed of  $50ms^{-1}$  and a heading of  $180^\circ$ . The nominal path consists of a  $90^\circ$  left turn at  $[-2000m \ 2000m]$  followed by a second turn to intercept the upwind axis.

It is assumed that the traffic vehicle maintains its present speed and heading with standard deviations of  $2ms^{-1}$  and  $2^\circ$  respectively. Additionally, the discrete transitions are assumed to have a standard deviation of  $200m$ .

Figure 4.9 shows the curvilinear projection with and without discrete uncertainty compared against a Monte Carlo simulation at three distinct projection times. Figure 4.9a shows the the full Monte Carlo as 2D histograms. Figure 4.9b shows the projection without discrete uncertainty with a subset of the Monte Carlo results overlaid for comparison. It is clear that whilst the mean is well matched, the general shape of the distributions after encountering the discrete change is incorrect. Figure 4.9c shows the projection with discrete uncertainty, it can be seen that distribution capture the major features of the Monte Carlo results.

It is possible to directly use the Monte Carlo distributions, however the time taken to calculate each distribution is of the order of  $60s$ . A number of distributions will be required by UAS in order to assess potential of flight path conflicts, therefore the Monte Carlo method would seriously limit the real time potential. Utilising an analytical projection method as presented here reduces the computational time to around  $0.5s$  per distribution making real time implementation possible.

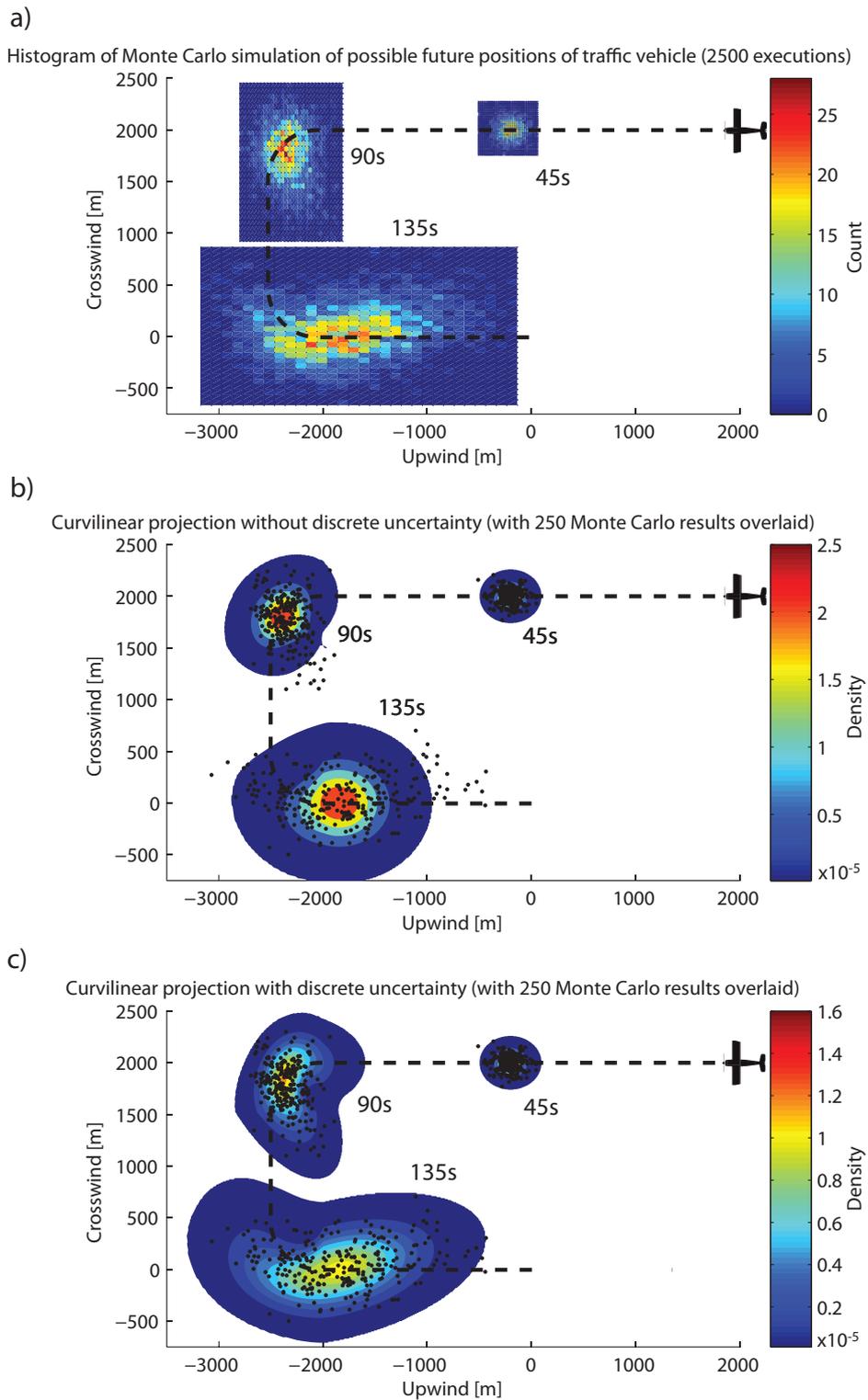


Figure 4.9: Comparison of curvilinear projection with and without discrete uncertainty vs Monte Carlo simulation

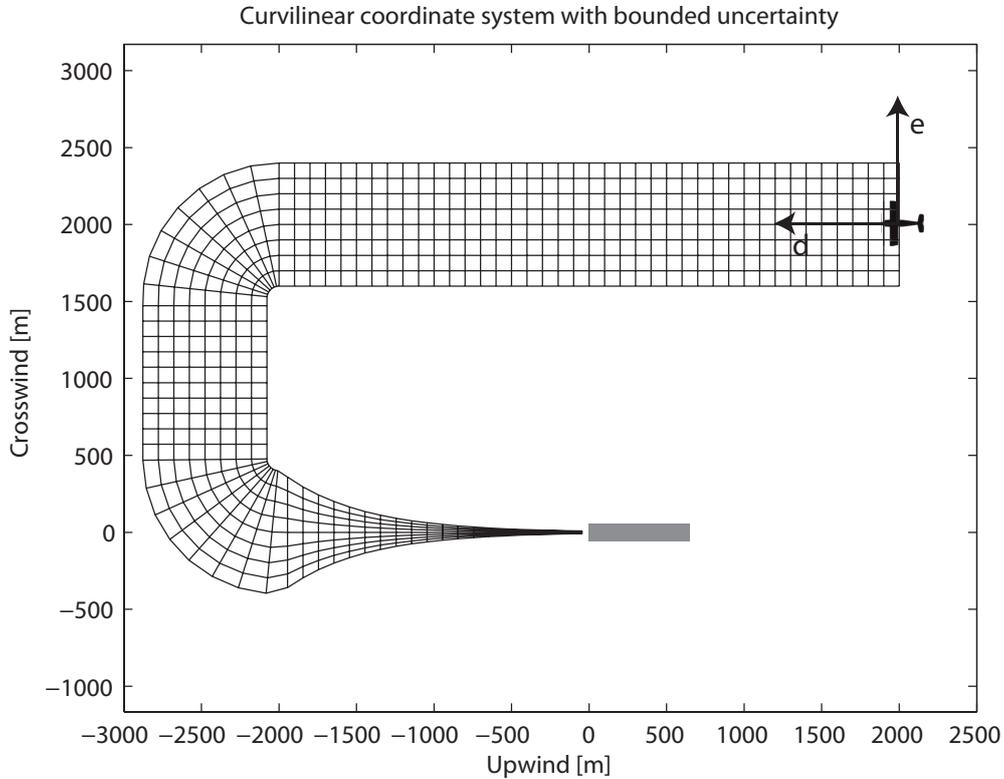


Figure 4.10: Curvilinear coordinates with bounded uncertainty

#### 4.3.4 Bounded uncertainty

It is clear from (4.23) that the uncertainties in distance and crosstrack error increase with projection time. The terminal area circuit, however, presents a unique bound to the uncertainties due to the presence of a runway as the final waypoint. Whilst it is true that uncertainty in distance continues to increase with time (due to accumulation of speed errors), uncertainty in crosstrack error increases to a point and then begins to reduce to almost zero at the runway threshold. This behavior can be captured either by a constriction applied to the curvilinear reference frame, Figure 4.10, or by manipulating the covariance matrix.

Figure 4.10 illustrates the constriction which occurs by applying an exponen-

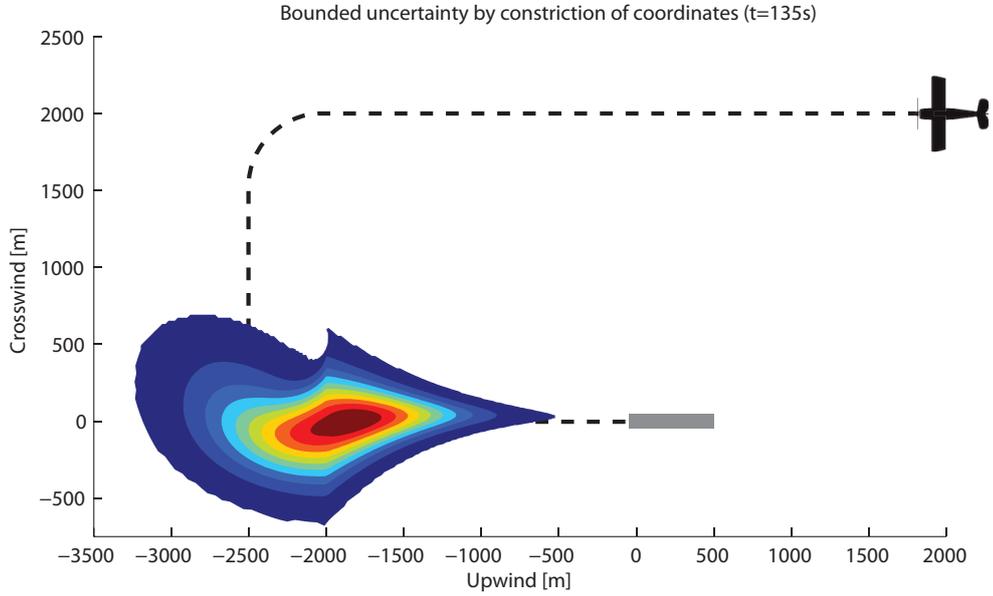


Figure 4.11: The result of coordinate constriction

tial decay to the crosstrack term in (4.16) for the final phase of flight.

$$c'(t) = c(t)e^{-k_1 d_f(t)/\hat{d}_f} \quad (4.26)$$

where  $d_f(t)$  is the distance traveled along the final phase,  $\hat{d}_f$  is the nominal length of the final phase and  $k_1$  is the decay rate.

Figure 4.11 illustrates the effect of coordinate constriction ( $k_1 = 3$ ) on the projection PDF by reproducing the 135s result from Figure 4.9 with (4.26) applied after the final turn. Due to the correlation present between crosstrack and distance uncertainty, it is necessary to check that the exponential decay is sufficient as the projection time tends towards the traffic vehicles ETA, i.e. the mean position approaches the runway threshold. This is illustrated in Figure 4.12 by increasing the projection time to 160s, it is apparent that the exponential decay with  $k_1 = 3$  seems sufficient to represent the problem.

An alternative method for representing bounded uncertainty is to alter the

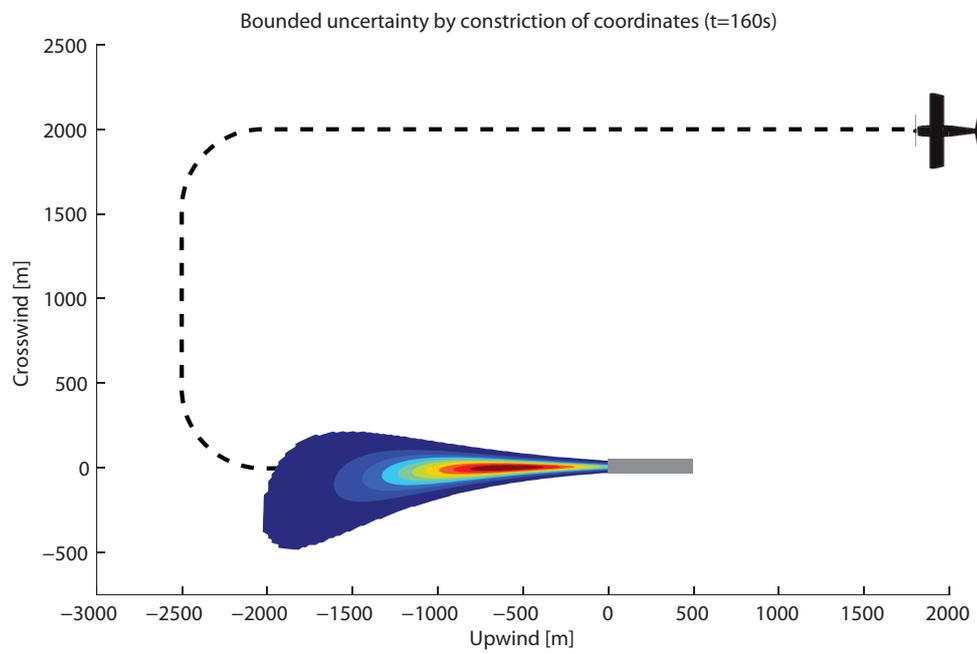


Figure 4.12: The result of coordinate constriction for projection times close to [ETA](#)

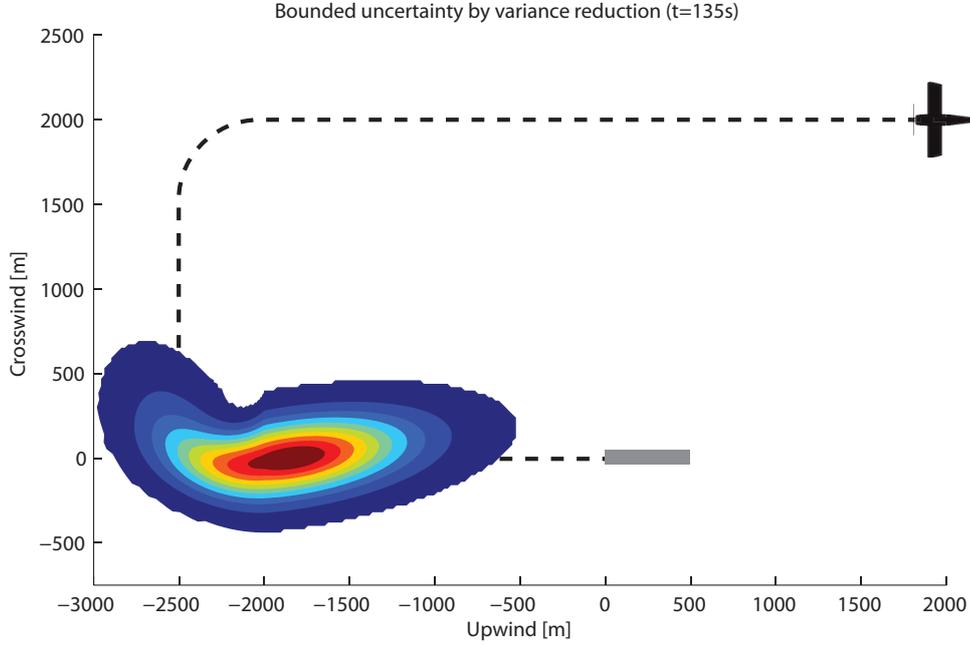


Figure 4.13: The result of variance reduction

covariance matrix to reflect the reduced uncertainty. This can be achieved by applying a similar exponential decay whilst maintaining positive semi-definiteness.

$$\Sigma'_{tot}(t) = \begin{bmatrix} 1 & 0 \\ 0 & e^{-k_2 \hat{d}_f(t)/\hat{d}_f} \end{bmatrix} \Sigma_{tot}(t) \begin{bmatrix} 1 & 0 \\ 0 & e^{-k_2 \hat{d}_f(t)/\hat{d}_f} \end{bmatrix}^T \quad (4.27)$$

where  $k_2$  is the decay rate and  $\hat{d}_f(t)$  is the mean distance traveled along the final leg at time  $t$ . Note that this is different from the  $d_f(t)$  term in (4.26) which refers to the distance of any point in the distribution.

Figures 4.13 and 4.14 are equivalent to Figures 4.11 and 4.12 but for the variance reduction approach. Whilst reducing the variance in order to effect a reduction in uncertainty may seem more intuitive than coordinate constriction, it can be seen that performance is actually impaired. Because the reduction given in (4.27) acts on the covariance matrix of the Gaussian PDF in curvilinear coordinates, it effects the whole distribution uniformly even when transferred to

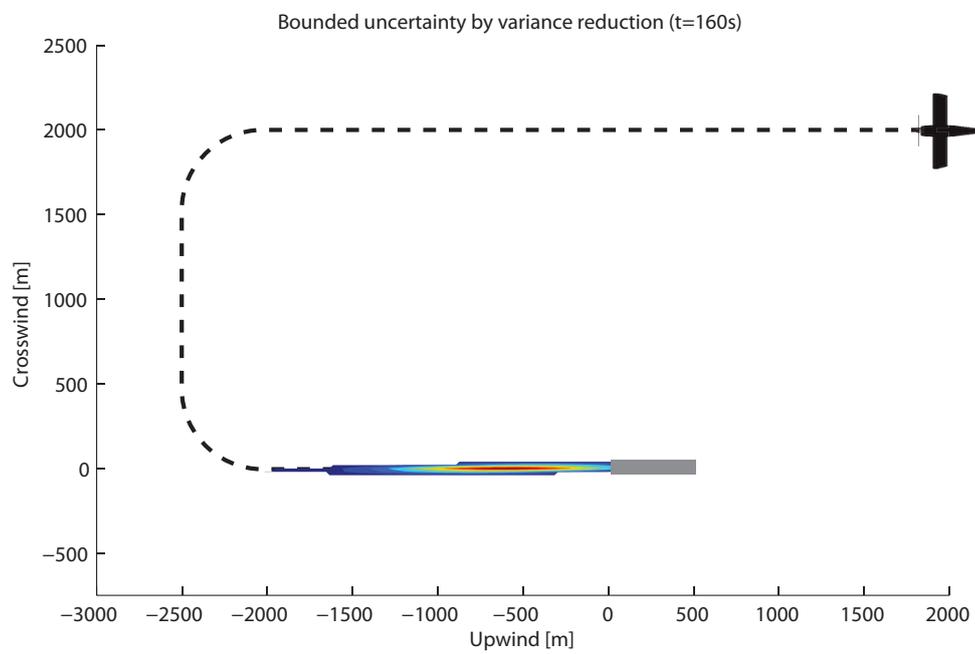


Figure 4.14: The result of variance reduction for projection times close to [ETA](#)

---

the global frame. In contrast, the coordinate constriction approach is applied to each point in the curvilinear system individually as it is transferred to global coordinates resulting in a non-uniform transformation. These effects are clear when comparing Figure 4.11 with Figure 4.13 where the former demonstrates a continual decrease in uncertainty as the runway is approached whereas the latter shows a one time reduction in uncertainty (visible when compared with the 135s PDF in Figure 4.9). Comparison of Figure 4.12 with Figure 4.14 illustrates the same effect.

The reduction given in (4.27) acts on the entire distribution so as to maintain the PDF as a valid probability distribution. It is possible to recalculate the covariance matrix at each point in the distribution by replacing  $\hat{d}_f(t)$  with  $d_f(t)$  to obtain a result similar to the coordinate constriction case but this is no longer a valid PDF as the volume under the surface is no longer guaranteed to be unity. To bound the uncertainty in this way would require the subsequent PDF to be normalised which is computationally expensive as it requires numerical evaluation of the CDF, therefore it is proposed that coordinate constriction is used for the most accurate results. Coordinate constriction always results in a valid PDF as it uses the unmodified covariance matrix in curvilinear coordinates directly.

### 4.3.5 Overview

Figure 4.15 illustrates the complete Curvilinear Projection with Discrete Uncertainty (CPDU) system discussed above and how it links to the comprehension stage discussed previously. The projection bow on the right of Figure 4.15 illustrates a single projection time, the algorithm must be executed repeatedly for additional projection times.

## 4.4 Results

The previous sections have detailed the development of a method for projecting the position of a traffic vehicle by incorporating continuous and discrete uncertainties in addition to the bounds imposed by terminal area operations. This section provides an assessment of the performance of the proposed method by

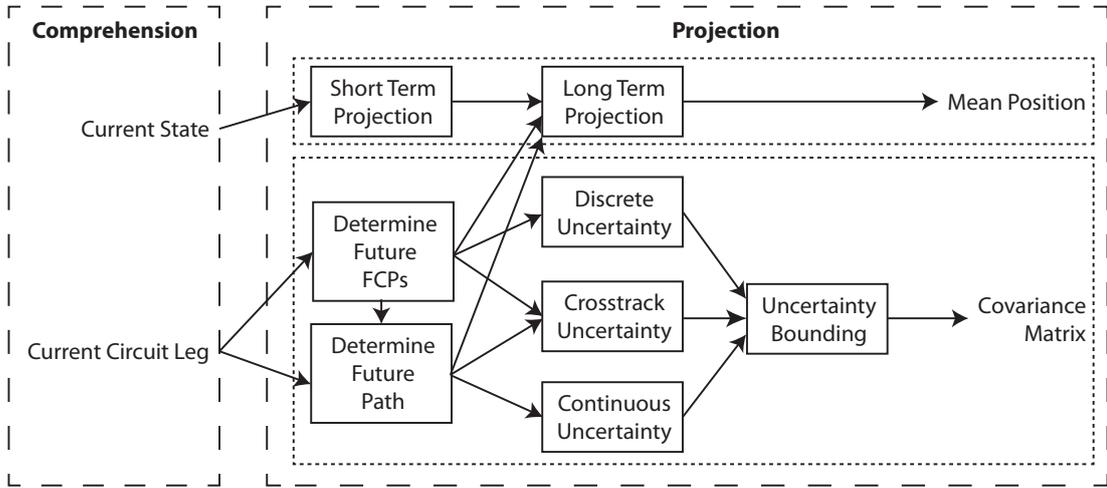


Figure 4.15: Diagram of the CPDU algorithm

comparing its results with those of a Monte Carlo simulation based on a stochastic aircraft model. The uncertainty parameters used in the model are assumed to be known exactly by the projection method so as to avoid the problems associated with parameter estimation. This also allows for an infinite number of test scenarios to be computed. For real world applications this would not be the case and parameters would have to be estimated from a large data sample. A sensitivity analysis is presented to determine acceptable tolerances with which parameters must be estimated in the future.

#### 4.4.1 Stochastic traffic model

To provide a comparison with the projected PDF, a model for a typical traffic vehicle is required which must be stochastic in nature to mimic the uncertainty and indeterminism of human navigation decision making. To simplify this model, only the two dimensional case is considered and it is assumed that the vehicles maintains a constant speed which is unknown to the UAS but can be estimated during the comprehension stage.

$$\begin{bmatrix} u \\ c \\ \psi' \end{bmatrix}_t = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ c \\ \psi' \end{bmatrix}_{t-1} + \begin{bmatrix} dt \cos(\psi'_{t-1}) & 0 \\ dt \sin(\psi'_{t-1}) & 0 \\ 0 & dt \end{bmatrix} \begin{bmatrix} v \\ v_\psi \end{bmatrix}_{t-1} \quad (4.28)$$

---

where  $v$  and  $v_\psi$  are control parameters determined with respect to the nominal path and are both Gaussian random variables. The velocity is a constant sampled from a Gaussian PDF at  $t = 0$

$$v_0 \sim \mathcal{N}(\hat{v}, \tilde{v}^2) \quad (4.29)$$

where  $\hat{v}$  and  $\tilde{v}$  are the mean and standard deviation in speed respectively.

The turn rate is held zero during straight flight and constant during any given maneuver, this value is also sampled from a Gaussian PDF

$$v_\psi \sim \mathcal{N}(\hat{v}_\psi, \tilde{v}_\psi^2) \quad (4.30)$$

where  $\hat{v}_\psi$  and  $\tilde{v}_\psi$  are the mean and standard deviation turn rates respectively. A maneuver is specified by sampling a distance from (4.17) and as this is reached a turn rate is sampled from (4.30) and applied to (4.28). Each maneuver also has a specified target heading which, along with the initial heading, is also a Gaussian random variable.

$$\psi'_i \sim \mathcal{N}(\hat{\psi}'_i, \tilde{\psi}'^2) \quad (4.31)$$

where  $\psi'_i$  is the target heading of the  $i$ th discrete phase (relative to runway heading),  $\hat{\psi}'_i$  is the mean target and  $\tilde{\psi}'$  is the standard deviation in heading. Note that the same standard deviation is used for all transitions hence the subscript  $i$  is omitted.

To ensure the traffic vehicle converges to the runway, the target heading is modified slightly in the final phase

$$\psi'_{thresh} = \hat{\psi}'_n + \tan^{-1} \left( \frac{c_t}{u_t} \right) \quad (4.32)$$

$$\psi' = \psi'_{thresh} + k(\psi'_{thresh} - \hat{\psi}'_n) \quad (4.33)$$

where  $\psi'_{thresh}$  is the target heading to reach the runway threshold,  $\hat{\psi}'_n$  is the mean heading of the final ( $n$ th) phase (which should always equal 0 or  $2\pi$  by definition) and  $k$  is a proportional gain for minimising the error between the heading to the threshold and the runway heading (equal to the heading of the final phase).

---

Table 4.1: Testing parameters

$\hat{v}$	Mean speed	$50ms^{-1}$
$\tilde{v}$	Speed standard deviation	$2ms^{-1}$
$\hat{v}_\psi$	Mean heading rate	$0.1rads^{-1}$
$\tilde{v}_\psi$	Heading rate standard deviation	$0.02rads^{-1}$

Table 4.2: Circuit parameters

$u_0$	Initial upwind position	$2000m$
$c_0$	Initial crosswind position	$2000m$
$\hat{\psi}'_0$	Initial heading	$\pi rad$
$\hat{\psi}'_1$	Heading of base leg	$3\pi/2rad$
$\hat{\psi}'_2$	Heading of final leg	$2\pi rad$
$\hat{d}_1$	Distance to end of downwind leg	$4000m$
$\hat{d}_2$	Distance to end of base leg	$5750m$
$\tilde{\psi}'$	Standard deviation of heading flown	$0.035rad$
$\tilde{d}$	Standard deviation of discrete changes	$200m$

This ensures that a vehicle which is initially off the nominal path will reach the runway threshold with its heading aligned to that of the runway so as to facilitate a landing. The gain  $k$  determines how quickly an off track vehicle converges to the runway heading, it can be chosen to provide either an aggressive or gentle manoeuvre (such as by a fast jet or trainer respectively).

#### 4.4.2 Model parameters

To test the projection method against the Monte Carlo model given in the last section, parameters for a typical light aircraft (e.g. a Cessna 172) were used, as given in Table 4.1. It is assumed that the vehicle is observed in the early downwind location at a typical small airfield (e.g. Cranfield), therefore the circuit is described by the parameters in Table 4.2.

#### 4.4.3 Error calculation

To determine the performance of the projection method verses the Monte Carlo simulation, it is necessary to define a metric to quantify the error. Due to the

---

two dimensional nature of the problem, it is difficult to visualise a histogram of the Monte Carlo results against a surface plot of the analytical PDF therefore marginal distributions are used. A marginal distribution is the PDF obtained by integrating over one or more variables in a MVPDF (i.e. marginalising them out) leaving only the remaining variables [Anderson, 1984]. By marginalising along the upwind and crosswind axes independently, two unidimensional PDFs are obtained which allow for comparisons to be drawn between the analytical and Monte Carlo approaches.

A 1000 iteration Monte Carlo simulation is used at each projection time. To marginalise these simulations, two histograms are produced of the upwind and crosswind coordinates (marginalising the crosswind and upwind axes respectively). Marginalising the analytical distribution is more complex as it is parameterised in curvilinear coordinates. A large number of samples must be drawn from the distribution (in curvilinear coordinates) and their corresponding location in the CCS must be determined (applying (4.16) and (4.26) where necessary). The bins of the Monte Carlo histogram are then used as boundaries to sum the probabilities of the PDF across, giving the Probability Mass Functions (PMFs) [Rao, 2009].

$$P(u_i < u \leq u_{i+1}) = \frac{\sum_{n=0}^N z_n \Gamma(u_i < u_n \leq u_{i+1})}{\sum_{n=0}^N z_n} \quad (4.34)$$

$$P(c_i < c \leq c_{i+1}) = \frac{\sum_{n=0}^N z_n \Gamma(c_i < c_n \leq c_{i+1})}{\sum_{n=0}^N z_n} \quad (4.35)$$

where  $u_i$ ,  $u_{i+1}$ ,  $c_i$  and  $c_{i+1}$  represent the boundaries of the  $i$ th Monte Carlo histogram bin,  $z_n$  is the value of the  $n$ th sample from the PDF,  $N$  is the total number of samples taken and the function  $\Gamma$  is defined as follows.

$$\Gamma(a < x \leq b) = \begin{cases} 1 & \text{if } a < x \leq b \\ 0 & \text{otherwise} \end{cases} \quad (4.36)$$

Comparison of the PMFs given by (4.34) and (4.35) with the Monte Carlo

---

histograms allows for an **RMS** error to be calculated for each marginal.

$$error_{upwind} = \sqrt{\frac{1}{M} \sum_{i=0}^M (P(u_i) - P_{MC}(u_i))^2} \quad (4.37)$$

$$error_{crosswind} = \sqrt{\frac{1}{M} \sum_{i=0}^M (P(c_i) - P_{MC}(c_i))^2} \quad (4.38)$$

where  $(P(u_i))$  and  $P_{MC}(u_i)$  are the values of the  $i$ th bin of the projection and Monte Carlo **PMF**s respectively and  $M$  is the total number of bins. Therefore the total error is

$$error = error_{upwind} + error_{crosswind} \quad (4.39)$$

The analysis above gives a measure of the error between the analytical projection method and a Monte Carlo simulation which can be easily visualised (by plotting the histogram and **PMF**) and quantified (by calculation of the **RMS** error). It is by no means the only method of quantifying the error, however, it is believed that this metric sufficiently captures the major features of the problem.

#### 4.4.4 Test 1 - Known parameters

The first test evaluates the **RMS** error at projection times from 30 – 140s in intervals of 1s with the projection uncertainties ( $\tilde{v}$  and  $\tilde{\psi}'$ ) known exactly. Figure 4.16 shows the error plotted against projection time for Linear, Curvilinear and Curvilinear with Discrete Uncertainty with vertical lines indicating the nominal discrete transitions.

It is clear that Linear projection is sufficient prior to the first discrete transition at  $t = 82s$ <sup>1</sup>, however beyond this point its performance degrades due to discrete uncertainties and the final approach phase. Curvilinear projection only provides a slight improvement in performance during turns but this is more significant in the final phase ( $t > 118s$ ) due to the coordinate constriction being applied. Finally, **CPDU** provides the best performance with the error largely remaining below 5% with peaks occurring during turns. Henceforth only **CPDU**

---

<sup>1</sup>where  $t$  refers to the time along the nominal path

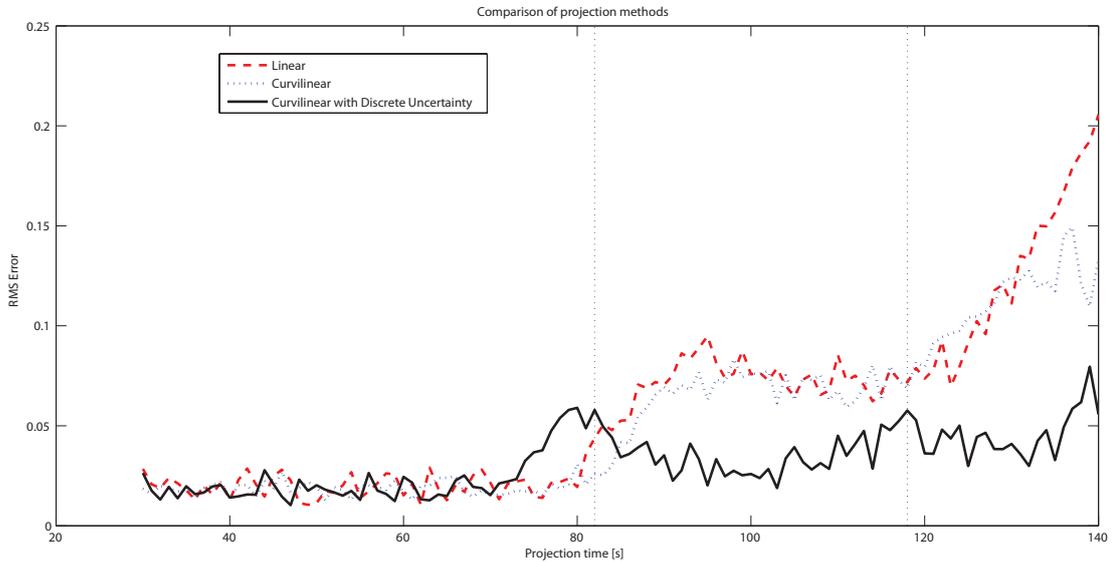


Figure 4.16: RMS error vs Projection time for first test

will be discussed.

The CPDU error increases as the first discrete transition is approached due to the effect of (4.24) switching on the discrete uncertainty covariance matrices. These matrices act to improve CPDU performance during and after discrete transitions as is demonstrated in Figure 4.16, however before a transition has occurred they act to increase the covariance uniformly both ahead of the mean position (where the transition may have begun) and behind the mean position (where it will not have begun) resulting in an overestimate of the uncertainty. This effect is undesirable, however, as it only effects a small portion of the projection domain and will always result in an overestimate of uncertainty (which would cause the UAS to plan more cautiously) it is not deemed unacceptable.

Figures 4.17 and 4.18 show the distribution and marginal distributions of the worst case points at the start of discrete transitions. The  $t = 50s$  distribution is given as a baseline comparison as this occurs in the region prior to any curved trajectories or discrete transitions.

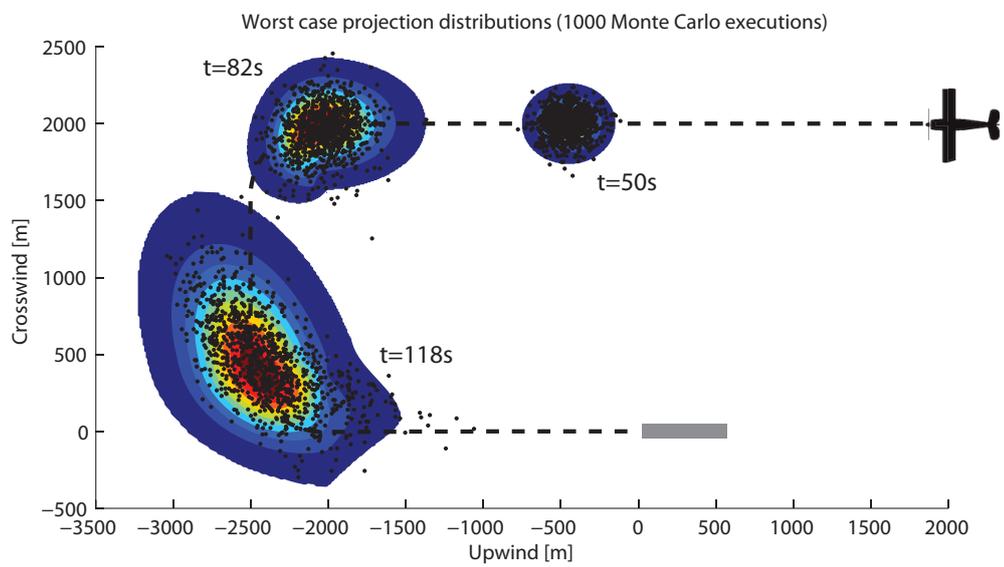


Figure 4.17: Worst case parameter combination - projection distributions

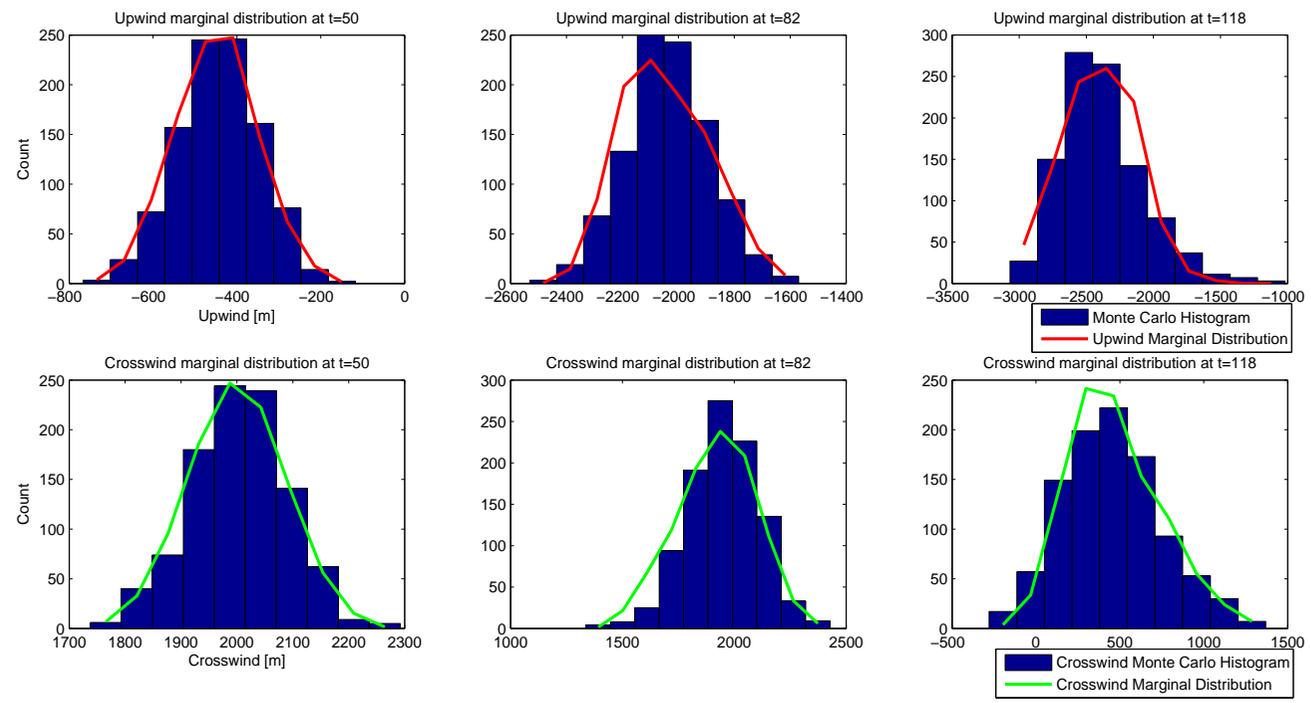


Figure 4.18: Worst case parameter combination - marginal distributions

---

Table 4.3: Test 2 average error results

$\tilde{v}$ \backslash $\tilde{\psi}'$	90%	100%	110%
90%	0.0088	0.0039	0.0057
100%	0.0012	-	0.0022
110%	0.0026	0.0018	0.0037

Figure 4.16 illustrates a very low (2%) error in the projection method prior to the first discrete transition. The increased error caused by discrete transitions is likely due to a number of factors such as the lack of turn rate uncertainty in the projection method, however a 6% error at these worst case points is acceptable and a significant improvement over the other methods tested. It is also apparent from Figure 4.16 that a general upward trend in error is present after the first transition is encountered, this is likely caused by the accumulation of errors introduced by the discrete transitions. At all times the error remains below 10% despite projection times of over 2 minutes, this is deemed to be highly acceptable.

#### 4.4.5 Test 2 - Unknown continuous parameters

This test looks at the sensitivity of the projection model to the continuous covariance parameters ( $\tilde{v}$  and  $\tilde{\psi}'$ ). In a complete system, these are the parameters which will have been estimated from collected data (such as radar traces) and therefore are subject to error. By varying the parameters of the stochastic model by  $\pm 10\%$  whilst keeping the projection model values constant the effects of this inevitable error can be illustrated. Table 4.3 shows the average increase in RMS error for each parameter combination. This is simply the mean error over the 30-140s period.

Figure 4.19 shows the worst ( $[0.9\tilde{v} \ 0.9\tilde{\psi}']$ ) and best case ( $[1.0\tilde{v} \ 0.9\tilde{\psi}']$ ) parameter variations compared with the exact values. It is clear that in the linear region before the first discrete transition, both parameter combinations result in poorer performance. At and after the discrete transition, the worst case parameters cause the error to increase whereas the best case causes a reduction. This reduction in error is what leads to the low average error and is likely a result of the lower heading uncertainty. With the projection method overestimating

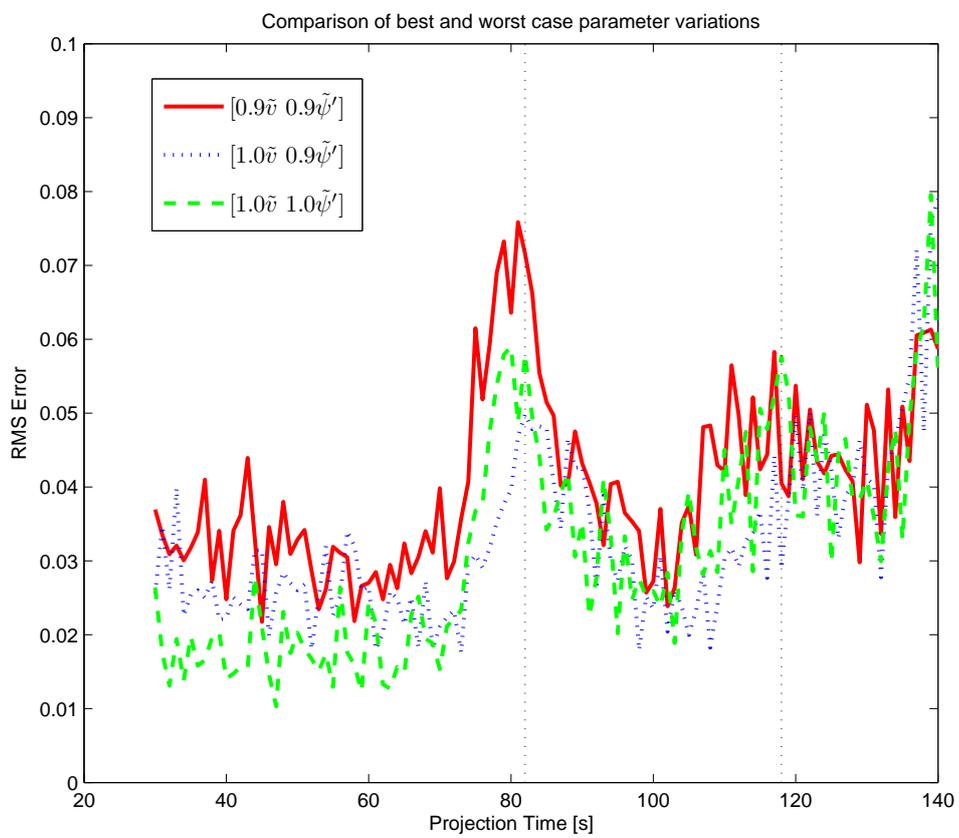


Figure 4.19: Worst and best case error comparison

---

the heading uncertainty, this will go some way to accounting for the turn rate uncertainty not captured in the model, hence its effect only appears during the first turn. An overestimate of velocity uncertainty as in the worst case parameter set, however, has a more significant increasing effect on the error.

With the 10% continuous parameter variation investigated in this section, the average increase in **RMS** error has been shown to be less 1% indicating the projection methodology is relatively insensitive to parameter estimation. When estimating the parameters from data, greater refinement should be given to the velocity variance as the algorithm seems more sensitive to this than heading variance.

#### 4.4.6 Test 3 - Unknown discrete variance

The previous section tested the sensitivity of projection to variations in the continuous uncertainty parameters, this test considers the discrete uncertainties. To test this, the distance variance in the stochastic model ( $\tilde{d}_i$ ) was varied by both  $\pm 10\%$  and  $\pm 25\%$ , the results are shown in Figure 4.20.

It can be seen from Figure 4.20 that the 10% variation has only a small effect on the **RMS** error. The 25% variations show a worst case increase in error of around 0.02 between transitions. As with the previous test, at all projection times the error remains below 10% which is considered to be excellent performance.

### 4.5 Summary

This chapter has detailed the development of a novel projection methodology for traffic vehicles in the terminal area. Prediction of the nominal future states based on comprehension of the current continuous and discrete state was presented and the sources of uncertainty in this identified. Uncertainty in both continuous and discrete variables have been captured to provide a system which is capable of predicting future traffic states and quantifying the inevitable uncertainty.

It has been shown that by considering the curved nature of the path and including discrete transition uncertainty, a high level of accuracy can be attained at projection times over 2 minutes. It has also been demonstrated that the

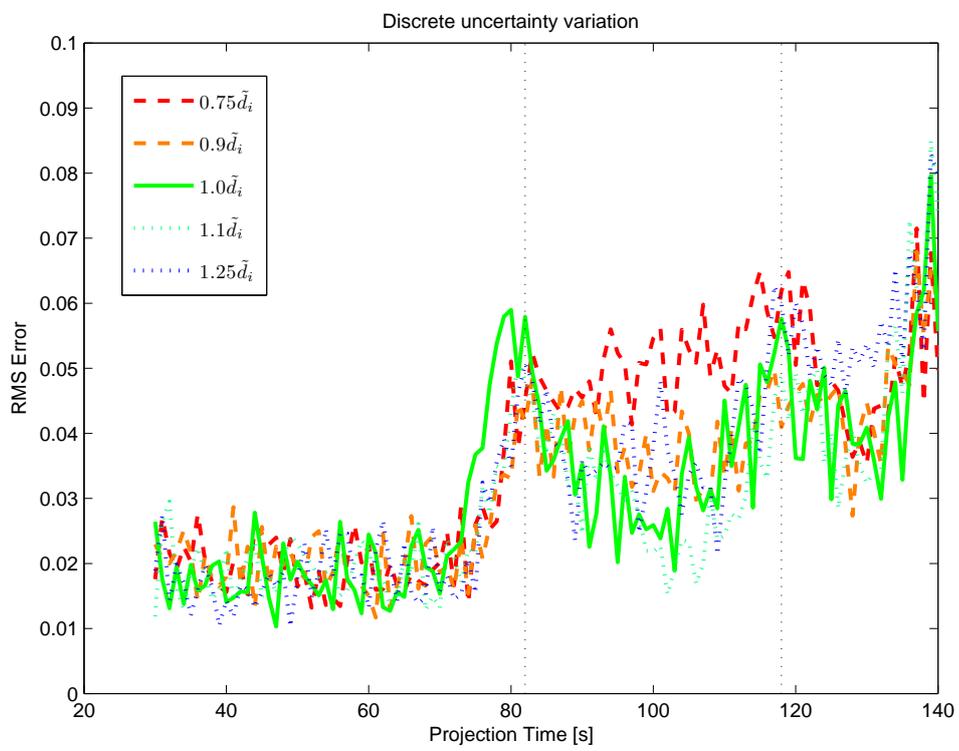


Figure 4.20: Sensitivity of projection to discrete uncertainty parameter variation

---

parameters representing the navigation uncertainties need not be known precisely to achieve good performance, this allows the parameters to be estimated from recorded radar or [GPS](#) data.

Referring to [Section 2.4.4](#) it is clear that the requirements for projection have not been fully met as only the future trajectory of the traffic is now known, not the separation from the [UAS](#). Separation is derived in the next chapter as a function of the projection [PDFs](#), providing a complete artificial situation awareness system.

# Chapter 5

## Self Separation

### 5.1 Introduction

The previous chapters have detailed the development of an artificial situation awareness system for [UASs](#) operating in the terminal area. Firstly, comprehension has provided an estimate of the continuous and discrete states of traffic vehicles. This comprehension has allowed the future states of traffic to be predicted and the inevitable uncertainty present in such a process has been quantified. Based upon this awareness an autonomous [UAS](#) is capable of making decisions about its future flight path so as to minimise conflicts with traffic. This chapter outlines the development of a self separation methodology to provide this functionality to the vehicle.

The next section details a number of separation metrics using basic linear projection to derive these in terms of a Gaussian [PDF](#). Section [5.3](#) generalises this to non-Gaussian distributions, based on the [CPDU](#) method described in the previous chapter. Finally, Section [5.4](#) uses these metrics as a basis for a simple decision making system to allow an autonomous [UAS](#) to operate safely in a crowded terminal area.

---

## 5.2 Self separation metrics

Spatial projection of traffic vehicles, such as that discussed in the previous chapters, provides the UAS with knowledge of the future positions of potentially conflicting aircraft. This information is not useful in its own right as what is required is a measure of the separation between the UAS and the traffic. Four different separation metrics can be derived from spatial projection.

1. Distance between vehicles

- *‘Vehicle X is 525m from Vehicle Y’*
- Often used for vortex wake separation [[Civil Aviation Authority, 2009a](#)]

2. Time separation

- *‘Vehicle X will land 90s before Vehicle Y’*
- Often used to separate arriving aircraft [[Civil Aviation Authority, 2013](#)]

3. Time to Point of Closest Approach (TPCA)

- *‘Vehicle X will be at its closest to Vehicle Y in 45s’*
- Used by TCAS [[Burgess et al., 1994](#)]

4. Distance at TPCA

- *‘Vehicle X will get no closer than 350m from Vehicle Y’*
- Used by newer collision avoidance techniques in conjunction with TPCA to eliminate false positives (for example the TCAS III concept [[Burgess et al., 1994](#)])

The following sections will derive relationships between the linear Gaussian spatial projection and the four separation metrics listed above. Later, these will be generalised to the non-Gaussian CPDU.

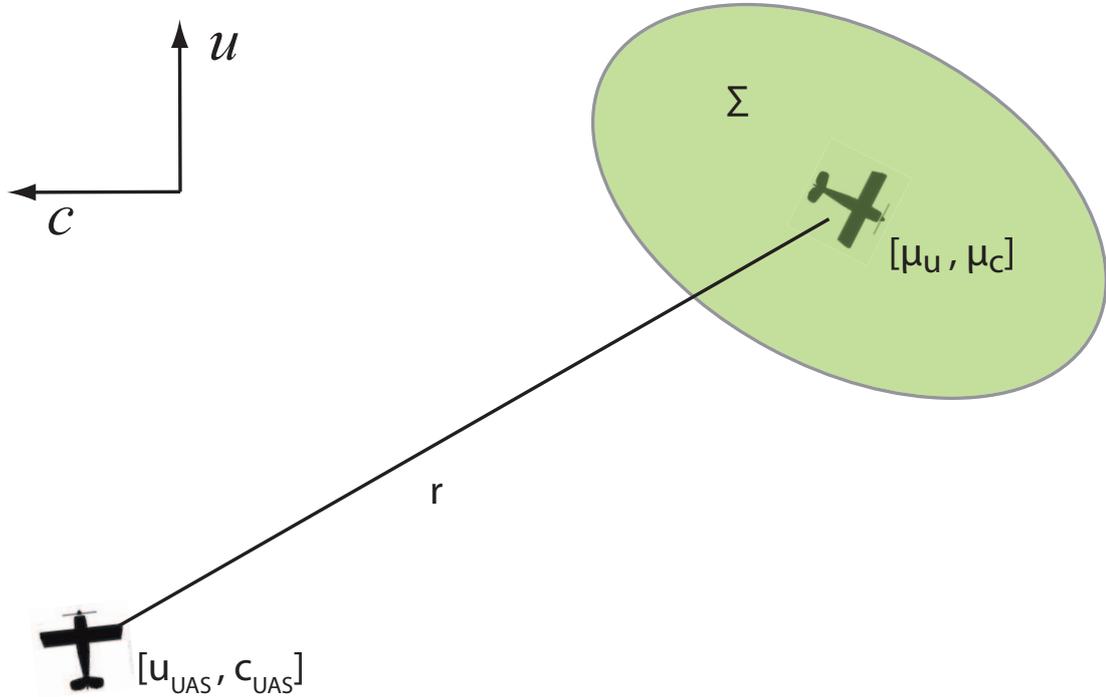


Figure 5.1: Distance between and known and uncertain point

### 5.2.1 Distance between vehicles

Calculation of the distance between a pair of two-dimensional vectors is trivial if both are known precisely, however Figure 5.1 illustrates the problem if one point is described by a MVPDF. The position of the UAS is assumed to be known with absolute certainty<sup>1</sup> and can be thought of as a point at a particular radius from the uncertain point. A probability distribution for the radius exists as a function of the uncertainty of the traffic position and is given by Weil [1954].

The radius distribution equation given by Weil [1954] applies only to an uncorrelated PDF, therefore the entire problem must first be transformed to remove any correlation. This is done by rotating the coordinate system to align with the axes of the ellipse described by the PDF shown in Figure 5.2.

---

**Definition** Let  $\theta_0$  be an angle of rotation which reduces the correlation of a

<sup>1</sup>If the future position of the UAS is uncertain due to ownship navigation errors, these may be included in the traffic vehicles PDF

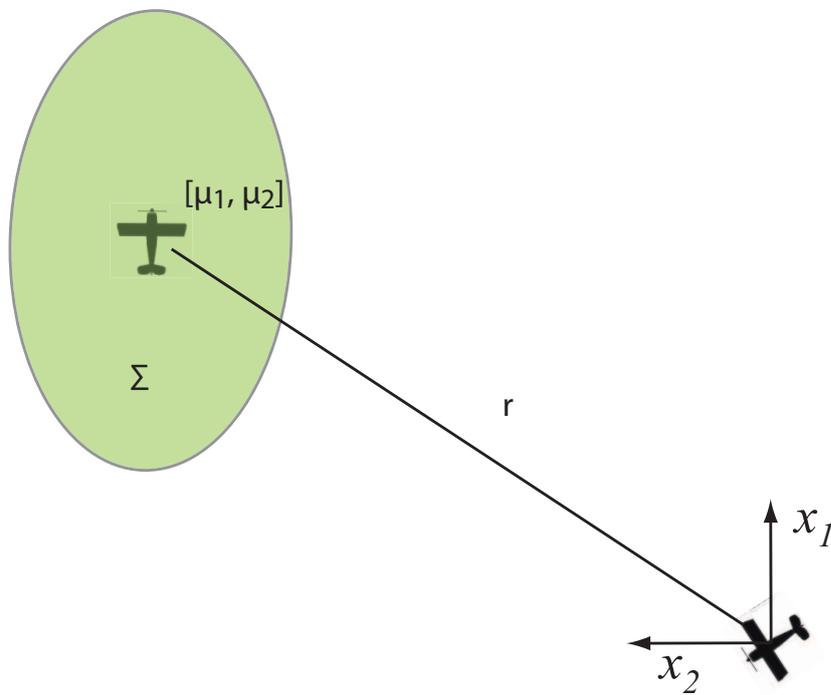


Figure 5.2: Distance between and known and uncertain point, coordinate system transformed

---

Bivariate Probability Density Function (BVPDF) to zero.

**Lemma 5.2.1** *Let  $\Sigma$  be the covariance matrix of a BVPDF. Then  $\theta_0$  is the required angle of rotation of the coordinate system to reduce the correlation to zero.*

**Proof** Let  $X \sim \mathcal{N}_2(\mu, \Sigma)$  and  $Y = R(\theta)X$  where  $R(\theta)$  is the 2D rotation matrix about and angle  $\theta$

$$R(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \quad (5.1)$$

then  $Y \sim \mathcal{N}(R(\theta)\mu, R(\theta)\Sigma R(\theta)^T)$  by Theorem 2.4.1 from Anderson [1984]. Rewriting  $\Sigma$  as

$$\Sigma = \begin{bmatrix} \sigma_a^2 & \sigma_{ab} \\ \sigma_{ab} & \sigma_b^2 \end{bmatrix} \quad (5.2)$$

and expanding the covariance term

$$R(\theta)\Sigma R(\theta)^T = \begin{bmatrix} \sigma_a^2 c^2 - 2\sigma_{ab}sc + \sigma_b^2 s^2 & \sigma_a^2 sc + \sigma_{ab}c^2 - \sigma_{ab}s^2 - \sigma_b^2 sc \\ \sigma_a^2 sc + \sigma_{ab}c^2 - \sigma_{ab}s^2 - \sigma_b^2 sc & \sigma_a^2 s^2 + 2\sigma_{ab}sc + \sigma_b^2 c^2 \end{bmatrix} \quad (5.3)$$

where  $\cos \theta$  and  $\sin \theta$  are abbreviated to  $c$  and  $s$  respectively.

Zero correlation implies the off diagonal terms in (5.3) equate to zero. Therefore

$$\sigma_a^2 \sin \theta_0 \cos \theta_0 + \sigma_{ab} \cos \theta_0^2 - \sigma_{ab} \sin \theta_0^2 - \sigma_b^2 \sin \theta_0 \cos \theta_0 = 0 \quad (5.4)$$

Rearranging

$$(\sigma_b^2 - \sigma_a^2) \sin \theta_0 \cos \theta_0 = \sigma_{ab}(\cos \theta_0^2 - \sin \theta_0^2) \quad (5.5)$$

Substituting double angle identities and rearranging

$$(\sigma_b^2 - \sigma_a^2) \sin 2\theta_0 = 2\sigma_{ab} \cos 2\theta_0 \quad (5.6)$$

Solve for  $\theta_0$

$$\theta_0 = \frac{1}{2} \tan^{-1} \left( \frac{2\sigma_{ab}}{\sigma_b^2 - \sigma_a^2} \right) \quad (5.7)$$


---

---

Applying Lemma 5.2.1 to find the required rotation angle and defining the new coordinates yields

$$\begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix} = R(\theta_0)\Sigma R(\theta_0)^T \quad (5.8)$$

$$\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} = R(\theta_0) \begin{bmatrix} \mu_u - u_{UAS} \\ \mu_c - c_{UAS} \end{bmatrix} \quad (5.9)$$

where  $\Sigma$ ,  $\mu_u$  and  $\mu_c$  are the covariance matrix and mean position of the traffic in the CCS respectively.  $u_{UAS}$  and  $c_{UAS}$  is the position of the UAS.

Equation (5.10) describes the radial PDF for  $r$ , the distance from the UAS to the traffic as illustrated in Figure 5.2, after this transformation [Weil, 1954]

$$p(r) = Ar \exp\left(\frac{-r^2(\sigma_1^2 + \sigma_2^2)}{4\sigma_1^2\sigma_2^2}\right) \left[ I_0(Br^2) I_0(Cr) + 2 \sum_{j=1}^{\infty} I_j(Br^2) I_{2j}(Cr) \cos(2ja) \right] \quad (5.10)$$

where

$$\begin{aligned} A &= \frac{1}{\sigma_1\sigma_2} \exp\left(\frac{-\mu_1^2\sigma_1^2 + \mu_2^2\sigma_2^2}{2\sigma_1^2\sigma_2^2}\right) \\ B &= \frac{(\sigma_1^2 - \sigma_2^2)}{4\sigma_1^2\sigma_2^2} \\ C &= \sqrt{\frac{\mu_1^2}{\sigma_1^4} + \frac{\mu_2^2}{\sigma_2^4}} \\ a &= \tan^{-1}\left(\frac{\mu_2\sigma_1^2}{\mu_1\sigma_2^2}\right) \end{aligned}$$

and  $I_n$  is the  $n$ th order modified Bessel function of the first kind.

Figure 5.3 illustrates the distribution given in (5.10) against a 1000 execution Monte Carlo simulation based upon the parameters

$$\begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix} = \begin{bmatrix} (500m)^2 & 0 \\ 0 & (400m)^2 \end{bmatrix}$$

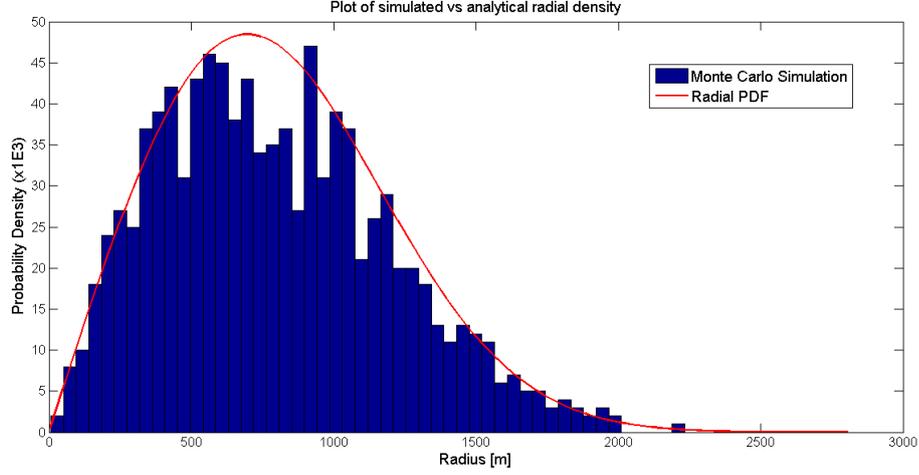


Figure 5.3: Comparison of simulated and analytical radial distribution

$$\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} = \begin{bmatrix} 600m \\ -200m \end{bmatrix}$$

It can be seen from Figure 5.3 that the analytical PDF closely matches the simulation, however it is computationally difficult to evaluate due to the infinite sum and exponential nature of the Bessel functions. Eckler [1969] demonstrates that the Wilson-Hilferty transformation [Wilson and Hilferty, 1931] of a weighted  $\chi^2$  distribution to a Gaussian variable as shown by Grubbs [1964] can be used to approximate (5.10) to a standard (zero mean, unit variance) Gaussian distribution. The Wilson-Hilferty transformation for  $r$  as given in Eckler [1969] is <sup>1</sup>

$$p_r = \frac{(r^2 / [(\sigma_1^2 + \sigma_2^2) m])^{\frac{1}{3}} - (1 - v / [9m^2])}{(v / [9m^2])^{\frac{1}{2}}} \quad (5.11)$$

where

$$m = 1 + \frac{\mu_1^2 + \mu_2^2}{\sigma_1^2 + \sigma_2^2}$$

and

$$v = \frac{2(\sigma_1^4 + \sigma_2^4 + 2\sigma_1^2\mu_1^2 + 2\sigma_2^2\mu_2^2)}{(\sigma_1^2 + \sigma_2^2)^2}$$

where  $p_r \sim \mathcal{N}(0, 1)$ .

<sup>1</sup>Note the erroneous  $(\bullet)^2$  in the numerator of  $v$  in Eckler [1969]

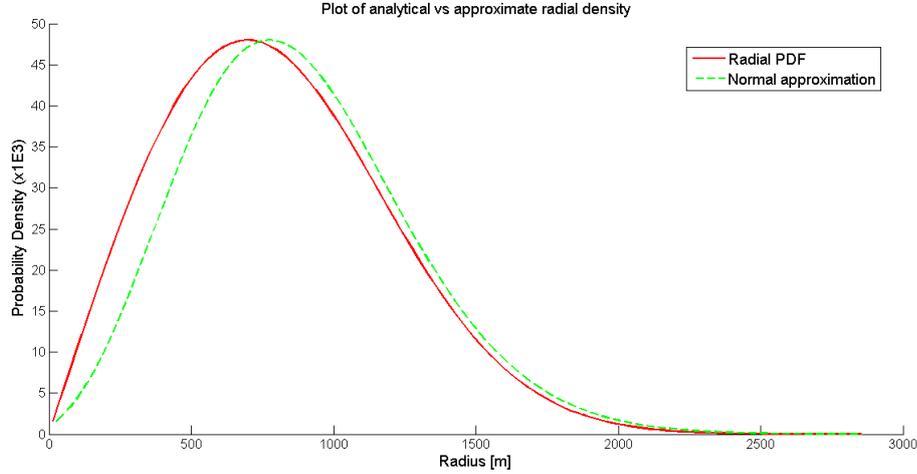


Figure 5.4: Comparison of analytical and approximate radial distribution

Figure 5.4 shows a comparison of the analytical radial PDF and the Gaussian approximation given in (5.11) for the same parameters as the previous figure. It can be seen that the shape of the distribution is largely correct however the approximation is skewed to the right. This discrepancy is caused by the highly non-Gaussian nature of the problem due to the strictly positive nature of the distribution. The problem arises when the means  $\mu_1$  and  $\mu_2$  are of similar size to the standard deviations ( $\sigma_1$  and  $\sigma_2$ ) causing a large proportion of the distribution to appear behind the target point as illustrated in Figure 5.5.

Figure 5.6 shows a comparison when the means are an order of magnitude greater than the variances ( $\mu_1 = 2100m$  and  $\mu_2 = 1500m$ ). It is clear that the approximation is far better under these conditions as the analytical distribution tends closer to Gaussian.

Despite the poorer performance of the approximation under certain conditions the ease of calculation means it is best suited to real time implementation onboard an UAS. In addition it is clear from Figure 5.5 that the poor performance is associated with the case where the UAS is within the uncertain region of the traffic, a condition which is unlikely to occur.

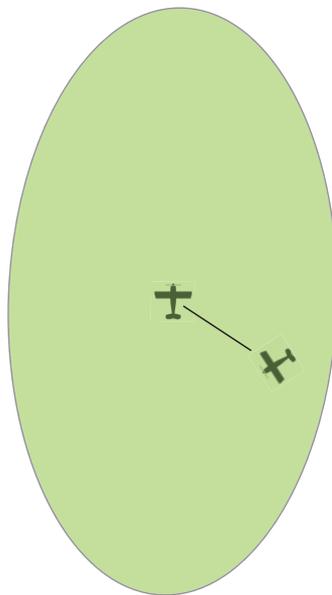


Figure 5.5: Illustration of large variance relative to mean

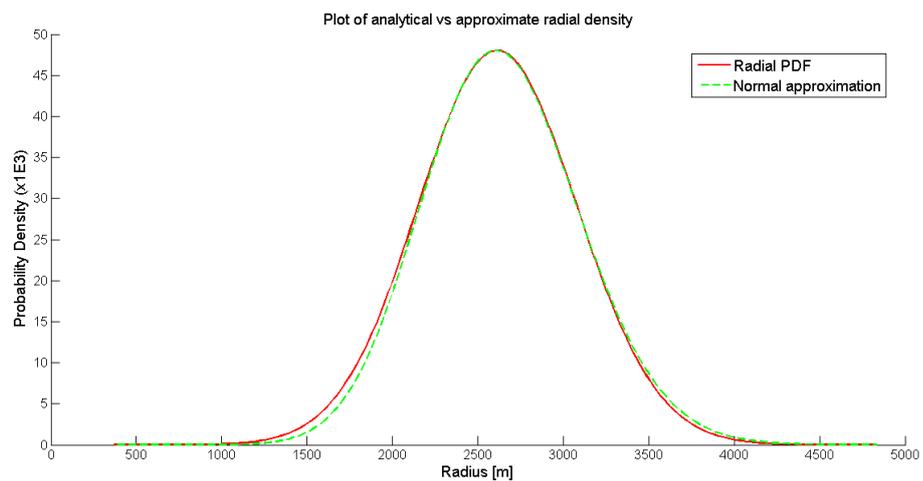


Figure 5.6: Comparison of analytical and approximate radial distribution with large mean relative to variance

---

## 5.2.2 Time separation

The previous section derived a distribution which describes the distance between an **UAS** and the projected position of a traffic vehicle. Temporal separation between vehicles is required as a means of preventing runway conflicts and is defined as the time taken for a following vehicle to reach the point at which the leading vehicle is currently positioned. The assumption made by this metric is that the **UAS** is intending to fly the same path as the traffic, however for a large proportion of the terminal area this is likely to be the case. This is beneficial to considering the relative velocity of the vehicles as the projected velocity of the traffic is uncertain leading to a ratio distribution problem when coupled with the spatial distribution derived previously.

To calculate the temporal separation distribution, the mean and covariance matrices must be modified by the **UAS** velocity prior to use

$$\begin{bmatrix} \mu_{1t} \\ \mu_{2t} \end{bmatrix} = \frac{1}{V_{UAS}} \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \quad (5.12)$$

$$\begin{bmatrix} \sigma_{1t}^2 & 0 \\ 0 & \sigma_{2t}^2 \end{bmatrix} = \frac{1}{V_{UAS}^2} \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix} \quad (5.13)$$

where the subscript  $t$  indicates temporal parameters and  $V_{UAS}$  is the **UAS** velocity.

Therefore the Gaussian approximation given in (5.11) may be used for temporal separation by rewriting as

$$p_t = \frac{(t^2 / [(\sigma_1^2 / V_{UAS}^2 + \sigma_2^2 / V_{UAS}^2) m])^{\frac{1}{3}} - (1 - v / [9m^2])}{(v / [9m^2])^{\frac{1}{2}}} \quad (5.14)$$

where  $m$  and  $v$  are unchanged.

Figures 5.7 and 5.8 illustrate the comparison between (5.14) and a temporally modified form of (5.10) using the same sets of parameters used in Figures 5.4 and 5.6 respectively and a **UAS** velocity of  $50ms^{-1}$ . It can be seen from Figure 5.7 that the breakdown in accuracy encountered by the Gaussian approximation occurs when the separation time is much less than 30s. Whilst there is no predefined separation criteria between small aircraft with which **UAS** are likely to interact

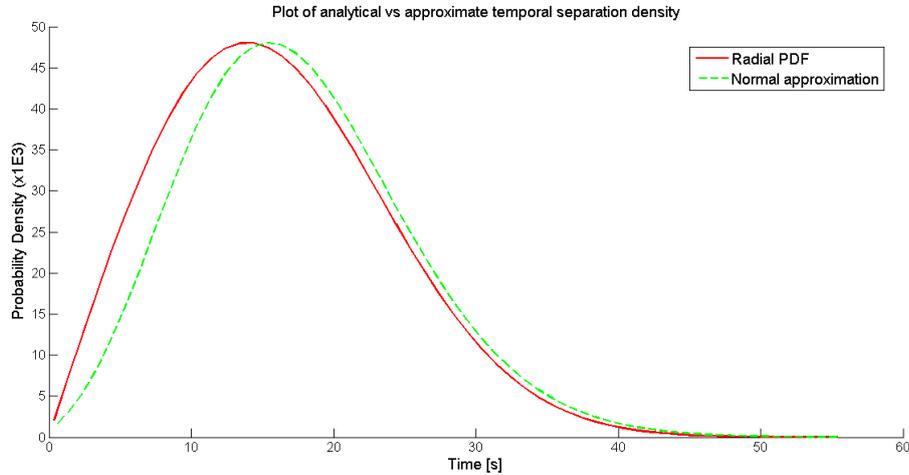


Figure 5.7: Comparison of analytical and approximate temporal distribution with comparative mean relative to variance

in the first instance (such as wake vortex separation between larger aircraft), a sufficient separation must be present to allow a leading vehicle to vacate the runway prior to the **UAS** attempting to land. It is proposed that when a **UAS** is making routing decisions based upon a prediction of the future state of traffic a temporal separation of at least 30s is desired to account for errors in the prediction and allow sufficient time for runway vacation. Figure 5.8 illustrates that the performance of the Gaussian approximation at separation times closer to the 30s target is excellent.

A closer look at the discrepancy between the analytical and approximate solutions in Figure 5.7 shows the approximation to under estimate the probability on side of the curve closest to the origin. When used in a **CDF**, see below, this will have the effect of under accounting for the risk associated with that situation. It is clear that under no circumstances should the **UAS** be attempting to position itself within 10s of another vehicle, but should this situation arise unexpectedly, it must be able to deal with it safely [Civil Aviation Authority, 2010b]. The Gaussian approximation shall be used from this point forwards and therefore the proposed method is not suitable for these short term collision avoidance situations. The proposed projection method is instead suited to the avoidance of the occurrence of such situations by keeping the **UAS** safely separated from all other vehicles.

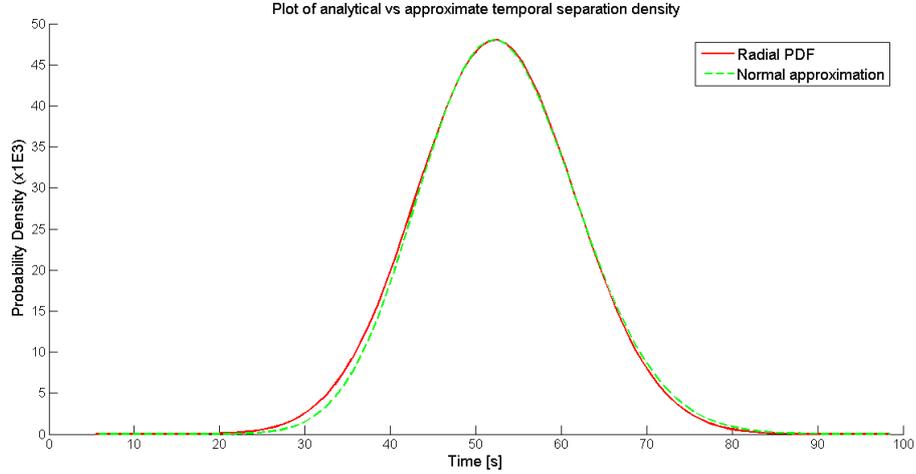


Figure 5.8: Comparison of analytical and approximate radial distribution with large mean relative to variance

### 5.2.3 Time to Point of Closest Approach (TPCA)

The concept of **TPCA** applies to known trajectories as it refers to the time at which the vehicles are at their minimum separation. For such known trajectories the **TPCA** may be found analytically. For uncertain trajectories, however, the **TPCA** actually refers to the time at which a particular level of certainty corresponds to the shortest distance between the vehicles. The level of certainty is obtained by integrating under the **PDF** to produce a **CDF**. Figure 5.9 illustrates an example of **TPCA** for uncertain trajectories and Figure 5.10 illustrates the **TPCA CDFs** for the **UAS** and traffic aircraft. It is clear that the **TPCA** (for a certainty level of 20%) occurs at  $t = 45s$ , denoted  $t_{P=0.2} = 45s$ .

#### Linear **TPCA**

Calculation of the **TPCA** in this way is computationally expensive as the spatial projection for all projection times up to some horizon must be evaluated and the minimum found. An alternative method may be employed if the flight paths of both aircraft concerned are linear. If both trajectories were linear and known

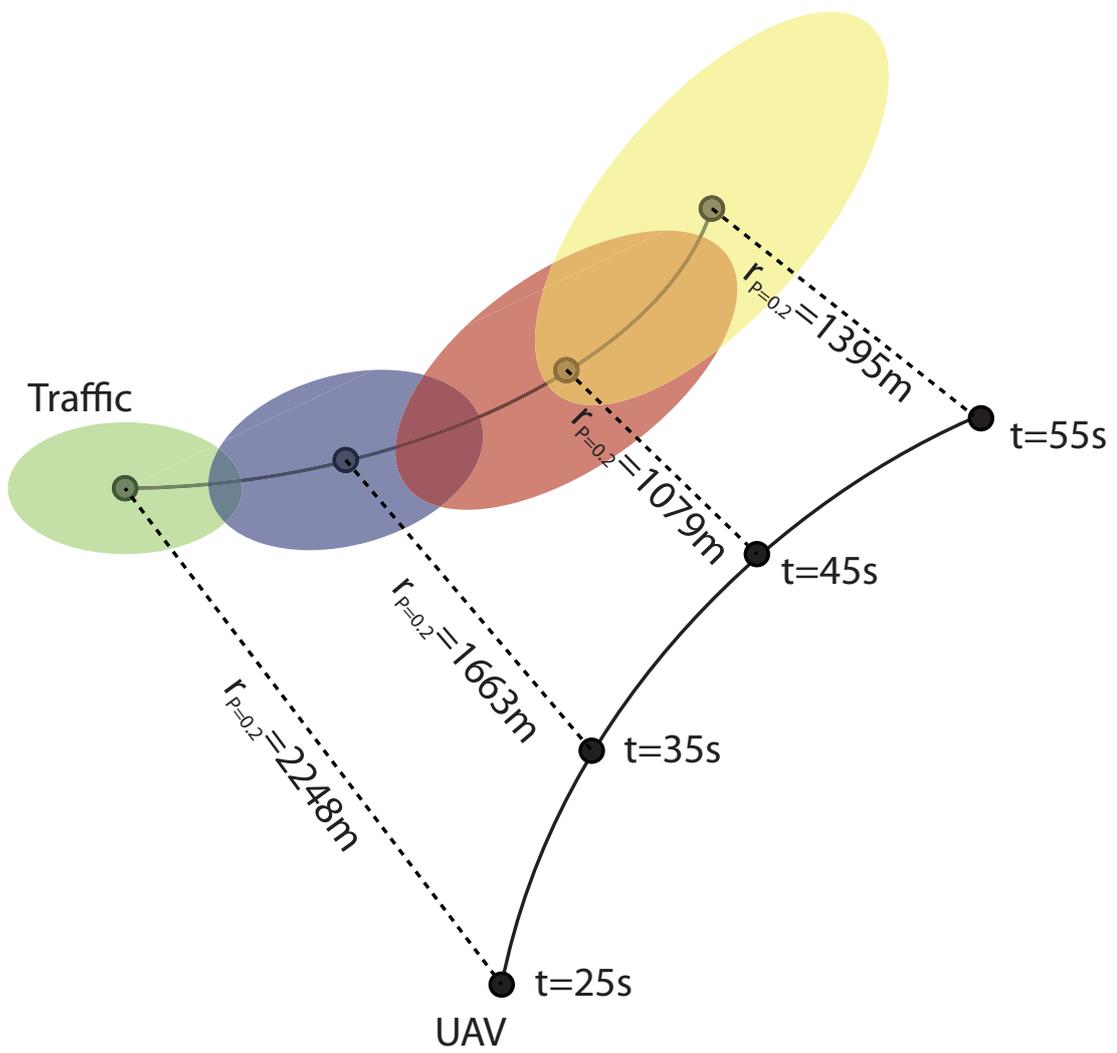


Figure 5.9: TPCA example

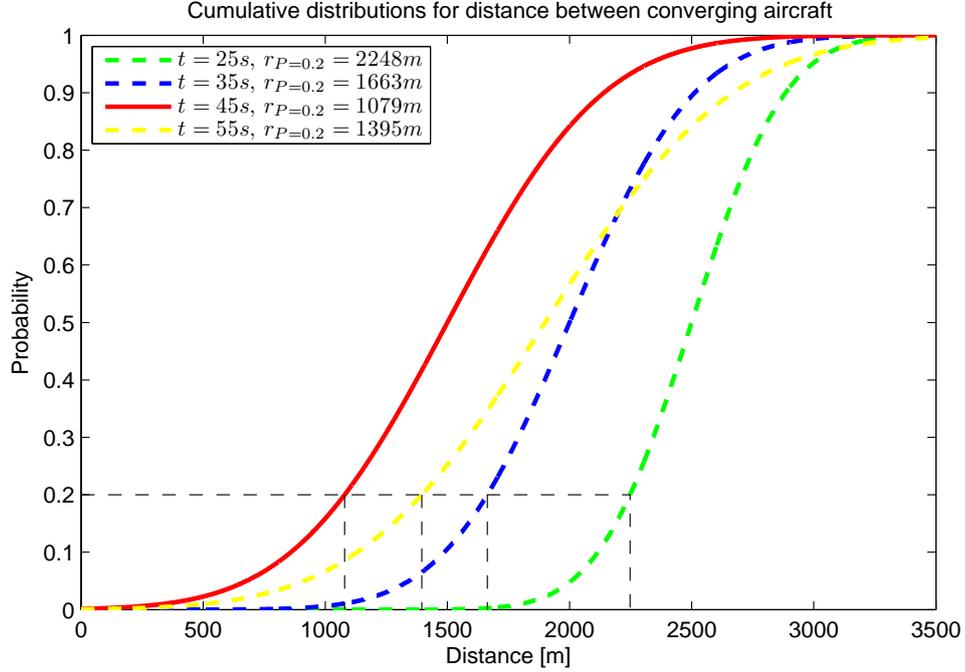


Figure 5.10: Cumulative distribution plot of distance for converging aircraft

absolutely, the [TPCA](#) could be evaluated as

$$t_{PCA} = \frac{-X_0 \cdot V}{|V|^2} \quad (5.15)$$

where  $X_0$  is the initial relative position vector of the traffic to the [UAS](#) and  $V$  the relative velocity vector.

If  $X_0$  is represented by a Gaussian [BVPDF](#) (but assuming  $V$  is known), then the dot product in the numerator of (5.15) can be evaluated as a univariate [PDF](#), leading to

$$t_{PCA} \sim \mathcal{N}(\bar{V} \cdot \hat{X}_0, \bar{V}^T \tilde{X}_0 \bar{V}) \quad (5.16)$$

where  $\bar{V} = V/|V|^2$  is the normalised relative velocity vector,  $\hat{X}_0$  is the mean initial relative position and  $\tilde{X}_0$  is its variance. By utilising an inverse [CDF](#) the [TPCA](#) to any probability value may be evaluated from (5.16).

This analytical calculation of the [TPCA](#) has limited applicability as it assumes the relative velocity is known accurately (or at least with significantly higher

---

accuracy than the position) and is constant (i.e. the path is linear and the flight unaccelerated). Additionally, it is assumed that from the time that the measurement  $X_0$  was made the covariance matrix  $\tilde{X}_0$  remains constant. Despite these limitations, (5.16) may be used in addition to full spatial projection to calculate an estimate of the **TPCA**. For example, spatial projection may be conducted at intervals of 10s with the **TPCA** between these intervals evaluated directly from (5.16).

#### 5.2.4 Distance at **TPCA**

Based on a **TPCA** calculated from the previous section it is trivial to determine the distance at which this occurs. If the **TPCA** has been evaluated from spatial projection the distance is already known (see Figure 5.10). If the **TPCA** was evaluated from (5.16) then the velocity vector can be used along with the initial position and  $t_{PCA}$  to evaluate the distance.

### 5.3 Non-Gaussian projection

The previous derivations of separation metrics has focused on linear projection resulting in Gaussian **PDFs**. In the case of **CPDU**, the **PDFs** describing traffic position are non-Gaussian and as such require a different approach. This section covers the numerical methods available to calculate the non-Gaussian separation metrics in addition to approximations which can be made to reduce the computational complexity of the problem.

#### 5.3.1 Distance between vehicles

To calculate the distance between the **UAS** and a traffic aircraft whose position is described by a **CPDU PDF**, the distribution must be discretised in the global frame of reference. Firstly, a random sample of  $N$  points is taken in the curvilinear frame ( $C_i$  for  $i = 1 : N$ ), distributed by the **CPDU PDF**. For each of these points the corresponding probability density  $f_i$  is found from the **CPDU PDF** and the equivalent global location  $X_i$  from (4.16). From  $X_i$ , with knowledge of the **UAS**

---

position  $X_{UAS}$  the distance can be found as

$$r_i = |X_{UAS} - X_i| \quad (5.17)$$

where  $|\bullet|$  denotes the Euclidean norm

The pairs of distance and corresponding probability density  $\{r_i, f_i\}$  must then be sorted in ascending order of  $r_i$ . The unnormalised CDF is given as

$$P'(r \leq r_j) = \sum_{i=1}^j f_i \quad (5.18)$$

This is normalised by

$$\alpha = \sum_{i=1}^N P'(r \leq r_i) \quad (5.19)$$

therefore

$$P(r \leq r_i) = \frac{1}{\alpha} P'(r \leq r_i) \quad i \in \{1, \dots, N\} \quad (5.20)$$

It is preferable not to store  $N$  points to characterise the CDF therefore a Gaussian approximation is performed. The value of  $r$  corresponding to probabilities of 0.25, 0.5 and 0.75 are interpolated from the discrete CDF (5.20), denoted  $r_{P=0.25}$ ,  $r_{P=0.5}$  and  $r_{P=0.75}$  respectively. Assuming a Gaussian CDF, the mean is equal to the median and can then be found directly

$$\hat{r} = r_{P=0.5} \quad (5.21)$$

The standard deviation may be calculated from Theorem 8.7.5 in [Patel and Read \[1982\]](#), by observing that

$$\frac{1}{2}(r_{P=0.75} - r_{P=0.25}) = 0.6744898\tilde{r} \quad (5.22)$$

where  $r_{P=0.75} - r_{P=0.25}$  is the Inter-Quartile Range (IQR) and  $\tilde{r}$  the standard deviation. Therefore

$$\tilde{r} = \frac{r_{P=0.75} - r_{P=0.25}}{1.3489796} \quad (5.23)$$

In order to demonstrate the performance of both the numerical ( $P(r \leq r_i)$ )

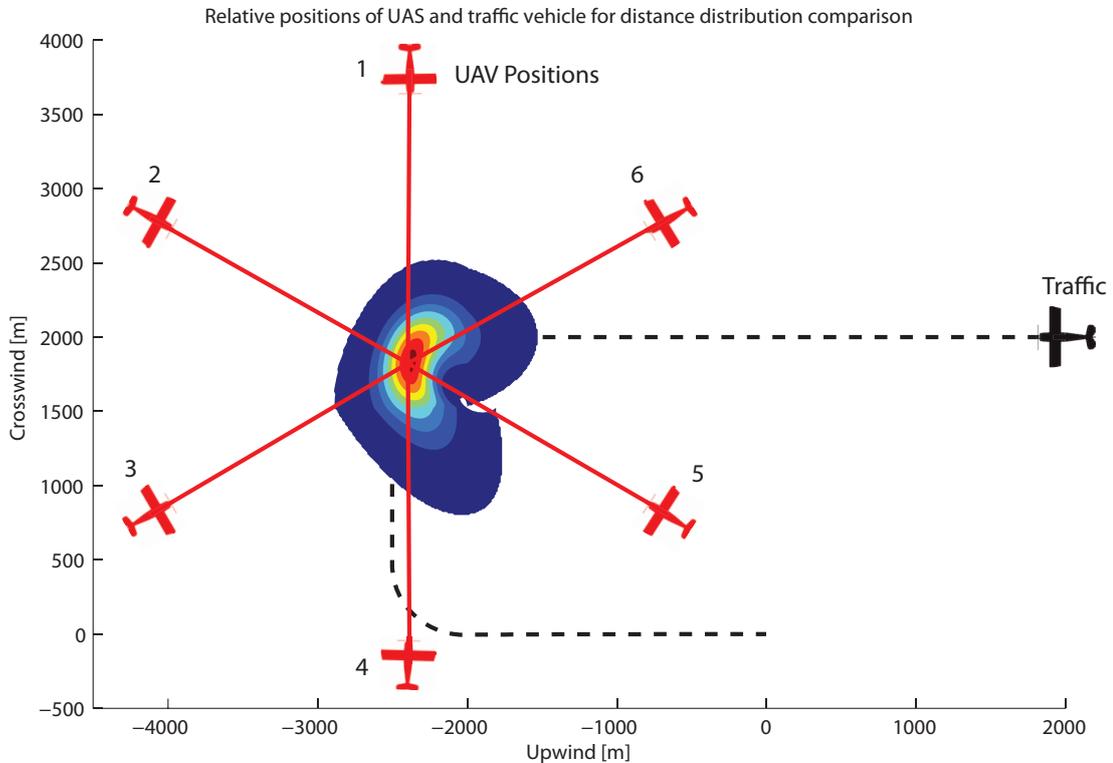


Figure 5.11: Positions of UAS relative to traffic

given above, with  $N = 1000$ ) and approximate Gaussian distributions, a simulation study was performed which placed the UAS at six different location relative to the projected traffic, Figure 5.11. Figure 5.12 shows plots of the numerical and Gaussian CDFs at each UAS location compared with a Monte Carlo simulation of 1000 executions. It can be seen that the PMF follows the Monte Carlo simulation closely, this is illustrated further by Table 5.1 where a positive error represents a projected distance greater than the Monte Carlo result. In the majority of cases the error is smaller than  $100m$  which is of sufficient accuracy for self separation. Performance can be improved by increasing the number of samples  $N$ , however this is a trade off with computational time. The mean time to calculate a single distribution on a mid-range performance computer ( $\sim 7.5$  GFLOPS) was  $0.21s$ , this increased to  $0.75s$  when  $N$  was increased to  $5000$ . These computational times are significant in comparison with the Gaussian distributions discussed previously which require only  $0.01s$  to calculate.

Cumulative distribution comparison of numerical and Gaussian approximated distance calculations with Monte Carlo simulation

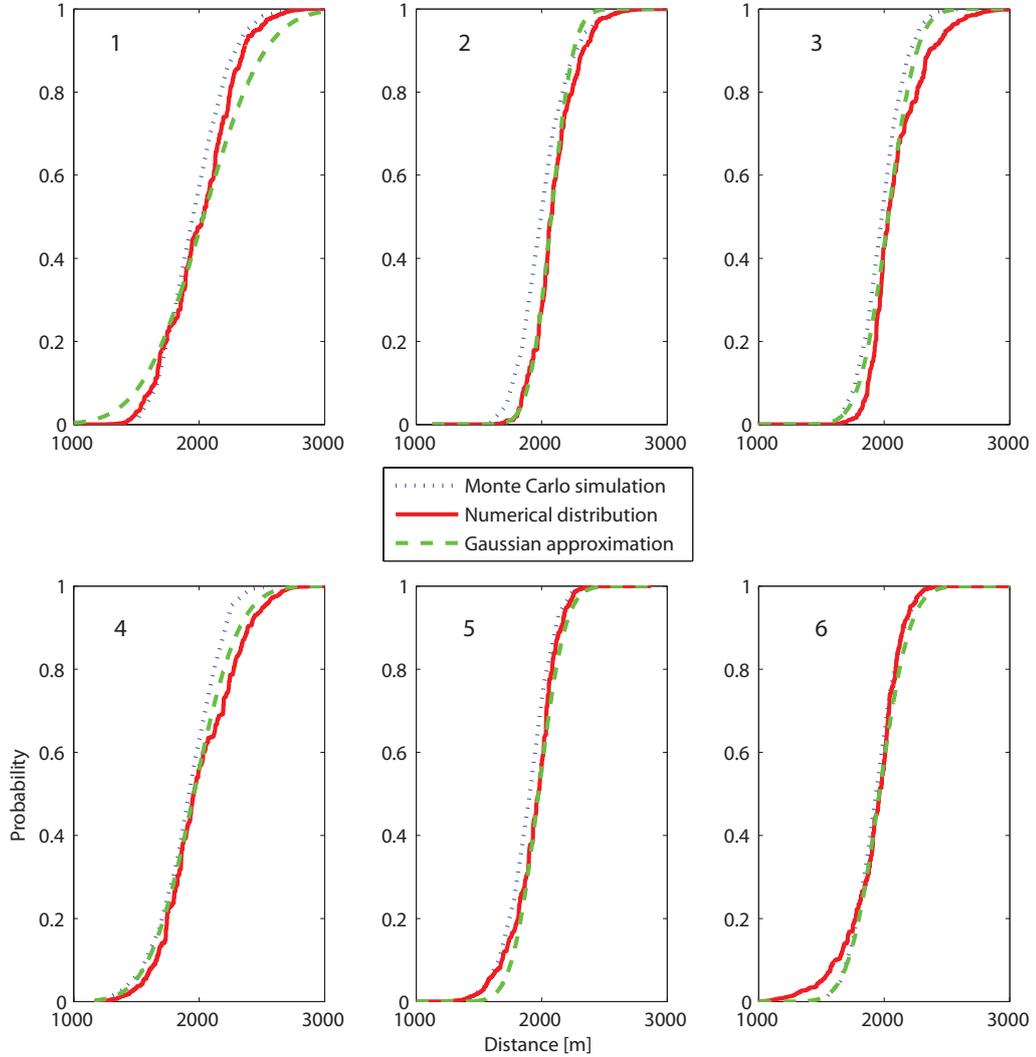


Figure 5.12: Comparison of numerical CDF against Monte Carlo simulation (1000 executions)

Table 5.1: Distance error at  $P = 0.2$

Position	Numerical Error	Gaussian Error
1	$-35m$	$-9m$
2	$108m$	$117m$
3	$35m$	$89m$
4	$15m$	$39m$
5	$92m$	$-69m$
6	$17m$	$-8m$

---

### 5.3.2 Time separation

The calculation of time separation for non-Gaussian distributions is analogous to that for Gaussian distributions, modifying the parameters of the distribution to represent time rather than distance. Specifically (5.20) is modified to

$$P\left(t \leq \frac{|X_{UAS} - X_i|}{V_{UAS}}\right) = \frac{1}{\alpha} P'(tV_{UAS} \leq |X_{UAS} - X_i|) \quad i \in \{1, \dots, N\} \quad (5.24)$$

where  $V_{UAS}$  is the speed of the UAS,  $P(\bullet)$  and  $\alpha$  are identical to (5.18) and (5.19) respectively.

The worst case distance error shown in Table 5.1 corresponds to a time error of 2.34s which is less than 10% of the proposed minimum time separation of 30s.

### 5.3.3 TPCA and distance at TPCA

The separation metrics relating to TPCA are both derived from the CDF in exactly the same way as for the Gaussian case. Unlike the Gaussian case, however, it is not possible to make a linear approximation to simplify calculation under certain circumstances.

## 5.4 Decision making

The previous sections have covered the quantification of separation between two aircraft in general. Due to the number of metrics available and their varying applicability in different situations, it is not immediately clear how they can form the basis of a self separation system for an UAS. To demonstrate the improvements in safety and efficiency which can be gained by artificial situation awareness, it is necessary to implement a decision making system to perform self separation of an UAS from terminal area traffic. Whilst this will be used to demonstrate the utility of the proposed methods, it is not the primary focus of the work and therefore a simple rule based system is used.

---

### 5.4.1 Circuit constraints

The terminal area at an uncontrolled airfield<sup>1</sup> typically follows a standard circuit pattern which all arriving aircraft must join. Figure 5.13 illustrates a typical left hand circuit and associated joining procedures from [Civil Aviation Authority \[2009b\]](#). Airfields not equipped with radios typically restrict arriving aircraft to using only the *overhead* join (often referred to as the Standard Overhead Join (OHJ)) as this allows pilots to overfly both the circuit and the airfield to observe other traffic and the prevailing wind direction. Airfields equipped with radios often do not use (or do not mandate) OHJs as they are less efficient and there is some evidence to suggest they increase the chances of mid-air collisions due to all aircraft converging on a small patch of airspace [[Civil Aviation Authority of New Zealand, 1999](#); [Transportation Accident Investigation Commission, 2008](#)]. At such airfields it is possible for arriving aircraft to receive all of the requisite information regarding the airfield and traffic from either the radio operator or directly from the pilots of the other aircraft. Additional reasons may also exist for not permitting overhead joins, for example if both circuit directions are in use simultaneously, if gliders are being launched via a winch cable or if sky diving is in progress [[Civil Aviation Authority, 2011b](#)]. Typical circuit joining procedures for an airfield which does not use overhead joins are illustrated in Figure 5.14 [[Civil Aviation Authority, 2009a](#)], these will be used for this work.

An UAS joining the circuit will first choose its joining point based upon Figure 5.14. Any traffic observed to be joining the circuit will also be assumed to be following Figure 5.14. The UAS then has some freedom to manoeuvre in order to separate itself from any observed traffic [[Civil Aviation Authority, 2009a](#)], these manoeuvres can be broken down into two groups.

#### 1. Pre Circuit

- Delay - Perform an orbit (360° turn) well clear of the circuit and other joining traffic
- Delay - Decrease speed to joining point

---

<sup>1</sup>An ‘uncontrolled’ airfield is defined as one which may or may not have a radio but whose radio operators do not have authority to issue instructions to aircraft, they may only transmit information and guidance

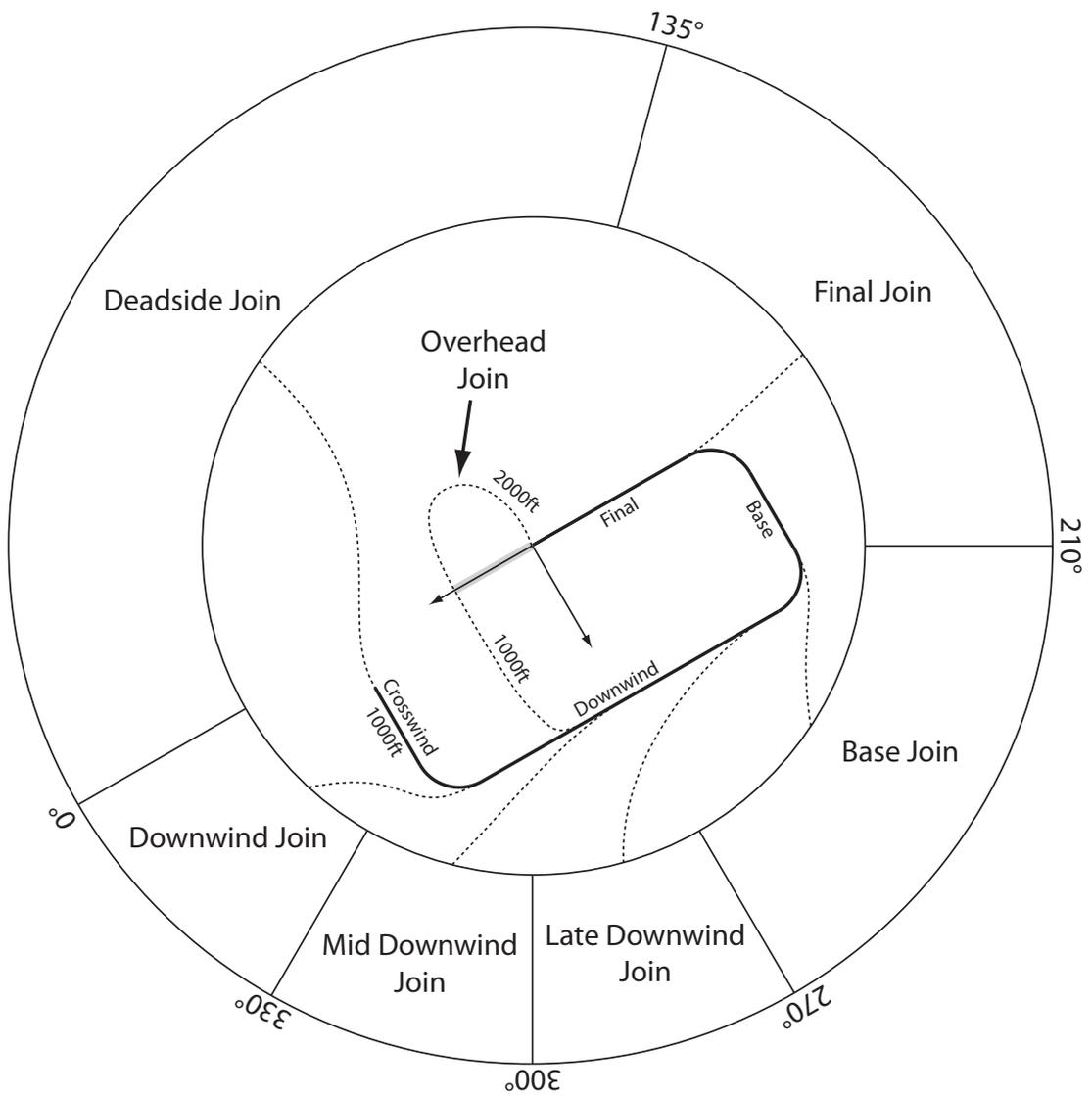


Figure 5.13: Typical circuit joins (derived from [Civil Aviation Safety Authority \[2010\]](#))

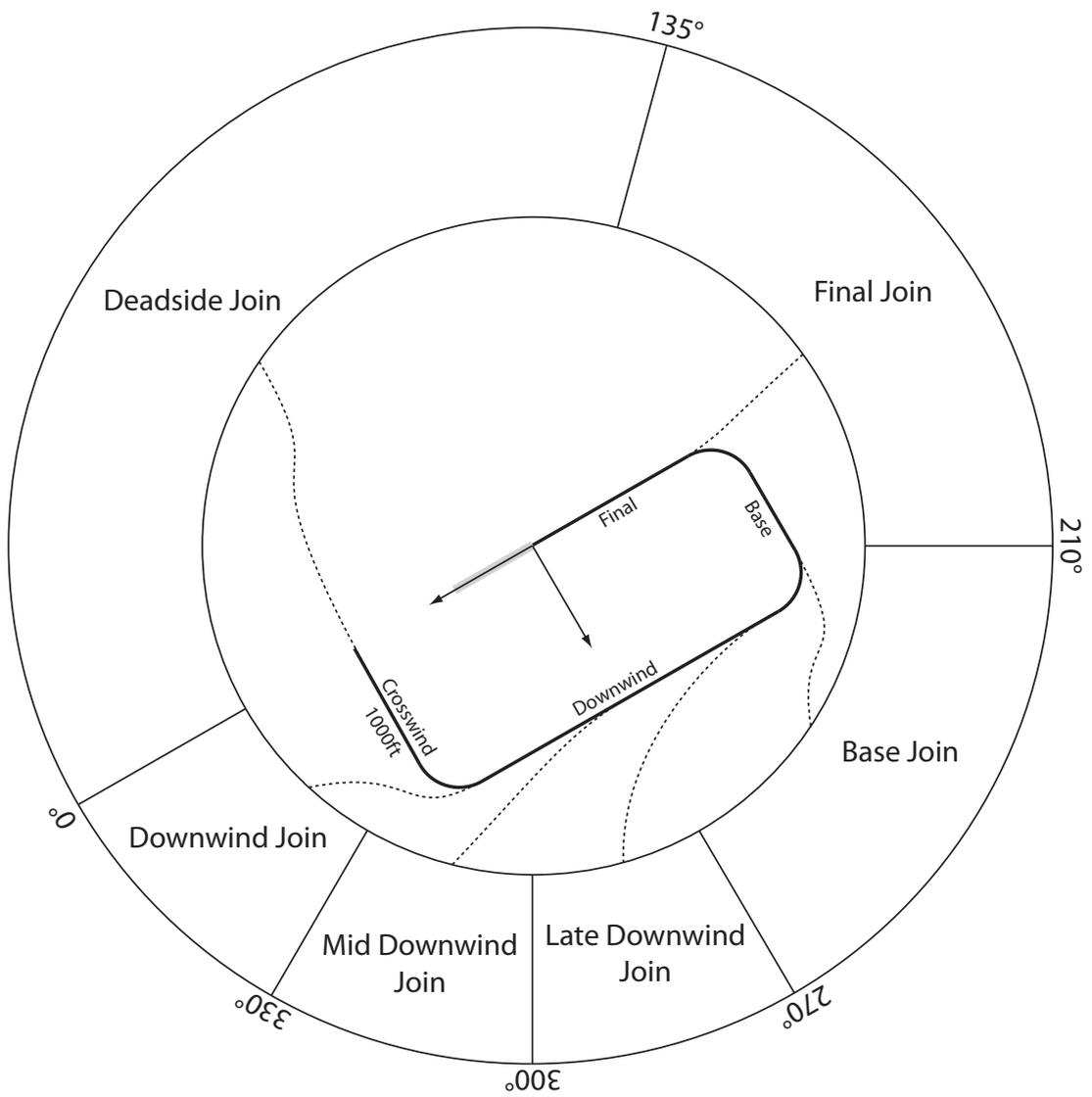


Figure 5.14: Typical circuit joins excluding the overhead (derived from [Civil Aviation Safety Authority \[2010\]](#))

- 
- Expedite - Increase speed to joining point

## 2. In Circuit

- Delay - Increase circuit dimensions
- Delay - Decrease circuit speed
- Expedite - Decrease circuit dimensions
- Expedite - Increase circuit speed

It is also permitted for aircraft to perform orbits whilst in the circuit, however this is usually reserved for controlled airfields and left to the discretion of [ATC](#) [[Civil Aviation Authority, 2009a, 2011b](#)].

Only certain variations in circuit dimensions are permitted, and whilst these are not defined rigidly, [Figure 5.15](#) illustrates the values to be used. The down-wind leg can not be moved closer to the runway as this would not leave sufficient time to descend along the base and final legs.

### 5.4.2 Self separation rules

Prior to entering the circuit the primary concern of the [UAS](#) must be separation to other joining aircraft whilst also determining its place in the landing sequence. Separation from circuit traffic and traffic joining at the same point as the [UAS](#) should be quantified by time separation to facilitate landing sequence separation. For traffic joining at other circuit points, a distance at [TPCA](#) is employed to ensure separation prior to circuit entry, with time separation being used once either vehicle enters the circuit.

[Figure 5.16](#) illustrates the decision making process followed by the [UAS](#) to ensure separation is maintained whilst outside the circuit. If the [UAS](#) determines a traffic vehicle is joining by a different point than itself (i.e. it is not presently in the circuit and is not co-joining) it must determine its Distance at Point of Closest Approach ([DPCA](#)). A sufficient value for [DPCA](#) may be 500m, however this depends on factors such as aircraft type and manoeuvrability (for example a greater separation would be required if the traffic was a large, wake producing, vehicle). If the [DPCA](#) is insufficient the [UAS](#) can only orbit to improve separation

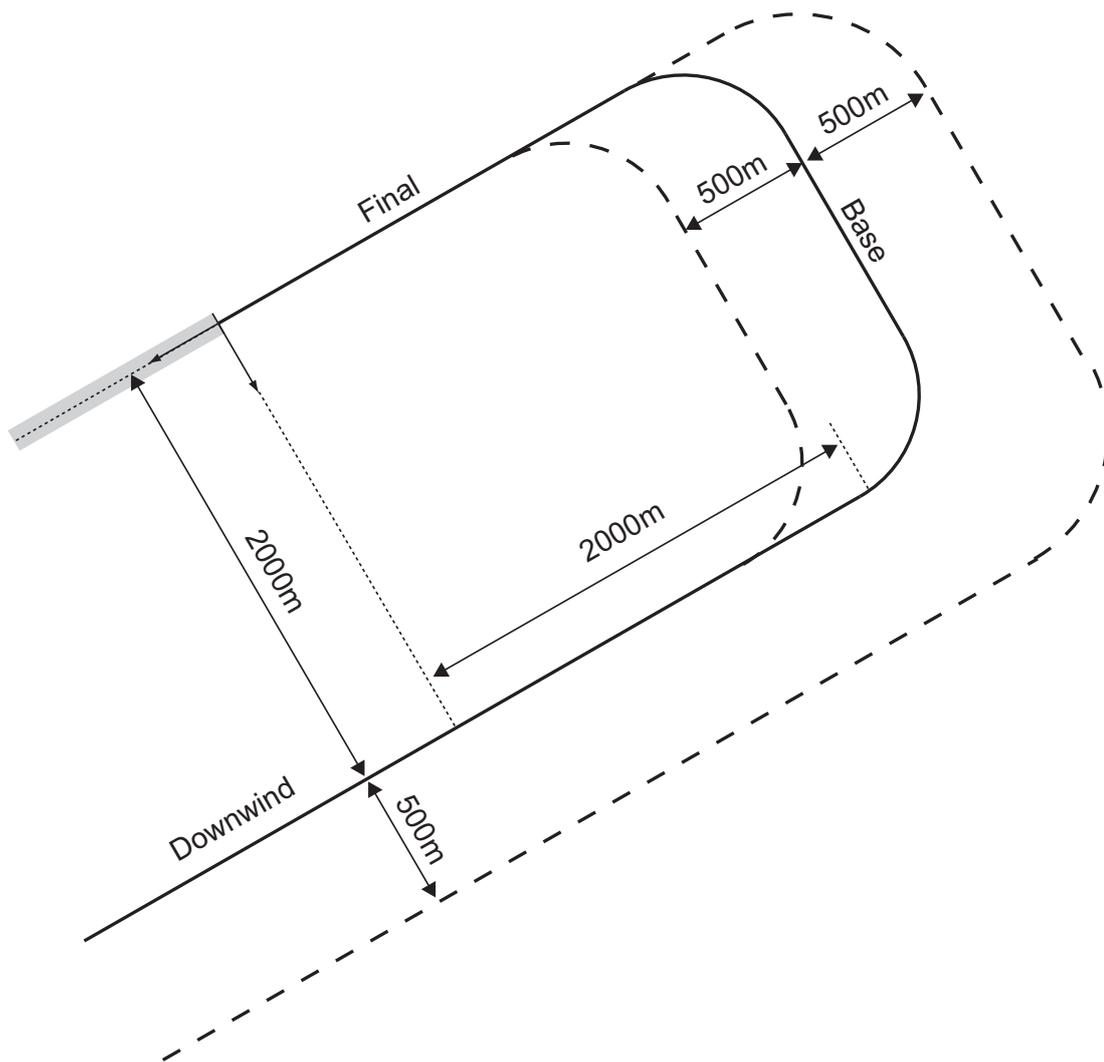


Figure 5.15: Allowable circuit variations

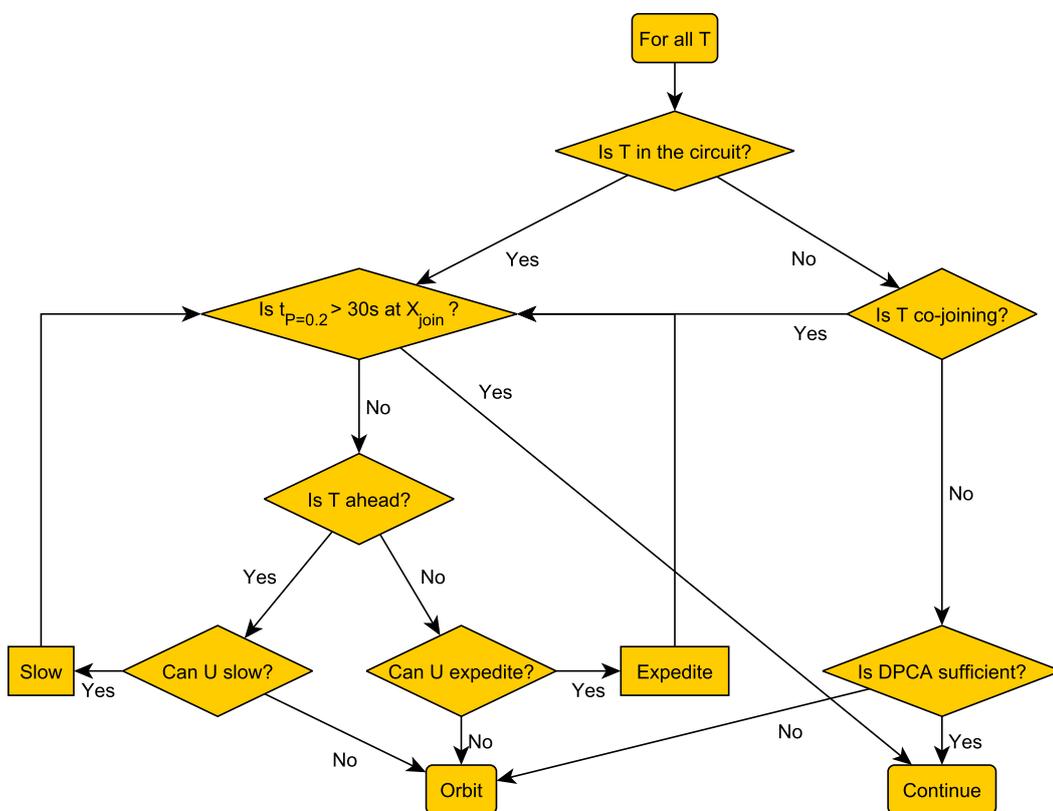


Figure 5.16: Decision making process for a UAS outside of the circuit. T: Traffic, U: UAS

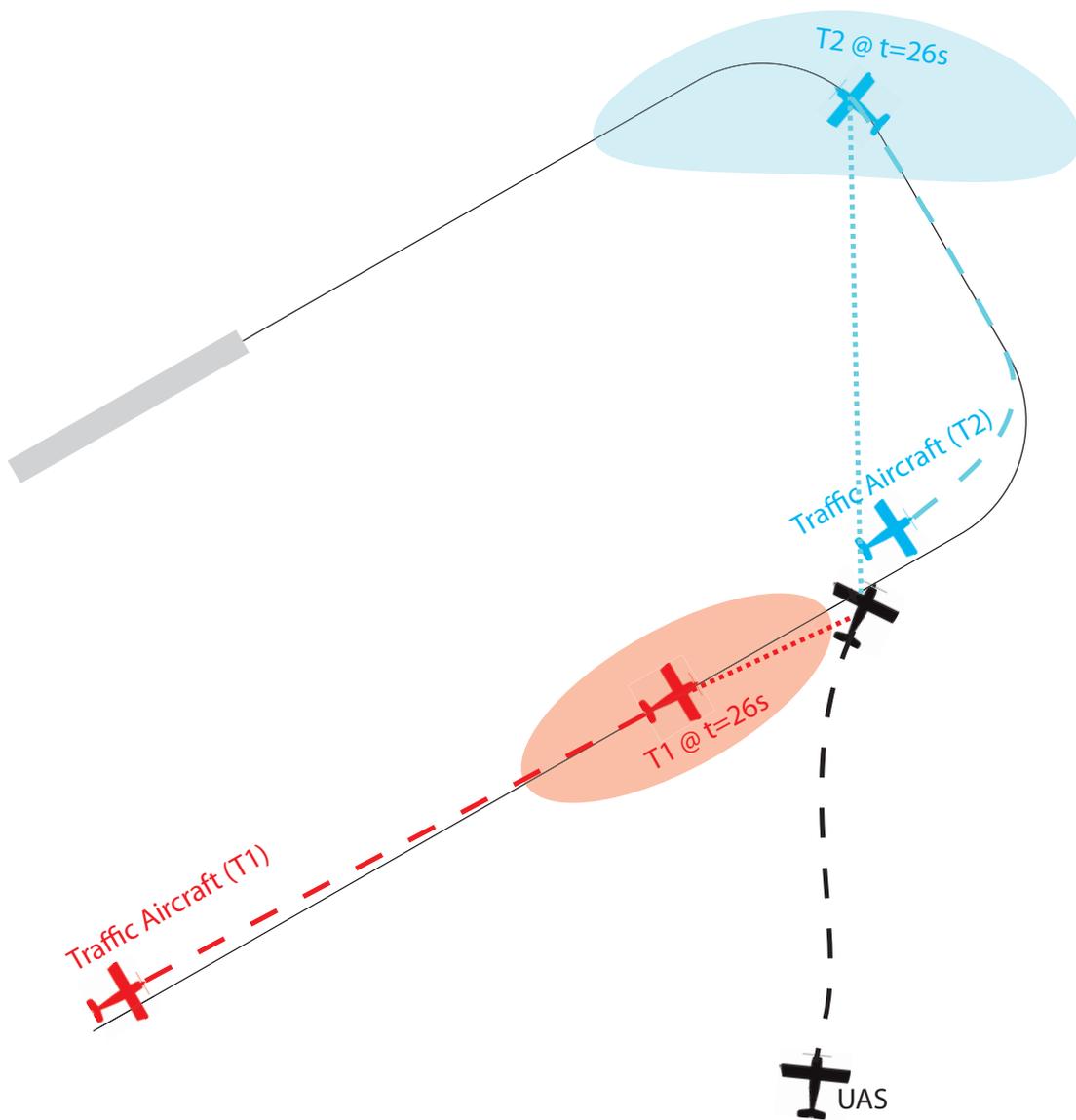


Figure 5.17: Example of UAS joining the circuit

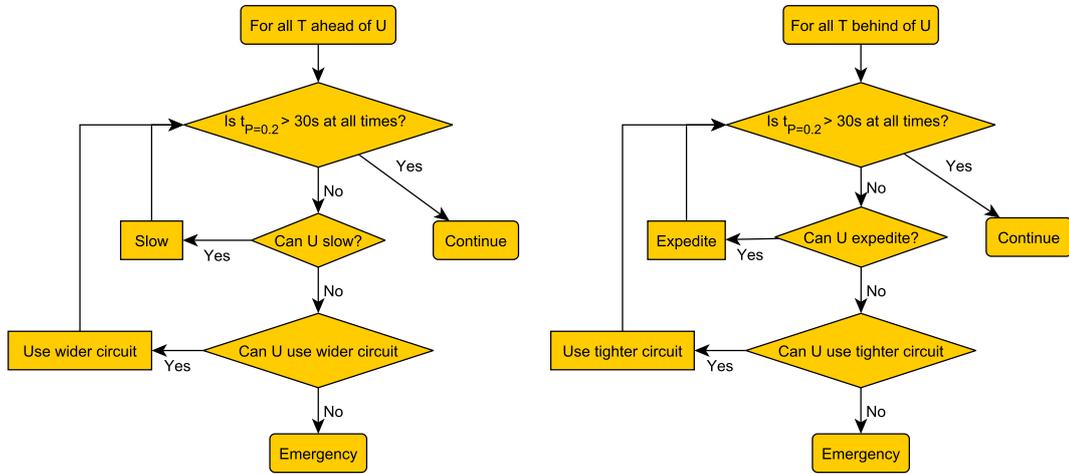


Figure 5.18: Decision making process for a UAS inside the circuit. T: Traffic, U: UAS

as it is not immediately obvious whether increasing or decreasing speed would be advantageous as both vehicles are following different courses.

For vehicles in the circuit or co-joining, the UAS is concerned with the time separation at the point where it will enter the circuit  $X_{join}$ . If there is a significant probability ( $P > 20\%$ ) that this separation will be insufficient to enable the leading vehicle to clear the runway before the following vehicle then the separation must be increased so as to avoid a go-around manoeuvre. If the traffic vehicle is ahead of the UAS (i.e. it is closer to the runway), the UAS must slow down to improve separation. If the UAS is already operating at its minimum speed it must orbit instead. For traffic behind the UAS, the UAS should attempt to expedite (speed up) but if it can not then it must orbit and not attempt to enter the circuit ahead of the traffic.

An example of this circuit joining separation is shown in Figure 5.17 where two traffic vehicles (T1 and T2) are already in the circuit. The UAS determines that it will enter the circuit in 26s (by projecting its own, presumed known, path), it then projects the positions of the traffic vehicles at this time and subsequently calculates the time separation for that projection. Since both vehicles are separated by over 30s the UAS will continue on its current path and perform the calculation again periodically.

---

Figure 5.18 shows a similar decision making process for the UAS inside the circuit, dealing with aircraft ahead and behind of the UAS separately. When widening or tightening the circuit the UAS is only permitted to operate at the outer and inner extents illustrated in Figure 5.15. The process prioritises changes in UAS velocity over changes of flight path so as to not maneuver needlessly causing possible confusing for traffic aircraft. The emergency action in the case that there is no suitable solution is the referral of control back to the UAS pilot.

### 5.4.3 Practical considerations

Due to the computational complexity in the calculation of the non-Gaussian separation metrics, it is not possible to continuously determine separation from all vehicles at a high rate. Traffic outside of the circuit are most likely to route directly toward their most convenient joining location so as to minimise their flight time. This fact can be used to simplify the time separation and DPCA calculation as with a linear flight path the projection PDF remains Gaussian therefore the distance calculation becomes analytical. To simplify the calculation process when the UAS is established in the circuit, all traffic vehicles will be ignored with the exception of the aircraft immediately preceding the UAS. This is a valid simplification because aircraft behind the UAS or otherwise outside the circuit are obliged to give way. Likewise, the aircraft immediately ahead of the UAS must give way to any aircraft ahead of it causing the UAS to do the same.

When in the circuit and calculating the time separation to the aircraft ahead it is the minimum value that is of interest. This requires the projection PDF and separation CDF to be evaluated at a number of projection times up to some horizon. To minimise the computational burden of this, it will only be performed at a low rate of around 0.1Hz.

## 5.5 Summary

This section has investigated the means by which artificial situation awareness may be used to perform self separation of an autonomous UAS from traffic vehicles in a terminal area circuit pattern. A number of different separation metrics have

---

been introduced and examples where each are applicable given. Calculations of these metrics from uncertain traffic information have been shown to be non-trivial for even the simplest Gaussian cases. Approximations were necessary to calculate separation distributions from Gaussian position distributions. These break down under certain situations but this has been shown to occur well within the emergency manoeuvre threshold which is beyond the scope of this thesis.

The non-Gaussian, CPDU, case introduces significant computational complexities due to the numerical steps involved in determining a separation distribution from a position distribution. Approximations have been made where appropriate to reduce the computation and memory burden of these calculations, however a computational time of 0.21s is still required to calculate a single distribution.

The two metrics most suited to terminal area operations are time separation and DPCA. Time separation provides the UAS with a means of separating from traffic sufficiently to enable the lead vehicle to clear the runway before the following vehicle attempts to land. It is also trivial to modify the time separation criteria to account for wake separation requirements (simply increase the time separation required). DPCA is used prior to entering the circuit to ensure separation is not lost with other joining aircraft.

A rule based decision making framework based upon standard circuit procedures has been developed to give an autonomous UAS freedom to alter its trajectory in order to maximise separation. These rules have been designed to minimise the number of separation distribution calculations so as to allow the system to run in a real time fashion. Nothing in the system design limits the number of traffic vehicles which a UAS can deal with, however the computational burden increases in direct proportion to the number of vehicles.

The self separation decision making system designed in this chapter enables the utility of artificial situation awareness developed previously to be tested. The remaining chapters of this thesis detail the experimental test bed and results obtained.

# Chapter 6

## Test Environment

### 6.1 Introduction

The previous chapters have detailed the development of an artificial situation awareness and self separation system for an autonomous [UAS](#) operating in the terminal area of an uncontrolled airfield. The testing of methods and algorithms has so far been limited to a MATLAB/Simulink environment as this allows for rapid prototyping of mathematical and logical techniques. The proposed system, however, is designed for implementation on real-world [UAS](#) which introduces a number of additional challenges over and above this simple test environment.

Firstly, little consideration has been given to the dynamics of the [UAS](#) as only a simple kinematic model has been used. As soon as dynamics is considered, however, it is no longer possible to directly control the flight path of a vehicle as only its attitude may be effected by control actuation. A higher fidelity vehicle model also requires an autopilot system to abstract the flight path commands from direct attitude control.

When considering a higher fidelity model and autopilot, it is necessary to bear in mind the applicability to a real-world system. This chapter details the high fidelity test environment developed during the course of this project to facilitate both simulation and small scale real-world test. Extensive use of [COTS](#) products has been made to accelerate development times.

Figure [6.1](#) presents an expanded version of Figure [1.1](#) which highlights how

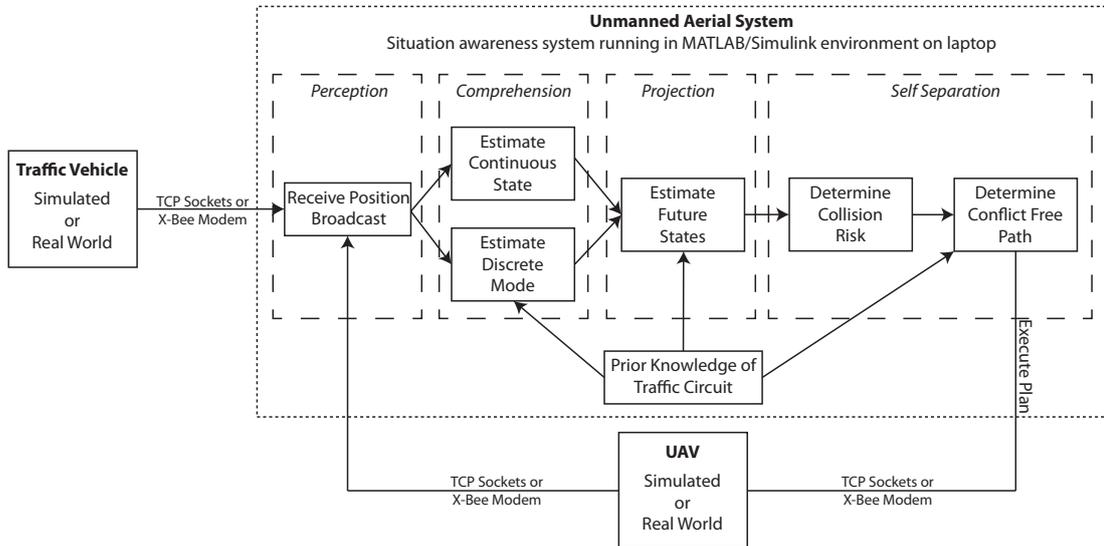


Figure 6.1: Overview of the systems presented in this thesis within the test environment context

the artificial situation awareness, self separation and decision making systems developed in this thesis fit in to the test environment discussed in this chapter.

The next section provides an overview of the test environment and how it is used in development of autonomous functions for UASs. Section 6.3 details the COTS autopilot system and Section 6.4 covers its communication protocol. Sections 6.5 and 6.6 discuss the simulation environment and MATLAB/Simulink interface respectively. Finally, Sections 6.7 and 6.8 detail the real world test bed and provide some basic test results of the system.

## 6.2 Environment overview

Three COTS systems form the basis of the test environment: The X-Plane flight simulator, the ArduPilot Mega (APM) autopilot system and MATLAB/Simulink. These systems are used during the three phases of development employed by the test environment.

- Software In the Loop (SIL)
  - No physical equipment required

- 
- X-Plane used to simulate vehicle dynamics
  - [APM](#) software used to stabilise vehicle
  - MATLAB/Simulink used to prototype algorithms
  - Hardware In the Loop ([HIL](#))
    - No physical vehicle required
    - X-Plane used to simulate vehicle dynamics
    - [APM](#) hardware used to stabilise vehicle
    - MATLAB/Simulink used to prototype algorithms
  - Real World
    - Small aircraft platform used
    - [APM](#) hardware used to stabilise vehicle
    - MATLAB/Simulink used to prototype algorithms

The [SIL](#) phase is used to prototype brand new functions with minimal cost and no risk to real hardware. [HIL](#) represents an increase in cost whilst improving the realism of the test. Finally, real world testing demonstrates functions on a small scale aerial platform, providing the highest fidelity results possible at the fundamental research level.

The use of [COTS](#) systems to provide the core functionality of the test environment greatly simplifies its development. It is still necessary, however, to integrate the [COTS](#) systems together to provide the complete environment. Figure 6.2 illustrates the integration of the [COTS](#) components into the test environment, items in green were developed to facilitate integration. The following sections discuss this development in detail.

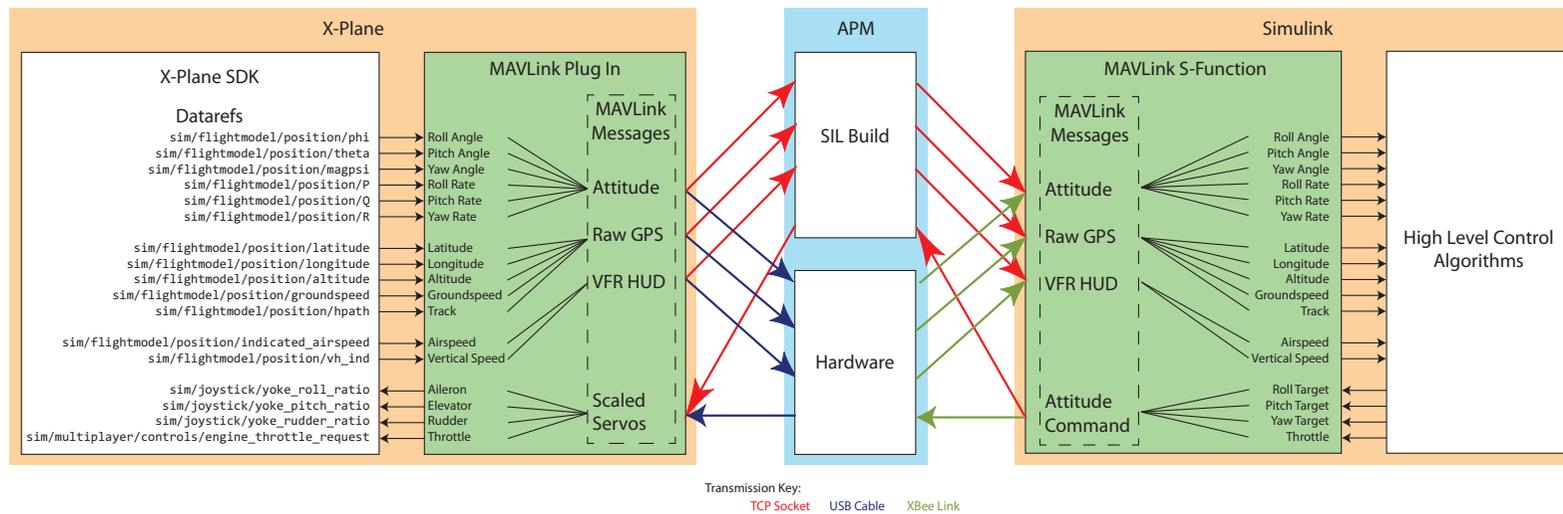


Figure 6.2: X-Plane, APM, Simulink integration

---

## 6.3 Autopilot system - ArduPilot Mega (**APM**)

The central component of Figure 6.2 is the ArduPilot Mega (**APM**), an open source **COTS** autopilot system for small unmanned aircraft designed by the hobbyist community [Anderson, 2010]. It consists of a sensor board containing 3-axis Micro Electromechanical Sensor (**MEMS**) accelerometers and gyroscopes, a static pressure sensor and an optional magnetometer. A **GPS** receiver and dynamic pressure sensor are also attached to this board to provide outdoor flight capability. The autopilot software runs on a separate circuit board containing an Atmel ATmega 2560 processor running the Arduino bootloader. This board incorporates a failsafe function to provide a manual override of all control signals (via conventional Radio Controlled (**RC**) equipment) in the event of autopilot failure. The sensor (blue) and autopilot (red) boards can be seen in Figure 6.3.

The **APM** software is written in C++ and is completely open source allowing for straightforward development of advanced functions. The software has been optimised for a number of different aerial platforms including fixed-wing, helicopter and multi-rotor (tri-, quad-, hex- and octirotor) vehicles in addition to ground vehicles and boats. The initial test environment detailed here focuses only on fixed-wing platforms, but this is not a fundamental limitation of the system simply a limitation of scope.

### 6.3.1 Communication

It is possible to pre-program **APM** with a mission which is then enabled by switching to the automatic mode using an **RC** transmitter. For more advanced operation, however, a real time data link between the vehicle and the ground is desirable. This functionality is provided by utilising a pair of XBee Radio Frequency (**RF**) modules as a wireless serial link. XBee modules operate at 2.4GHz using the ZigBee communications protocol which is a robust, long range, serial data link. The maximum data rate supported by XBee is 115,200bps, however **APM** is configured to use 57,600bps to improve data integrity (due to the low speed nature of the ATmega processor).

By installing an XBee module in the vehicle and connecting another to a laptop a real time telemetry link is established. Ground station software is provided

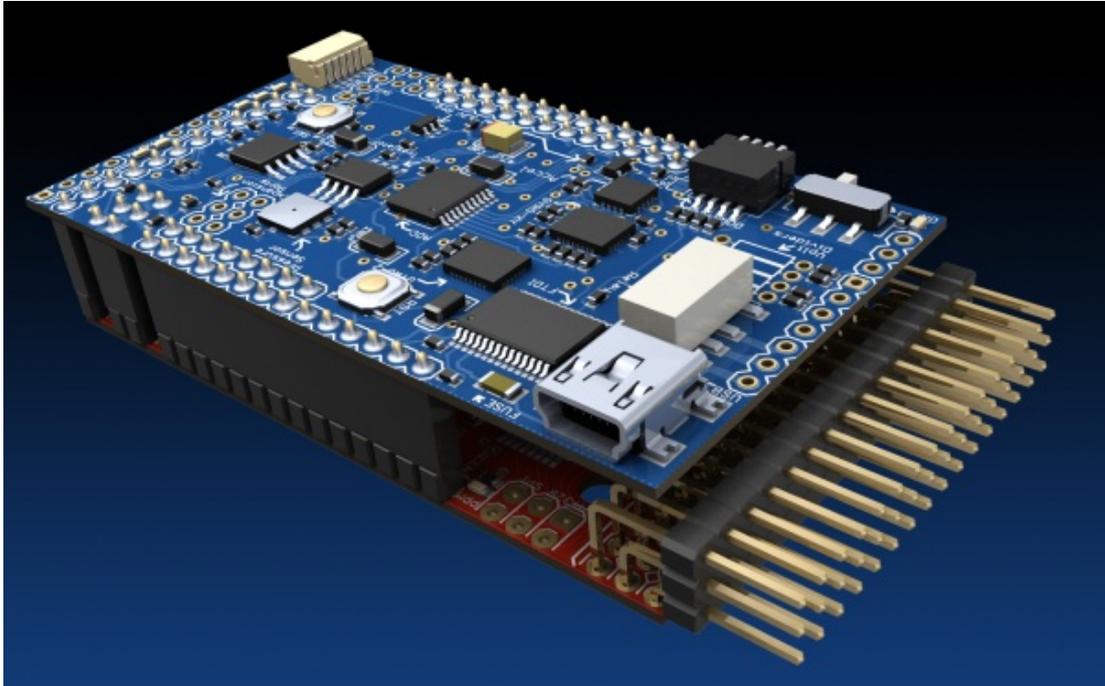


Figure 6.3: The APM autopilot - <http://code.google.com/p/ardupilot-mega/wiki/HardwareDetails>



Figure 6.4: The APM ground station - <http://code.google.com/p/ardupilot-mega/wiki/Mission>

---

by [APM](#), Figure 6.4, and can be used to monitor the flight, update the mission or ever modify the autopilot parameters (for example control gains or saturations) in flight. It is possible to replace the default ground station with any software capable of communicating over a serial connection, this enables advanced [UAS](#) functionality to be prototyped on a laptop and control a real aircraft in flight.

### 6.3.2 Autopilot functions

In its default configuration [APM](#) provides a number of functions desirable for a test environment. The most commonly used modes are

- Manual
  - The pilot has complete control through conventional [RC](#) equipment, all autopilot function is disabled
- Stability Augmentation System ([SAS](#))<sup>1</sup>
  - The pilot commands angular rates through conventional [RC](#) equipment, input is saturated
- Carefree Handling ([CFH](#))<sup>2</sup>
  - The pilot commands angular positions through conventional [RC](#) equipment, input is saturated
- Autonomous<sup>3</sup>
  - The pilot has no input to the system, a predefined flight plan is executed

The [SAS](#) operates at 50Hz, far quicker than the human operator, allowing for rapid yet smooth disturbance rejection when properly tuned. This provides the ability to conduct flight testing in weather conditions which would otherwise be unsuitable for small [RC](#) vehicles, such as in strong gusty wind.

---

<sup>1</sup>Termed *Stabilize* by [APM](#)

<sup>2</sup>Termed *Fly By Wire (FBW)* by [APM](#)

<sup>3</sup>Termed *Auto* by [APM](#)

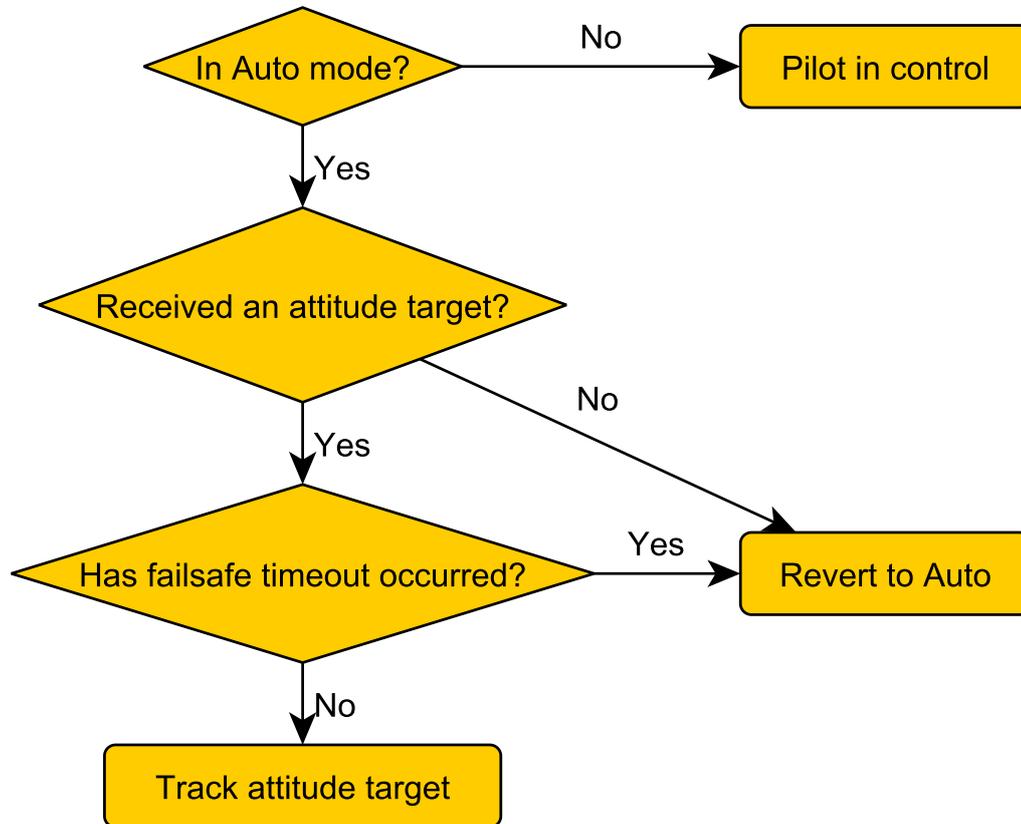


Figure 6.5: [APM](#) auto mode execution logic

The manual, [SAS](#) and [CFH](#) modes can all be used at the discretion of the pilot when they are in control of the vehicle. For the test environment it is necessary to allow an external system (e.g. MATLAB/Simulink) to provide control signals to [APM](#). This was achieved by modifying the autonomous mode to function similarly to [CFH](#) but with the commands provided from an external system. This modification includes a failsafe feature to revert to autonomous mode if no external command is received, this ensures the vehicle returns to its home location. The failsafe logic is shown in Figure 6.5.

---

### 6.3.3 Prototyping features

To facilitate rapid prototyping of new features [APM](#) supports both [SIL](#) and [HIL](#) development. In both of these capacities a simulated vehicle is used to provide [APM](#) with state feedback in reaction to its control outputs. [SIL](#) mode allows the [APM](#) autopilot software to be compiled on a desktop computer, this enables fundamental bugs in the code to be eliminated prior to involving any actual hardware. [HIL](#) compiles the software on to the actual [APM](#) hardware to test for any issues which may occur when interacting with the processor, the [RC](#) equipment or the XBee data link.

In addition to prototyping changes to the autopilot software, the [SIL](#) mode is particularly useful for prototyping advanced [UAS](#) functionality. This is because many tests can be conducted in quick succession without risk to any actual hardware.

## 6.4 Micro Air Vehicle Communication Protocol ([MAVLink](#))

[MAVLink](#) is a lightweight, efficient message marshalling protocol for data transmission developed by the PIXHAWK project which is employed by the [APM](#) system [[Meier et al., 2011](#)]. To ensure that all elements of the test environment communicate in a consistent manner the [MAVLink](#) protocol was used for all custom components.

[MAVLink](#) is written as a header only C library which is easily used by C or C++ software. All data is transmitted using binary encoding which means individual bytes are combined in to packets that can be easily interpreted by a computer but are not human readable. Binary encoding is significantly more efficient than the alternative, human readable, American Standard Code for Information Interchange ([ASCII](#)) encoding. This is illustrated by considering the transmission  $\pi$  to 14 decimal places. [ASCII](#) transmission is human readable, therefore  $\pi$  would be transmitted as 3.14159265358979. As each character corresponds to a single byte in [ASCII](#), a total of 16 bytes must be transmitted, shown in [Table 6.1](#).

Table 6.1: [ASCII](#) representation of  $\pi$  (bytes shown in hexadecimal)

3	.	1	4	1	5	9	2	6	5	3	5	8	9	7	9
33	2F	31	34	31	35	39	32	36	35	33	35	38	39	37	39

Table 6.2: [IEEE](#) float (binary) representation of  $\pi$  (bytes shown in hexadecimal)

11 2D 44 54 FB 21 09 40

Binary encoding utilises the fact that decimal numbers (known as floating point data, or simply floats) are stored in computers using an Institute of Electrical and Electronics Engineers ([IEEE](#)) standard requiring only 8 bytes [[IEEE, 2008](#)]. Unlike [ASCII](#) where each byte has a human readable equivalent character, binary encoding only provides byte values which must be recombined to form an [IEEE](#) float, shown in Table 6.2.

Floating point values and other data types can be placed in to a single packet to be transmitted [[MAVLink, 2012](#)]. [MAVLink](#) defines a number of these packets such as

- Attitude information ( $\phi, \theta, \psi, P, Q, R$ )
- [GPS](#) information (latitude, longitude, altitude, ground speed, track)
- System status (autopilot mode, battery voltage, battery power remaining)

Each packet has a header containing information such as which vehicle the data has come from and how much data is being transmitted. This is shown in Figure 6.6, and a full description given in Table 6.3.

## 6.5 High fidelity flight model - X-Plane

To provide a high fidelity flight model in the test environment, the X-Plane flight simulator by Laminar Research was used [[Laminar Research, 2012b](#)]. X-



Figure 6.6: [MAVLink](#) packet structure

Table 6.3: MAVLink packet description

Byte	Content	Value	Details
0	Start	0x55	Indicates start of a new packet
1	Length	0-255	How much data is in the message (the value of $n$ )
2	Sequence	0-255	Increments with each packet sent, allows determination of dropped packets
3	System	0-255	Identity of the system sending the packet
4	Component	0-255	Identity of the component sending the packet
5	Message	0-255	Identity of the message being sent
$\left\{ \begin{array}{c} 6 \\ \vdots \\ n+6 \end{array} \right\}$	Data		Message data
$\left\{ \begin{array}{c} n+7 \\ n+8 \end{array} \right\}$	Checksum		ITU X25 hash of bytes 1 to $n+6$ (i.e. excluding start sign) [International Telecommunications Union, 1996]

---

Plane is a [COTS](#) flight simulation package which is certified for Federal Aviation Administration ([FAA](#)) flight training and uses blade element theory to power its flight model. This ensures high accuracy, high fidelity, simulation of a large array of vehicles across a large range of operating conditions [[Craighead et al., 2007](#)]. Included in the X-Plane package is an airfoil and aircraft modelling program which enables development of custom vehicles to either match real vehicles or test concept designs.

To integrate X-Plane within the test environment a custom plug-in was required to expose the required data to the [APM](#) system and accept control signals using the [MAVLink](#) protocol as shown in [Figure 6.2](#). This section details the development of the plug-in.

### 6.5.1 Software Development Kit (SDK)

X-Plane provides a fully functional [SDK](#) allowing custom plug-ins to be created for a number of purposes. The [SDK](#) interfaces with a number of programming languages, C++ was used to provide maximum flexibility. Each plug-in must be compiled as a Dynamically Linked Library ([DLL](#)) which X-Plane then links with at runtime to provide the additional functionality. Interaction between X-Plane and a plug-in is handled in two ways, callbacks and *datarefs*.

#### Callbacks

A callback is a special function defined in the plug-in which X-Plane will call at certain times. Certain callbacks are required for a plug-in to function correctly, these are

- `XPluginStart` - Called when X-Plane starts the plug-in
- `XPluginStop` - Called when X-Plane stops the plug-in
- `XPluginReceiveMessage` - Called when X-Plane (or another plug-in) sends a message to the plug-in

The majority of callbacks are optional and their use depends on the purpose of the plug-in. The callback which contains the majority of the code for the test environment is the *Flight Loop Callback* which is called each time X-Plane calculates

---

its flight model. This high frequency callback allows a plug-in to communicate with external software performing autopilot functions.

### Data references (*datarefs*)

Data references (known as *datarefs* in the [SDK](#)) are memory locations where simulation data can be accessed by the plug-in [[Laminar Research, 2012a](#)]. A vast amount of data is accessible through *datarefs*, the most useful of which are.

- Control signals for elevator, aileron and rudder. Scaled from -1 to 1.

```
sim/joystick/yoke_pitch_ratio
sim/joystick/yoke_roll_ratio
sim/joystick/yoke_heading_ratio
```

- Throttle signal. Scaled from 0 to 1.

```
sim/multiplayer/controls/engine_throttle_request
```

- Latitude, longitude and altitude

```
sim/flightmodel/position/latitude
sim/flightmodel/position/longitude
sim/flightmodel/position/altitude
```

- Horizontal and vertical airspeed

```
sim/flightmodel/position/indicated_airspeed
sim/flightmodel/position/vh_ind
```

- [GPS](#) speed and track

```
sim/flightmodel/position/groundspeed
sim/flightmodel/position/hpath
```

- Attitude.  $\phi$ ,  $\theta$  and  $\psi$

```
sim/flightmodel/position/phi
sim/flightmodel/position/theta
sim/flightmodel/position/magpsi
```

- 
- Rotational rates.  $P$ ,  $Q$  and  $R$

sim/flightmodel/position/P

sim/flightmodel/position/Q

sim/flightmodel/position/R

- Normalised Accelerations.  $a_x/g$ ,  $a_y/g$  and  $a_z/g$

sim/flightmodel/forces/g\_axil

sim/flightmodel/forces/g\_side

sim/flightmodel/forces/g\_nrml

It is clear that the naming convention for *datarefs* is not always clear or consistent, neither are the units or sign conventions. The [MAVLink](#) plug-in performs conversion of units to International System of Units ([SI](#)).

## 6.5.2 Multiplayer vehicles

X-Plane natively supports up to 20 vehicles, these are conventionally used for multiplayer gaming (such as formation flying or combat) or as [AI](#) traffic aircraft. The [MAVLink](#) plug-in is used to take control of any number of these vehicles in a similar way to the primary vehicle. This enables the test environment to simulate a number of traffic vehicles, all of which can be visualised within the same X-Plane window simultaneously.

## 6.5.3 Communication

The plug-in utilises the Microsoft Windows Sockets Application Program Interface ([API](#)) to provide a Transmission Control Protocol ([TCP](#)) network connection over which [MAVLink](#) packets can be transmitted [[Microsoft, 2011](#)]. In addition to a [TCP](#) connection, a serial port interface is also provided to allow [APM](#) hardware to connect directly to X-Plane in [HIL](#) mode.

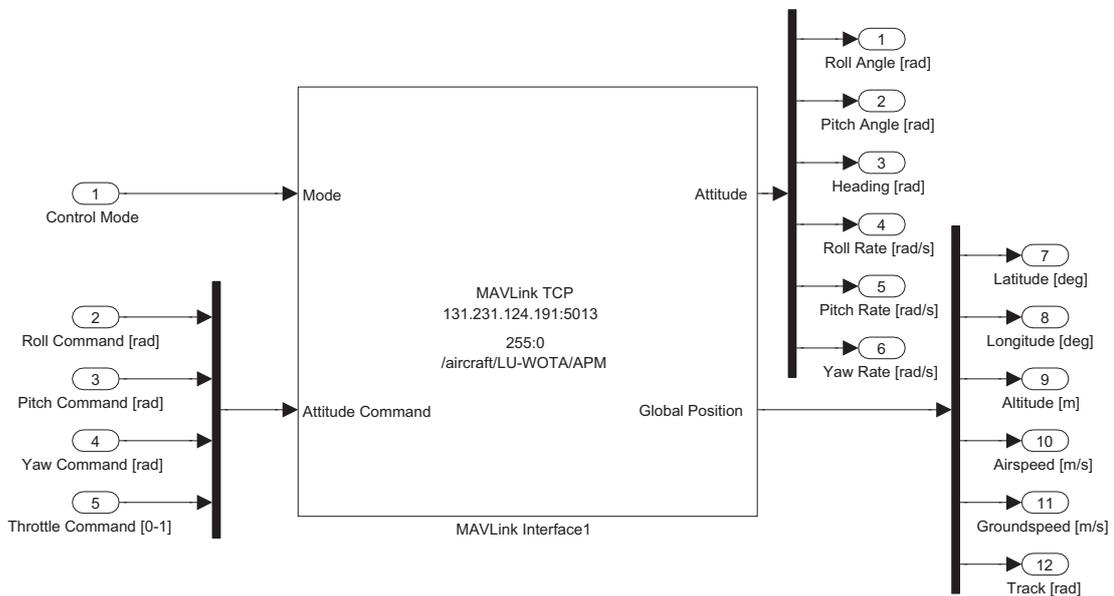


Figure 6.7: Custom Simulink block for MAVLink communication

## 6.6 MATLAB/Simulink interface

Similar to the X-Plane plug in, a custom Simulink block was created to communicate directly with [APM](#) using the [MAVLink](#) protocol. Figure 6.7 shows a typical implementation of this block.

The centre of the block in Figure 6.7 contains information regarding the connection, in this case a [TCP](#) connection to a computer with Internet Protocol ([IP](#)) address 131.231.124.191 on port 5013. A number of different input and output signals are selectable, the most commonly used are shown in Figure 6.7. The connection information along with the selection of inputs and outputs are configurable via a pop up dialogue window.

Multiple instances of the Simulink block can be used to command multiple vehicles in [SIL](#) or [HIL](#) modes, in addition to real vehicles.

## 6.7 Real world environment

To perform small scale, real world testing an airframe was required which met some basic requirements stemming from risk and operational considerations.



Figure 6.8: Ripmax Wot4 Foam-E with [APM](#) installed

- [COTS](#)
- Low cost
- Light weight
- Electric powered
- Robust
- Internal space for [APM](#)

Figure 6.8 shows the Ripmax Wot4 Foam-E<sup>1</sup> with [APM](#) installed internally. This airframe is constructed from Expanded Polyolefin ([EPO](#)) foam which is lightweight, strong and robust. The wingspan is 1.2m and the operating weight is 1.25kg with [APM](#) installed and a 2600mAh Lithium Polymer ([LiPo](#)) battery which provides around 20 minutes endurance.

---

<sup>1</sup><http://www2.ripmx.net/Item.aspx?ItemID=A-CF020&Category=010>



Figure 6.9: X-Plane model of the Wot4 airframe

In order to keep the aerial platform consistent throughout [SIL](#), [HIL](#) and real world tests a model was created using the X-Plane aircraft design program (the *Plane Maker*), Figure 6.9.

The architecture of the test environment for real world testing is shown in Figure 6.10. When compared with Figure 6.2 it is clear that the Simulink interface requires no modification from that used in [SIL](#) and [HIL](#).

### 6.7.1 Multi vehicle and mixed environment testing

Multi vehicle real world tests may be performed by simply using multiple real world aircraft. However, additional complexities arise when operating multiple fixed wing aircraft in close proximity to one another (such as the requirement for a number of skilled safety pilots), for this reason it is desirable to allow mixed use of [SIL](#), [HIL](#) and real world environments. In all environments, the Simulink implementation is identical and can be configured to communicate with a [SIL](#) autopilot build, an actual [APM](#) in [HIL](#) mode or a real world vehicle. Multiple instances of the Simulink block, operating in any of the three modes, can be utilised in a Simulink model without restriction. It should be noted, however, that the computational burden of running X-Plane, the [SIL](#) build and Simulink on a single computer is significant and may produce poor simulation results (due

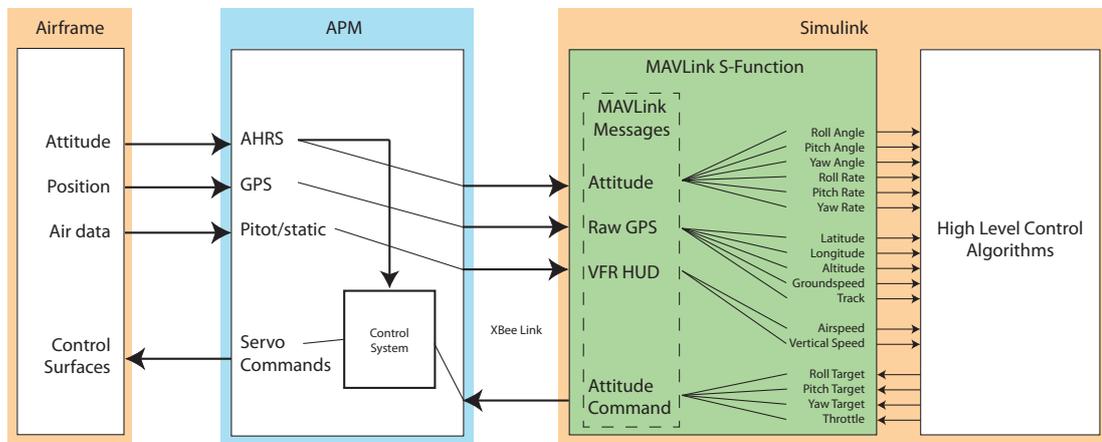


Figure 6.10: Real world interface between Simulink and [APM](#)

to a lack of real time execution). It is recommended that mixed environment testing be performed with multiple computers to distribute this load and improve results.

## 6.8 Results

To demonstrate the utility of the test environment a number of different experiments have been conducted. These experiments were used to determine both the accuracy of the X-Plane Wot4 model and the differences which occur between [SIL](#), [HIL](#) and real world testing [[Coombes et al., 2012](#)].

### 6.8.1 Speed hold test

Figure [6.11](#) illustrates the performance of a simple speed hold controller implemented in Simulink. It can be seen that [SIL](#) and [HIL](#) results match very closely with only a slight difference in the time period of the dynamics. Both these tests use the X-Plane dynamic model, therefore very similar performance is to be expected. [SIL](#) executes the autopilot on a desktop computer which functions significantly faster than the Arduino processor used for [HIL](#). This difference in processing is likely to be the cause of the minor difference in transient behaviour. The real world results can be seen to broadly match those of both the [SIL](#) and [HIL](#), however increased damping is observed. An exact match between simulation

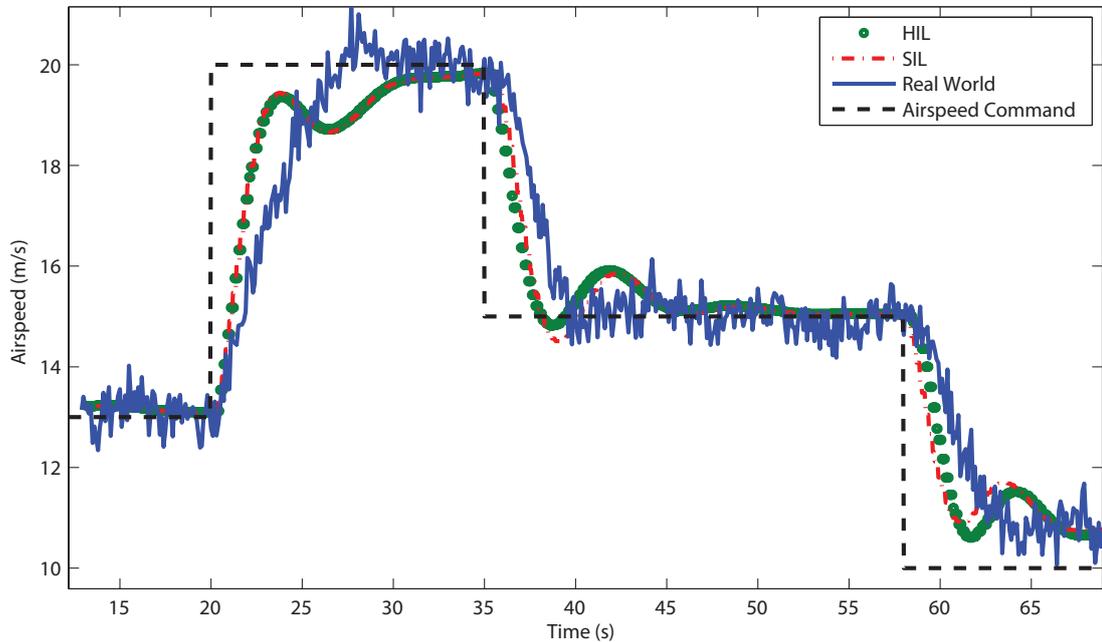


Figure 6.11: Speed hold test for [SIL](#), [HIL](#) and real world

and real world is not expected due to the vastly increased complexity of the real system, therefore this performance is adequate.

## 6.8.2 Heading hold test

Figure 6.12 illustrates the performance of a Simulink heading hold controller. Once again [SIL](#) and [HIL](#) match very closely with real world results exhibiting a more damped response. It can be seen from the 30–40s period that the frequency and amplitude of the short period dynamics are also closely matched<sup>1</sup>.

At approximately 4s and again at 20s a flat spot in the real world data is present. This is caused by a drop out in communication between the vehicle and the ground. Drop outs like these were observed frequently, particularly during manoeuvres such as turns. This is due to the change in orientation of the XBee antenna during the manoeuvre causing a mismatch in polarisation with the ground antenna, coupled with the relatively low power of the XBee devices. Use of higher power or multiple antennae radios is a potential means of mitigating

<sup>1</sup>Short period dynamics are excited in X-Plane by setting the wind conditions to be representative of the real world tests

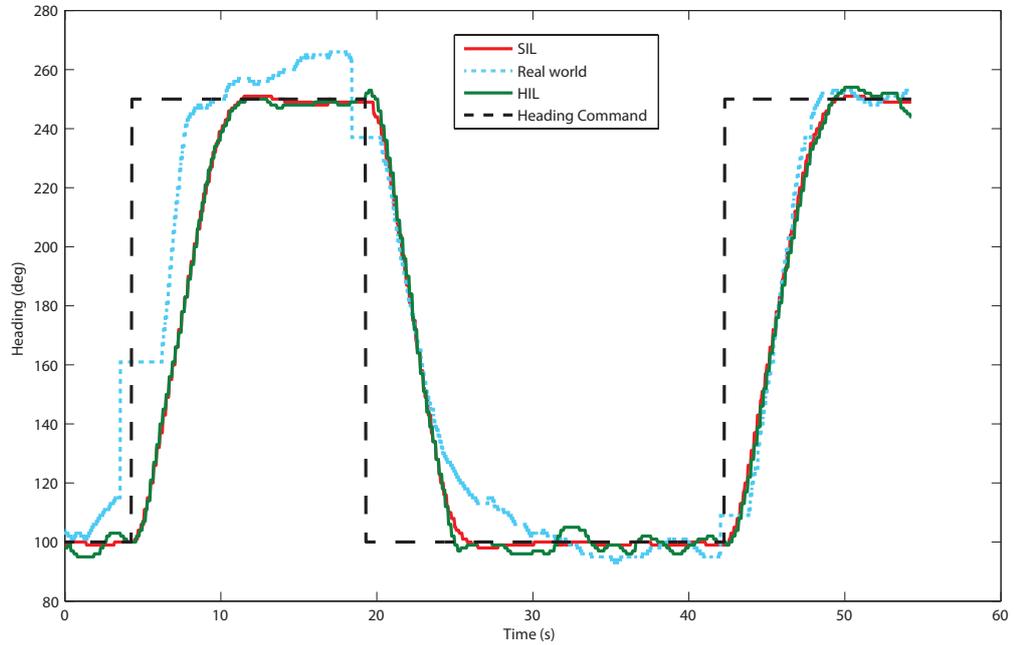


Figure 6.12: Heading hold test for [SIL](#), [HIL](#) and real world

this problem. Alternatively, if the range to the vehicle is kept low, drop outs can be minimised.

### 6.8.3 Mixed mode multiple vehicle test

Due to the complexities of real world multiple vehicle testing, the concept of mixed mode operation was examined. A formation flying experiment was conducted with a simulated [SIL](#) aircraft tasked with following the trajectory of a real vehicle, seen in Figure [6.13](#).

This test was broadly successful, however the importance of communication integrity was highlighted. As with the previous test, drop outs in communication with the real vehicle occur which cause the following vehicle to lose its reference trajectory. When this occurs the following vehicle simply continues on a linear path which may result in a loss of separation if the leader is in fact manoeuvring. This issue can be resolved by improving the communications integrity as discussed in the previous section.



Figure 6.13: Mixed mode (SIL and real) formation flight

---

## 6.9 Conclusion

This chapter has detailed the development of a test environment for autonomous UASs utilising COTS equipment. The APM autopilot has been shown to be an easy to use system which allows for SIL, HIL and real world testing to be conducted with minimal change of infrastructure, including multi vehicle tests. The X-Plane flight simulation package was used as a high fidelity model due to its SDK allowing ease of development.

Custom software has been developed to allow X-Plane to provide aircraft dynamics information to the APM system for SIL and HIL testing. Additionally, a custom MATLAB/Simulink block has been created to receive state information from and send commands to the APM system. These components provide complete integration of the X-Plane and APM system across SIL, HIL and real world testing.

A number of tests have been performed to demonstrate the utility and performance of the system. Very little difference between SIL and HIL test results has been observed. This allows initial development of algorithms to focus on the use of the SIL technique without loss of fidelity. Real world testing has been completed without any software changes from SIL, demonstrating a high level of abstraction has been achieved.

Despite a number of successful flight tests, the problems associated with reliable data communication have been highlighted. Future development of the system should focus on replacing the XBee system with higher integrity radios utilising multiple antennae in addition to higher transmission powers.

# Chapter 7

## Experimental Results

### 7.1 Introduction

This thesis has discussed the development of an artificial situation awareness system for [UASs](#) capable of providing sufficient information to perform autonomous navigation through a crowded terminal area. Each stage of the system has been tested independently using simplified numerical models for the vehicles concerned. This chapter expands on this by conducting experimental assessment in a high fidelity [SIL](#) environment detailed in the previous chapter. [HIL](#) tests were not performed as these are most useful when fundamental control structures are changed, not for high level assessment. Real world tests were not conducted at this stage due to the lack of high integrity communication making multiple vehicle testing unreliable. Despite this, all [SIL](#) experiments were conducted so as to be consistent with real world tests. This allows real world testing to be conducted in the future if the data integrity problems are overcome.

The next section introduces the experimental set up including the software and hardware implementations. Section [7.3](#) discusses the test scenarios used. Section [7.4](#) discuss the [SIL](#) test results.

---

Table 7.1: Laptop Specifications

Processor	Intel Core i5-2430M
Speed	2.4GHz
FLOPS <sup>3</sup>	61 GFLOPS
Memory Bandwidth	12GB/s

## 7.2 Experimental set up

The previous chapter discussed the test environment in detail, this section covers the specifics of the set up used for the experimental tests. To enable real world tests in the future, a laptop was used to run the [SIL](#) tests, the specifications of the which are given in [Table 7.1](#).

The X-Plane and MATLAB/Simulink environment run in the native operating system (Windows 7), however the [APM SIL](#) build must be executed from Linux. To prevent two separate computers being required, the freely available Virtual-Box<sup>1</sup> virtualisation package was used to run an installation of Ubuntu<sup>2</sup> Linux from inside Windows, [Figure 7.1](#). However, an incompatibility between Windows 7 and VirtualBox prohibits direct network communication through [TCP](#) sockets as is required by the [SIL](#) environment (as discussed in the previous chapter). To mitigate this problem the laptop must be connected to a network router and this connection bridged to VirtualBox, this router is then responsible for handling the communication between Windows (the *host* system) and Linux (the *guest* system), [Figure 7.2](#).

## 7.3 Test scenarios

In order to provide a test scenario consistent with future real world tests, the constraints which occur during real world testing were also applied to the [SIL](#) study. These constraints are imposed by both the aircraft platform (the Ripmax Wot4 Foam-E, discussed in the previous chapter) and the operating area used for flight tests. The area used for flight testing is a small field located at the South West

---

<sup>1</sup><https://www.virtualbox.org/>

<sup>2</sup><http://www.ubuntu.com/>

<sup>3</sup>Floating Point Operations Per Second

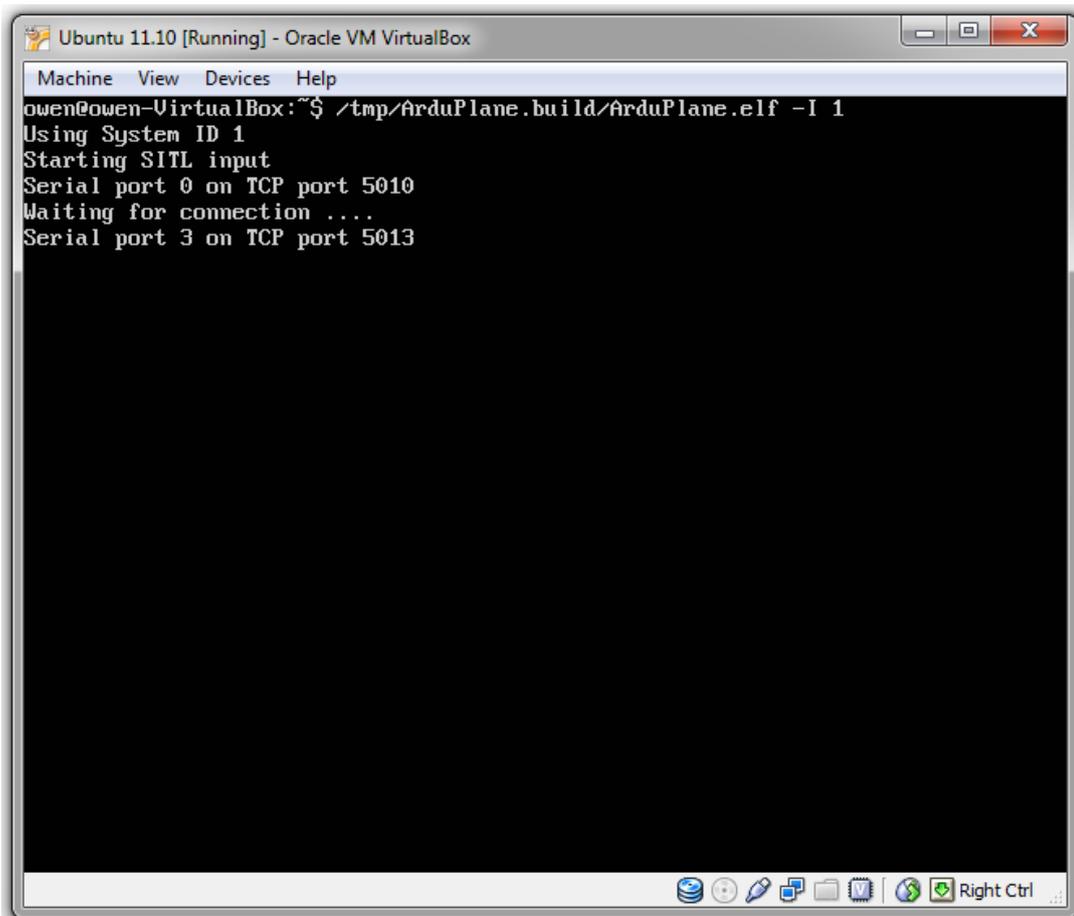


Figure 7.1: [APM SIL](#) build executing in a VirtualBox virtual machine

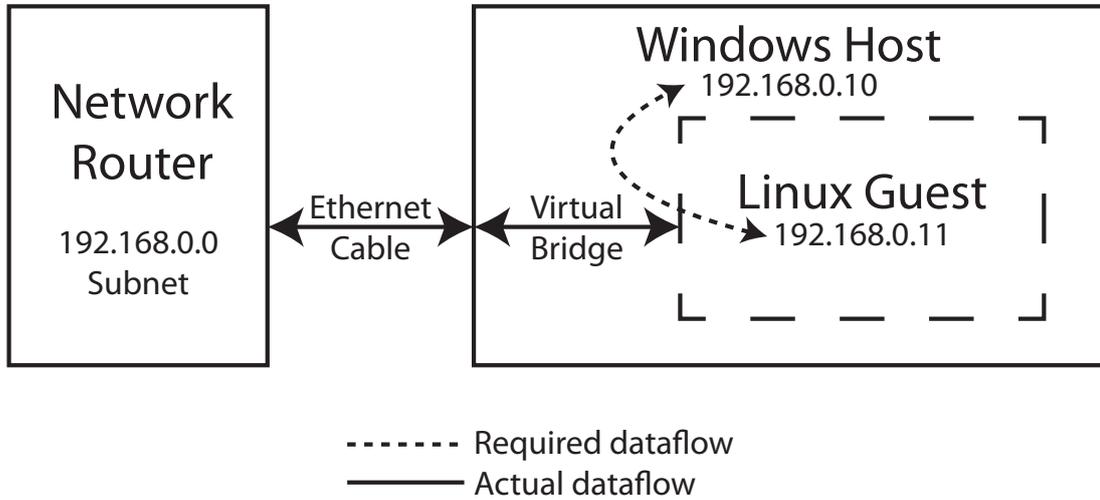


Figure 7.2: Network router to provide TCP link for VirtualBox

Table 7.2: Traffic navigation errors, standard deviations

Heading	$2^\circ$
Speed	$1ms^{-1}$
Discrete transition	$20m$

end of Loughborough University, centred approximately at  $52.75603^\circ\text{N}$   $1.2465^\circ\text{W}$ .

Figure 7.3 illustrates the test scenario used. The areas shaded in blue are beyond visual range of the pilot in real world tests, where the location of the the pilot is indicated by the yellow star. This maximum range has been determined experimentally to be approximately 225m based on the vehicle being used, flying beyond this range would not permit the pilot to safely control the vehicle manually should the autopilot fail. The red shaded areas are too close to buildings and/or persons to allow flights to be conducted safely. The path marked in white represents the nominal circuit dimensions to be followed by traffic aircraft. The upwind axis of the circuit (marked  $U$ ) is aligned with magnetic heading of  $200^\circ$  which is approximately equal to the prevailing wind direction. The crosswind axis (marked  $C$ ) is in the direction of the circuit. The blue path represents a holding point for the UAS and its subsequent path in to the circuit.

The traffic aircraft is commanded to follow the nominal circuit path with random navigation errors imposed, given in Table 7.2.

The UAS is commanded to remain in the hold until the traffic aircraft enters

---

the downwind leg, indicated by the position of the white aircraft in Figure 7.3. The UAS will be at a random position in the hold and be commanded to take the shortest route to the beginning of the base leg. This results in a number of different possible trajectories, some of which are illustrated in Figure 7.4. These trajectories provide the opportunity for the UAS to enter the circuit either in front of or behind the traffic vehicle.

To provide the UAS with freedom to manoeuvre in order to self separate from the traffic its speed may be varied in the range of  $10 - 15\text{ms}^{-1}$  (where  $12\text{ms}^{-1}$  is nominal) and the position of the base leg may be modified as indicated by the arrows in Figure 7.3. These variations will be governed by the decision making rules laid out in Section 5.4.2.

Only a single traffic vehicle is to be investigated as the decision making rules detailed in Section 5.4.2 make clear that the UAS is only concerned with vehicles immediately ahead or behind. The condition where an aircraft is both ahead and behind is both complex to set up and presently intractable due to the computational burden of the situation awareness determination.

It should be noted that the UAS is not given the ability to orbit if required for self separation (as per Figure 5.16) due to the space constraints present during real world testing.

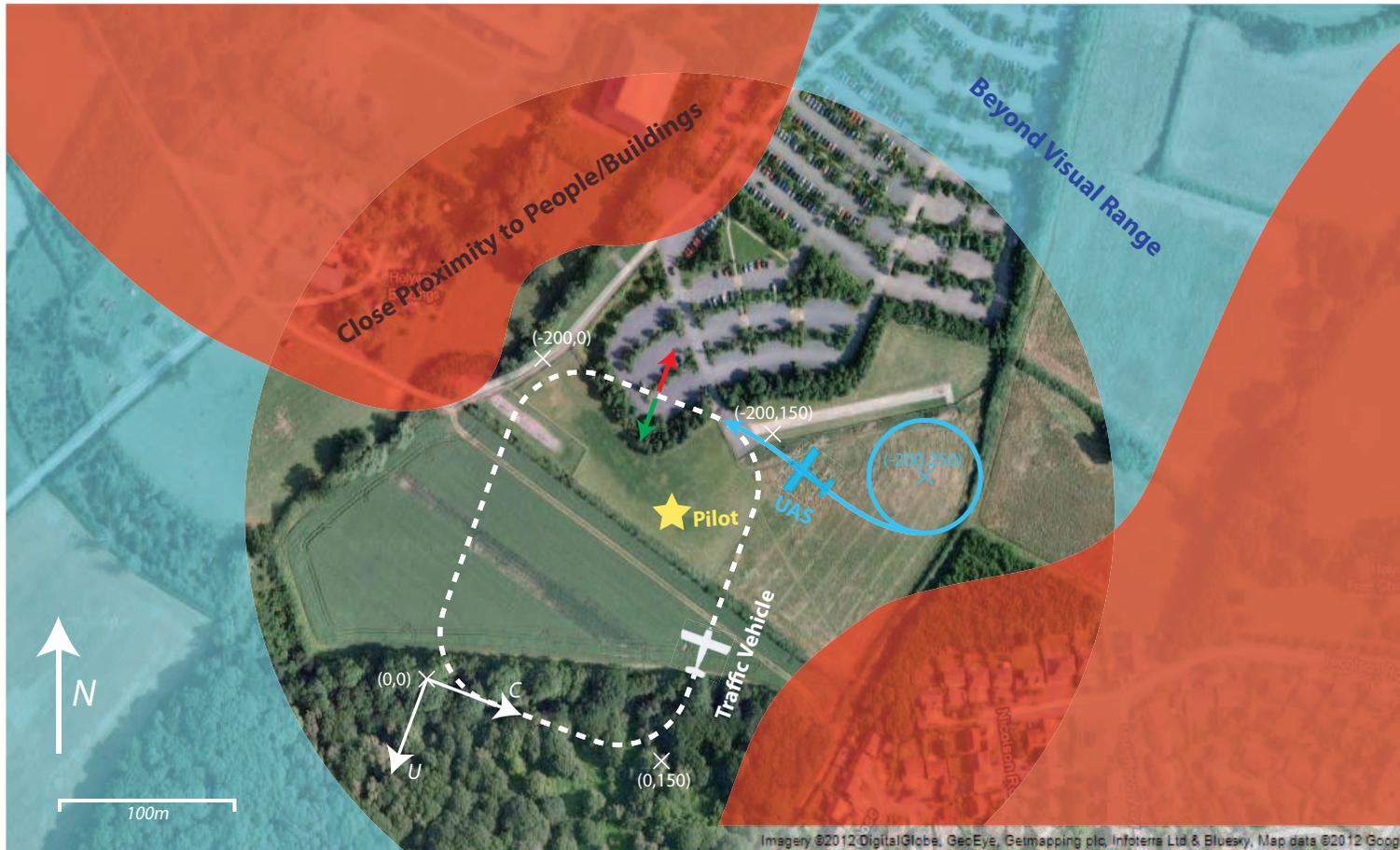


Figure 7.3: Diagram of the SIL and real world test scenario



Figure 7.4: Alternate routes for UAS from hold to circuit

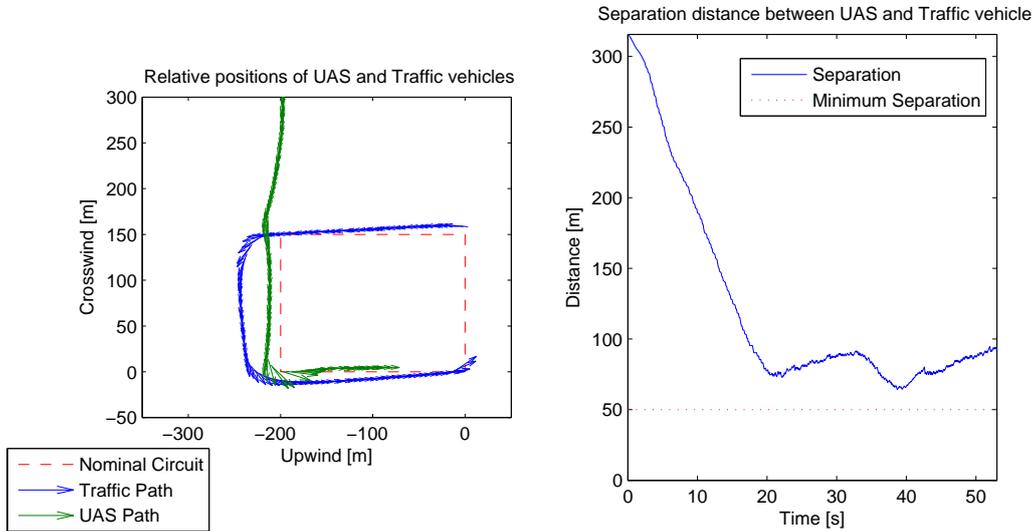


Figure 7.5: SIL results - Test 1

## 7.4 SIL Results

This section presents results from the SIL study based on the test scenario described in the previous section.

### 7.4.1 Typical trajectories

Figures 7.5 and 7.6 illustrate typical results for the case where the UAS determines it should follow behind the traffic. Figure 7.5 shows the traffic then delaying its turn on to base by around 30m, whereas Figure 7.6 shows the traffic turning early. In both cases the separation between vehicles remains satisfactory. Note that 50m has been chosen as satisfactory separation which is approximately 50 times the wingspan of the vehicles.

Figure 7.7 illustrates a typical result when the UAS enters the circuit ahead of the traffic. This case is of less interest because once positioned ahead of the traffic the UAS may then elect to follow the nominal path and expect the traffic to give way. The giving way of traffic was not modelled in this study (traffic vehicles are simply ‘dumb’), therefore infringement of minimum separation after a UAS has correctly entered the circuit ahead of traffic is not considered a problem. It can be seen from Figure 7.7 that the UAS elects to enter the circuit ahead of the

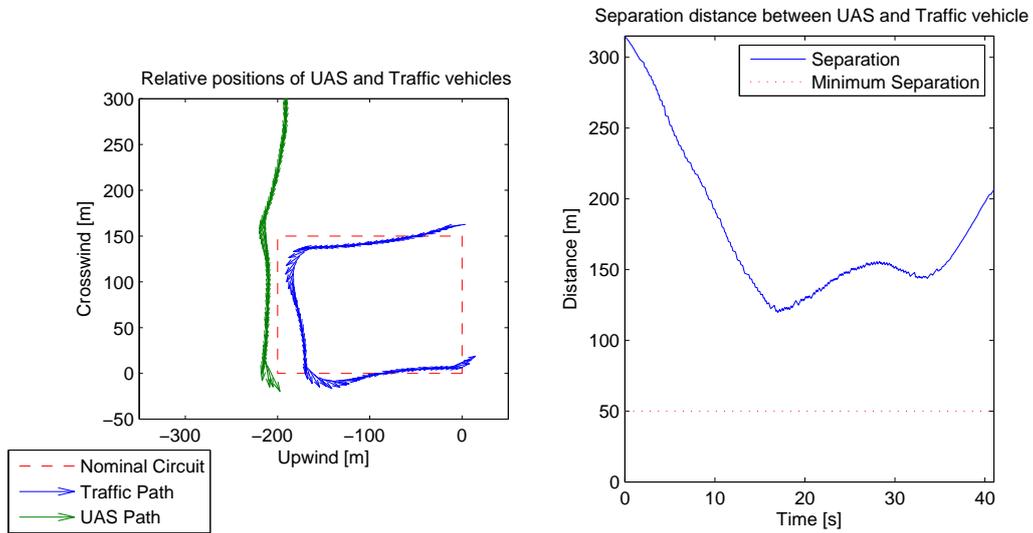


Figure 7.6: SIL results - Test 2

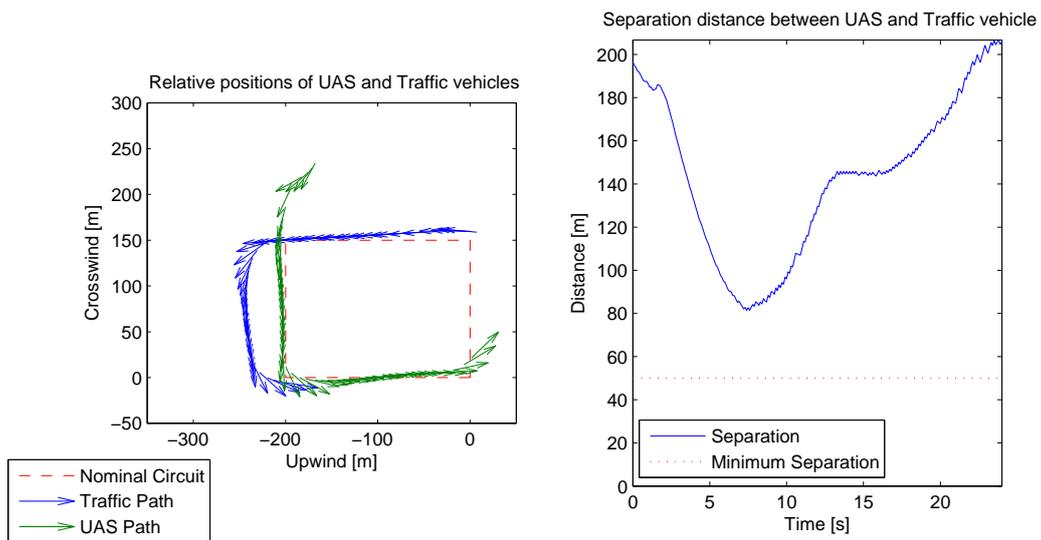


Figure 7.7: SIL results - Test 3

---

traffic only when it exists the hold at a point near the circuit, giving it a short path to base.

## 7.4.2 General trends

In addition to the specific examples illustrated in the previous section, this section highlights some general trends in the performance of the system. Figure 7.8 shows two stacked histograms illustrating the minimum separation distance between the UAS and the traffic vehicle over the course of one hundred simulated circuits. It is clear that the minimum separation criteria of  $50m$  was not met in all cases, primarily when the UAS enters ahead of the traffic. It was noted in the previous section that the traffic aircraft are ‘dumb’ and will not separate from the UAS as is required in reality. The loss of separation when the UAS is ahead of the traffic is caused when the randomly generated traffic trajectory causes a conflict long after the UAS has correctly entered the circuit. These data points can be considered as anomalous as they would not be present in a fully developed system deployed against manned traffic aircraft (which would separate correctly).

The loss of separation in the case where the UAS is behind the traffic aircraft is of more interest as this is the responsibility of the UAS. It can be seen from Figure 7.3 that the limited space available presents the UAS with a finite amount of freedom to manoeuvre. As an orbit manoeuvre may not be performed (again due to space constraints) there are situations where the traffic vehicle is flying sufficiently fast that the UAS can not slow sufficiently to maintain adequate separation. In a complete system these situations would be considered as the ‘Emergency’ case in Figure 5.18 and the decision making authority referred back to the operator (or an alternative emergency collision avoidance system).

In the cases where adequate separation is maintained it can be seen that the majority remain within  $100m$  separation. This behaviour is desirable as within the terminal region it is beneficial to maintain as minimal a separation as is safe so as to maximise the airfield utilisation.

The high separation cases are caused when the traffic vehicle opts to fly a short, fast circuit and the UAS must fly almost entirely round its hold prior to entering the circuit (see Figure 7.4). In such a scenario it is only desirable for the

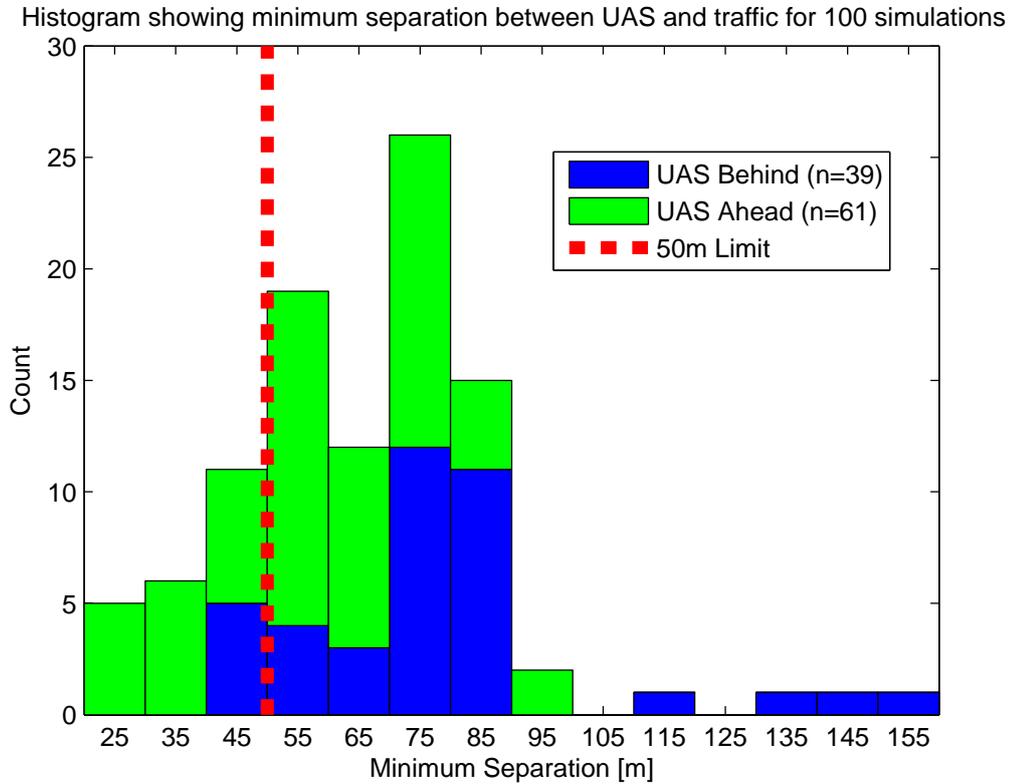


Figure 7.8: Minimum separation distance between UAS and traffic over 100 simulations

UAS to increase speed if further traffic is present which was not the case during this test.

## 7.5 Conclusion

This chapter has detailed the experimental set up and results of a SIL test of artificial situation awareness for a fully autonomous UAS. HIL tests have not been performed as these have been shown to match SIL tests very closely in the previous chapter. Real world tests have not been performed due to the lack of high integrity communications. At all times during the SIL set up, however, consideration has been given to constraints imposed by real world tests. It is believed that once the communications integrity has been improved, real world testing can be

---

conducted with no modification to the MATLAB/Simulink implementation of artificial situation awareness.

SIL results have shown the general trends of the system are very promising, with the UAS acting as expected in all situations. The reaction of the UAS to traffic vehicles follows the procedures that a human pilot would be expecting, demonstrating a high level of transparency and equivalence.

Two situations were encountered where safe separation between the UAS and traffic was lost, both of which were limitations of the test scenario and not the system itself. A fully deployed system would not routinely experience ‘dumb’ traffic vehicles which refuse to give way correctly. When this does occur it is down to the emergency systems on board the UAS to avoid a collision.

# Chapter 8

## Conclusions

### 8.1 Summary

This thesis has approached a key issue regarding the integration of **UAS** in to the **NAS**, that is maintaining equivalent levels of safety to manned aircraft. It has been shown that to safely separate from manned vehicles it is necessary for an **UAS** to possess a high degree of autonomy. This thesis has considered the extreme case of a fully autonomous **UAS**, as may be required to mitigate communications loss. The ability of an autonomous **UAS** to make safe decisions about its future path has been shown to require high levels of situation awareness, that is an understanding of where traffic vehicles are and how they will behave in the future. Determination of the current state of traffic is aided by emerging technologies such as **ADS-B**, however future trajectories are not always available from flight plans or **ATC** instruction. To make meaningful predictions about the future states of traffic, in the absence of future trajectory information, this thesis has incorporated additional sources of knowledge such as the **ROTA** and heuristics governing traffic operation. The most critical phase of flight for high autonomy and therefore high situation awareness has been identified as the terminal area, which has been the focus of this thesis.

The situation awareness problem has been broken down in to three distinct areas, perception, comprehension and projection. It has been assumed a perception task already exists which can provide position data of traffic vehicles at a

---

rate of around 1Hz. A comprehension system has been developed based upon an existing hybrid estimation technique capable of estimating both continuous and discrete state information. This novel application of an existing technique allows an autonomous **UAS** to estimate the location of a traffic vehicle in the predefined traffic circuit at an uncontrolled airfield. Knowledge of this information allows the future intent of the traffic to be inferred based upon the **ROTA**, this inference forms the basis of the projection element.

The comprehension of current state and inference of future trajectories has allowed a projection system to be designed which is able to estimate future traffic positions to projection times of the order of minutes. It has been shown that projection of position is subject to a number of uncertainties which must be quantified to provide a meaningful estimate. Uncertainties in traffic state have been captured by treating the vehicles position as a **PDF** whose covariance terms capture uncertainty in future position. Uncertainty in the intent information, such as when a vehicle is expected to manoeuvre, has been captured with a novel geometric approach. Any discrete transition (such as the start of a manoeuvre) is treated as a probabilistic event and the effect of the variance term on the uncertainty in future trajectories of traffic is derived geometrically. This additional uncertainty introduced at each discrete transition is summed together along with the continuous uncertainty to provide a complete quantification of the accuracy of the projected position.

To enable the **UAS** to make decisions based upon the projected positions of traffic, metrics for separation have been derived in terms of the projection **PDFs**. It has been shown that even for simple Gaussian cases (with no discrete uncertainty), calculation of the distance distribution from a known **UAS** to a traffic position **PDF** is non-trivial. For both Gaussian and non-Gaussian cases the metrics of distance, time separation, **TPCA** and **DPCA** have been derived by utilising a number of assumptions and approximations where necessary to maintain computational feasibility. These metrics have then formed the basis of a rule based decision making system to allow the **UAS** to safely separate from traffic vehicles autonomously.

To allow high fidelity testing to be conducted, a test environment was created based around a number of **COTS** products. The test environment allows the use

---

of MATLAB/Simulink to rapidly prototype advanced autonomous functions for an UAS. The environment enables the use of a high fidelity flight simulator to represent vehicle dynamics and the use of small scale vehicles to test the system in the real world. This environment has been used extensively throughout to quantify the performance of the artificial situation awareness and self separation systems, however only simulation studies have been performed to date. Other projects have made use of the test environment for real world studies and found the problems associated with low data transmission integrity to be prohibitive. All testing conducted during this project, however, may be directly transferred to real world test when this integrity issue is solved.

## 8.2 Research challenges

As an outcome from this project, a number of future research challenges have been highlighted. This section introduces these challenges along with possible techniques to be investigated.

### 8.2.1 Multiple model projection

The hybrid estimation technique used during the comprehension phase utilises multiple vehicle models dependant on the current discrete mode. This approach allows for improved performance by rejecting disturbances not associated with the mode. Further work should be done to extend this multiple model technique to the projection system presented in this thesis. A multiple model projection algorithm would allow for more intuitive quantification of uncertainties (such as turn rate uncertainty during manoeuvres) and should benefit from similar performance improvements at the comprehension system.

Using multiple projection models introduces challenges associated with the integration of uncertainty over time. For example, (4.9) requires the covariance matrix for each mode to be of the same form. A non-linear turning model which incorporates turn rate uncertainty, however, would not be compatible with a linear model of unaccelerated flight.

---

## 8.2.2 Constrained projection

The projection method presented in this thesis includes a small amount of work in the area of bounded uncertainty, enabling the covariance matrix to be restricted as the runway is approached. In more complex scenarios bounds may be imposed on the future trajectory of the vehicle from a number of other sources. These include guidance systems on-board the vehicle and restricted no-fly zones.

Whilst this thesis assumed that velocity errors may accumulate indefinitely until the runway is encountered, a real vehicle would likely attempt to minimise these errors through either manual or automated guidance. The scenarios considered in this thesis have been over relatively short distances making this effect small, however, more fully developed scenarios must account for this upper bound on projection uncertainty.

In addition, it is possible for complex terminal regions to include no-fly zones imposed by terrain, weather, or restricted airspace. Knowledge of these areas allow the **UAS** to further constrain the future trajectories of traffic vehicles by assuming they will not enter no-fly zones.

## 8.2.3 Computational complexity

Generation of the projection **PDFs** presently requires significant computational effort making a real time implementation limited. By considering a single traffic vehicle in this thesis and evaluating **PDFs** in a relatively coarse manner, this limitation has been mitigated. However, this does not provide a complete solution.

Reducing the computational burden may be realised by utilising a Particle Filter based technique to recursively determine projection **PDFs**, retaining information from previous calculations. Such a technique also relaxes the assumption that the initial distributions (of velocity, heading and discrete transition) are Gaussian, allowing a more generalised solution.

Removing the Gaussian assumption poses additional challenges, however, as a number of techniques used in this thesis can no longer be applied. The transformation of a Gaussian **BVPDF** into a radial error distribution, for example, is given in (5.10). Creating a radial error distribution from a Particle Filter distribution would require the calculation of radius for each particle, possibly negating

---

any computational complexity improvements.

#### **8.2.4 Decision making fidelity**

The purpose of this thesis was to develop an artificial situation awareness system for an autonomous [UAS](#). In order to demonstrate the utility of such a system, however, it was necessary to implement a decision making system based upon the situation awareness. The limitations in scalability of the situation awareness system (due to the computational complexity described above) limited the decision making to considering a single traffic vehicle. In addition, the use of a ruled based decision making framework limits the ability of the [UAS](#) to react to unusual situations.

A wide array of literature exists in the field of computational decision making, predominately from the field of [AI](#). Once improvements have been made to the scalability of the situation awareness system, it may be possible to implement a more complex decision making system.

#### **8.2.5 Communications integrity**

The real world test environment has seen limited use due to the lack of reliable communication with the vehicle. This is largely due to the limited power of the XBee radios used.

An improvement to the communications system could be realised by moving to an [IEEE 802.11](#) based wireless network. Such a network benefits not only from higher powers, but an increased data rate of up to 3000% compared with XBee.

# Appendix A

## Derivation of haversine formulae

### A.1 Distance between two points on Earth

To determine the distance between two points on earth given in latitude and longitude coordinates it is first necessary to find the angle subtended by the vectors from the centre of the earth to each point. Representing the points in Cartesian frame with the origin at the centre of the Earth

$$X_1 = \begin{bmatrix} R \cos \phi_1 \sin \lambda_1 \\ R \cos \phi_1 \cos \lambda_1 \\ R \sin \phi_1 \end{bmatrix} \quad X_2 = \begin{bmatrix} R \cos \phi_2 \sin \lambda_2 \\ R \cos \phi_2 \cos \lambda_2 \\ R \sin \phi_2 \end{bmatrix} \quad (\text{A.1})$$

where  $R = 6378km$  is the mean radius of the earth,  $\phi$  and  $\lambda$  at the latitude and longitude of the points.

Determine the angle between the vectors extending from the origin to the points using vector dot product

$$X_1 \cdot X_2 = |X_1||X_2| \cos \theta \quad (\text{A.2})$$

Expanding and noting that both points are on the surface of the Earth, therefore the magnitude of the vectors is simply  $R$

$$\cos \phi_1 \sin \lambda_1 \cos \phi_2 \sin \lambda_2 + \cos \phi_1 \cos \lambda_1 \cos \phi_2 \cos \lambda_2 + \sin \phi_1 \sin \phi_2 = \cos \theta \quad (\text{A.3})$$

---

Substituting the identity  $\sin \lambda_1 \sin \lambda_2 + \cos \lambda_1 \cos \lambda_2 = \cos \Delta \lambda$

$$\cos \phi_1 \cos \phi_2 \cos \Delta \lambda + \sin \phi_1 \sin \phi_2 = \cos \theta \quad (\text{A.4})$$

Whilst this may be solved directly for  $\theta$  (and subsequently distance), it should be noted that invoking the cosine function on small angles can introduce numerical errors due to the relatively shallow gradient of the function at that point. It is beneficial to utilise the haversine (from half-versed-sine) transform  $1 - \cos \theta = 2 \sin^2(\theta/2)$  to remove this problem.

First expand the sine product in (A.4) with the identity  $\cos \Delta \phi - \cos(\phi_1 + \phi_2) = 2 \sin \phi_1 \sin \phi_2$

$$\cos \phi_1 \cos \phi_2 \cos \Delta \lambda + \frac{1}{2} \cos \Delta \phi - \frac{1}{2} \cos(\phi_1 + \phi_2) = \cos \theta \quad (\text{A.5})$$

Now introduce the haversine relationship for  $\cos \Delta \lambda$

$$\cos \phi_1 \cos \phi_2 \left( 1 - 2 \sin^2 \frac{\Delta \lambda}{2} \right) + \frac{1}{2} \cos \Delta \phi - \frac{1}{2} \cos(\phi_1 + \phi_2) = \cos \theta \quad (\text{A.6})$$

Expanding the final term on the LHS with  $\cos(\phi_1 + \phi_2) = \cos \phi_1 \cos \phi_2 - \sin \phi_1 \sin \phi_2$

$$\cos \phi_1 \cos \phi_2 \left( 1 - 2 \sin^2 \frac{\Delta \lambda}{2} \right) + \frac{1}{2} \cos \Delta \phi - \frac{1}{2} \cos \phi_1 \cos \phi_2 + \frac{1}{2} \sin \phi_1 \sin \phi_2 = \cos \theta \quad (\text{A.7})$$

Expanding the brackets

$$\frac{1}{2} \cos \phi_1 \cos \phi_2 + \frac{1}{2} \sin \phi_1 \sin \phi_2 - 2 \cos \phi_1 \cos \phi_2 \sin^2 \frac{\Delta \lambda}{2} + \frac{1}{2} \cos \Delta \phi = \cos \theta \quad (\text{A.8})$$

Replacing the identity  $\cos \phi_1 \cos \phi_2 + \sin \phi_1 \sin \phi_2 = \cos \Delta \phi$

$$\cos \Delta \phi - 2 \cos \phi_1 \cos \phi_2 \sin^2 \frac{\Delta \lambda}{2} = \cos \theta \quad (\text{A.9})$$

Introducing the haversine relationship for the remaining cosines of small angles

$$1 - 2 \sin^2 \frac{\Delta \phi}{2} - 2 \cos \phi_1 \cos \phi_2 \sin^2 \frac{\Delta \lambda}{2} = 1 - 2 \sin^2 \frac{\theta}{2} \quad (\text{A.10})$$

---

Simplifying

$$\sin^2 \frac{\Delta\phi}{2} + \cos \phi_1 \cos \phi_2 \sin^2 \frac{\Delta\lambda}{2} = \sin^2 \frac{\theta}{2} \quad (\text{A.11})$$

The inverse tangent function is preferred for small angles, therefore replace the RHS in (A.11) with  $\sin^2 \frac{\theta}{2} = 1 - \cos^2 \frac{\theta}{2}$  and rearrange

$$1 - \sin^2 \frac{\Delta\phi}{2} - \cos \phi_1 \cos \phi_2 \sin^2 \frac{\Delta\lambda}{2} = \cos^2 \frac{\theta}{2} \quad (\text{A.12})$$

Using (A.11) and (A.12), solve for  $\theta$

$$\theta = 2 \tan^{-1} \left( \frac{\sin^2 \frac{\Delta\phi}{2} + \cos \phi_1 \cos \phi_2 \sin^2 \frac{\Delta\lambda}{2}}{1 - \sin^2 \frac{\Delta\phi}{2} - \cos \phi_1 \cos \phi_2 \sin^2 \frac{\Delta\lambda}{2} = \cos^2 \frac{\theta}{2}} \right) \quad (\text{A.13})$$

Finally, multiply by Earth's radius to determine distance between points.

$$d = 2R \tan^{-1} \left( \frac{\sin^2 \frac{\Delta\phi}{2} + \cos \phi_1 \cos \phi_2 \sin^2 \frac{\Delta\lambda}{2}}{1 - \sin^2 \frac{\Delta\phi}{2} - \cos \phi_1 \cos \phi_2 \sin^2 \frac{\Delta\lambda}{2}} \right) \quad (\text{A.14})$$

## A.2 Initial heading between two points on Earth

Transforming the longitudes of both points described in (A.1) to be relative to  $X_1$  yields

$$X_1 = \begin{bmatrix} 0 \\ R \cos \phi_1 \\ R \sin \phi_1 \end{bmatrix} \quad X_2 = \begin{bmatrix} R \cos \phi_2 \sin \Delta\lambda \\ R \cos \phi_2 \cos \Delta\lambda \\ R \sin \phi_2 \end{bmatrix} \quad (\text{A.15})$$

Taking the cross product of  $X_1$  with a unit vector in the direction of the north pole  $N = [0 \ 0 \ 1]^T$

$$N \times X_1 = \begin{bmatrix} -R \cos \phi_1 \\ 0 \\ 0 \end{bmatrix} \quad (\text{A.16})$$

---

and the two points

$$X_2 \times X_1 = \begin{bmatrix} R^2 \sin \phi_1 \cos \phi_2 \cos \Delta\lambda - R^2 \cos \phi_1 \sin \phi_2 \\ -R^2 \sin \phi_1 \cos \phi_2 \sin \Delta\lambda \\ R^2 \cos \phi_1 \cos \phi_2 \sin \Delta\lambda \end{bmatrix} \quad (\text{A.17})$$

Strictly, the initial bearing is equal to the angle between these normal vectors ( $N \times X_1$  being the normal to the plane  $NOX_1$  and  $X_2 \times X_1$  being the normal to the plane  $X_2OX_1$ ), this can be found by taking their dot product. It is simpler, however, to recognise that  $N \times X_1$  is parallel to the (minus)  $x$  axis, therefore calculating the angle between  $X_2 \times X_1$  and the  $x$ -axis is sufficient. This is illustrated in Figure A.1, where  $a$  and  $b$  are the magnitude of  $N \times X_1$  in the  $yz$  plane and along the  $x$  axis respectively.

The tangent of the angle between the normal vectors is therefore

$$\tan \psi = \frac{-a}{b} \quad (\text{A.18})$$

where  $a$  and  $b$  can be found from (A.17)

$$\tan \psi = \frac{-\sqrt{(-R^2 \sin \phi_1 \cos \phi_2 \sin \Delta\lambda)^2 + (R^2 \cos \phi_1 \cos \phi_2 \sin \Delta\lambda)^2}}{R^2 \sin \phi_1 \cos \phi_2 \cos \Delta\lambda - R^2 \cos \phi_1 \sin \phi_2} \quad (\text{A.19})$$

Simplifying the numerator by recognising  $\sin^2 \phi_1 + \cos^2 \phi_1 = 1$ , cancelling  $R^2$  and moving the minus sign to the denominator

$$\tan \psi = \frac{\cos \phi_2 \sin \Delta\lambda}{\cos \phi_1 \sin \phi_2 - \sin \phi_1 \cos \phi_2 \cos \Delta\lambda} \quad (\text{A.20})$$

Therefore the initial bearing is

$$\psi = \tan^{-1} \left( \frac{\cos \phi_2 \sin \Delta\lambda}{\cos \phi_1 \sin \phi_2 - \sin \phi_1 \cos \phi_2 \cos \Delta\lambda} \right) \quad (\text{A.21})$$

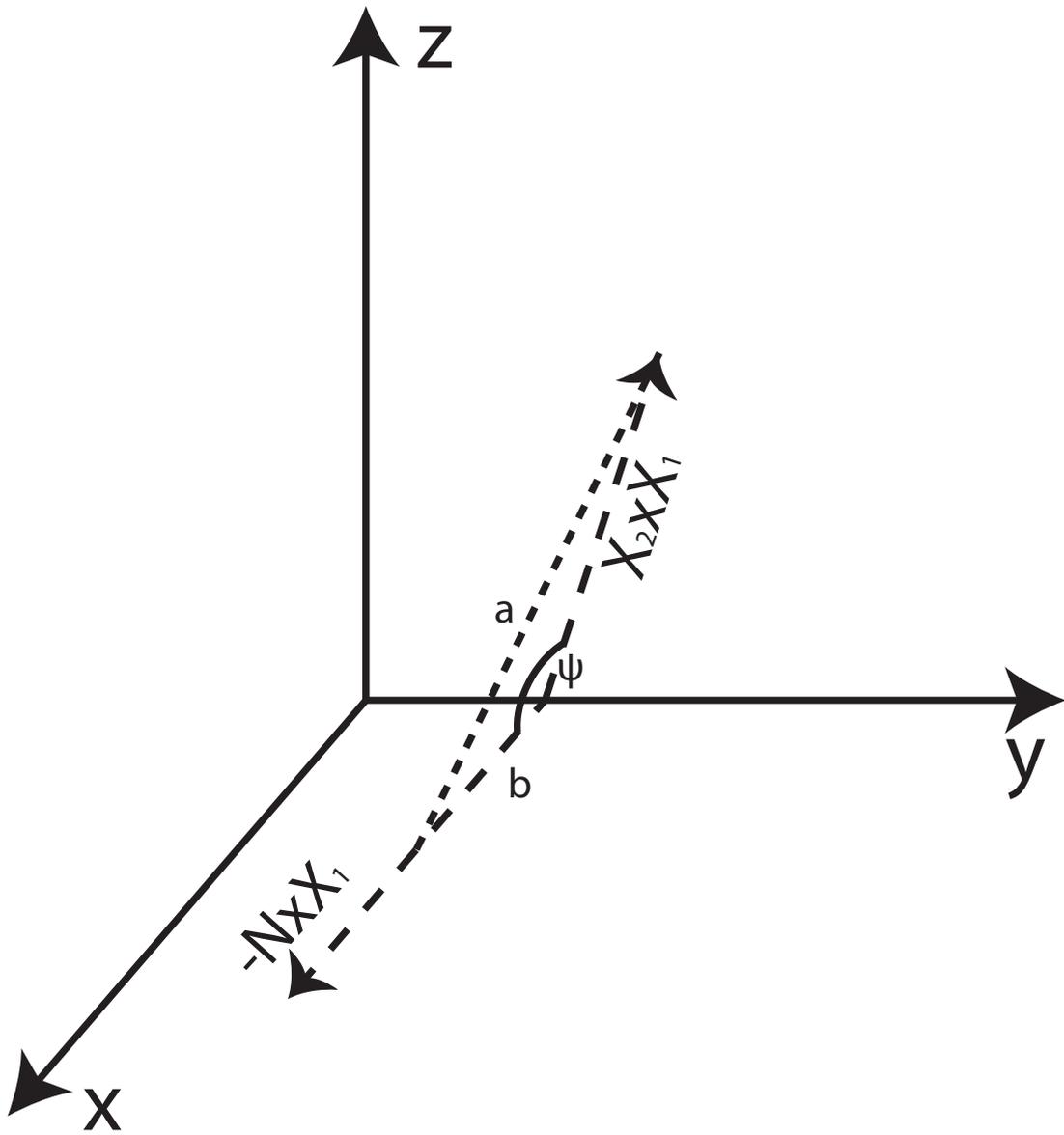


Figure A.1: Calculation of initial bearing from normal vectors

# References

- Adams, J. A. [2007], Unmanned Vehicle Situation Awareness: A Path Forward, Technical report, Vanderbilt University. 18, 19, 20
- Anderson, C. [2010], ‘ArduPilot Mega’, <http://www.diydrones.com/notes/ArduPilot>. Accessed: 26/11/2012. 139
- Anderson, T. W. [1984], *An Introduction to Multivariate Statistical Analysis, 2nd Edition*, 2 edn, Wiley-Interscience. 49, 75, 96, 110
- Bar-Shalom, Y., Li, X. and Kirubarajan, T. [2001], *Estimation with applications to tracking and navigation: theory algorithms and software*, Wiley-Interscience. 37
- Bellingham, J., Richards, A. and How, J. [2002], Receding horizon control of autonomous aerial vehicles, in ‘American Control Conference, 2002. Proceedings of the 2002’, Vol. 5, pp. 3741–3746 vol.5. 7
- Bottasso, C., Leonello, D. and Savini, B. [2008], ‘Path Planning for Autonomous Vehicles by Trajectory Smoothing Using Motion Primitives’, *Control Systems Technology, IEEE Transactions on* **16**(6), 1152–1168. 7
- Burgess, D., Altman, S. and Wood, M. [1994], ‘TCAS: MANEUVERING AIRCRAFT IN THE HORIZONTAL PLANE.’, *Lincoln Laboratory Journal* . 107
- Button, C. [2009], ‘Unmanned Aerial Vehicles on Operations: Overcoming the Challenges’, *RUSI Defence Systems* **12**(1), 76–78. 31
- Cantwell, H. [2009], ‘Operators of Air Force Unmanned Aircraft Systems: Breaking Paradigms’, *Airpower Journal* **23**(2), 67. 7

## REFERENCES

---

- Carnie, R., Walker, R. and Corke, P. [2006], Image processing algorithms for UAV, *in* 'Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on', IEEE, pp. 2848–2853. [12](#)
- Civil Aviation Authority [2001], Noise From Arriving Aircraft, An Industry Code of Practice, Technical report, Civil Aviation Authority. [29](#)
- Civil Aviation Authority [2007], CAP773 - Flying RNAV (GNSS) Non-Precision Approaches in Private and General Aviation Aircraft, Technical report, Civil Aviation Authority. [29](#)
- Civil Aviation Authority [2008], UK ATS Airspace Classifications, Technical report, Civil Aviation Authority. [25](#), [28](#)
- Civil Aviation Authority [2009*a*], CAP413 - Radiotelephony Manual, Technical report, Civil Aviation Authority. [10](#), [13](#), [25](#), [28](#), [69](#), [71](#), [107](#), [125](#), [128](#)
- Civil Aviation Authority [2009*b*], 'The Standard "Overhead" Join', *GA Safety Poster* . [27](#), [125](#)
- Civil Aviation Authority [2010*a*], CAP393 - Air Navigation: The Order and the Regulations, Technical report, Civil Aviation Authority. [12](#), [25](#), [32](#), [33](#)
- Civil Aviation Authority [2010*b*], CAP722 - Unmanned Aircraft System Operations in UK Airspace - Guidance, Technical report, Civil Aviation Authority. [9](#), [10](#), [12](#), [13](#), [22](#), [24](#), [31](#), [116](#)
- Civil Aviation Authority [2010*c*], Establishment and Dimensions of Aerodrome Traffic Zones (ATZ), Technical report, Civil Aviation Authority. [25](#)
- Civil Aviation Authority [2011*a*], Guide to Visual Flight Rules (VFR) in the UK, Technical report, Civil Aviation Authority. [28](#)
- Civil Aviation Authority [2011*b*], Safety Sense Leaflet 6e - Aerodrome Sense, Technical report, Civil Aviation Authority. [125](#), [128](#)
- Civil Aviation Authority [2012], Instrument Approach Chart - ICAO. Cranfield ILS/DME/NDB(L) RWY 21, Technical report, Civil Aviation Authority. [30](#)

## REFERENCES

---

- Civil Aviation Authority [2013], CAP493 - Manual of Air Traffic Services - Part 1 Amendment 14, Technical report, Civil Aviation Authority. [107](#)
- Civil Aviation Authority of New Zealand [1999], ‘Joining Overhead’, *Vector - Pointing to Safer Aviation* **5**, 14–15. [125](#)
- Civil Aviation Safety Authority [2010], Non-towered Aerodrome Circuit Procedures, Technical report, Australian Government. [126](#), [127](#)
- Coombes, M., McAree, O. and Chen, W.-H. [2012], Development of an autopilot system for rapid prototyping of high level control algorithms, *in* ‘UKACC International Conference on Control’. [152](#)
- Craighead, J., Murphy, R., Burke, J. and Goldiez, B. [2007], A Survey of Commercial Open Source Unmanned Vehicle Simulators, *in* ‘Robotics and Automation, 2007 IEEE International Conference on’, pp. 852–857. [146](#)
- Crisostomi, E., Lecchini Visintini, A. and Maciejowski, J. [2007], ‘Combining Monte Carlo and worst-case methods for trajectory prediction in air traffic control: a case study’, *Automatic Control in Aerospace* **2**, 259–269. [23](#)
- Cummings, M., Clare, A. and Hart, C. [2010], ‘The role of human-automation consensus in multiple unmanned vehicle scheduling’, *Human Factors: The Journal of the Human Factors and Ergonomics Society* **52**(1), 17–27. [7](#)
- Department of Defence [2005], ‘UAV Roadmap 2005-2030’, *Office of the Secretary of Defense* p. 11. [7](#), [11](#)
- Department of Defence [2007], ‘UAV Roadmap 2007-2032’, *Office of the Secretary of Defense* p. 11. [8](#), [9](#), [10](#), [11](#)
- Eckler, A. R. [1969], ‘A Survey of Coverage Problems Associated with Point and Area Targets’, *Technometrics* **11**(3), pp. 561–589. [112](#)
- Eele, A. and Richards, A. [2009], ‘Path-Planning with Avoidance Using Non-linear Branch-and-Bound Optimization’, *Journal of Guidance, Control, and Dynamics (0731-5090)* **32**, 384–394. [7](#)

- Endsley, M. [1988], Design and evaluation for situation awareness enhancement, *in* ‘Human Factors Society, Annual Meeting, 32 nd, Anaheim, CA’, pp. 97–101. [15](#)
- Endsley, M. [1993], ‘A survey of situation awareness requirements in air-to-air combat fighters’, *The International Journal of Aviation Psychology* **3**(2), 157–168. [31](#)
- Endsley, M. [1997], Situation awareness, automation and free-flight, *in* ‘USA/Europe ATM R&D Seminar’. [22](#)
- Endsley, M., Farley, T., Jones, W., Midkiff, A. and Hansman, R. [1998], Situation awareness information requirements for commercial airline pilots, Technical report, International Center for Air Transportation. [31](#)
- Endsley, M. R. [1995], ‘Toward a Theory of Situational Awareness in Dynamic Systems’, *Human Factors* **37**, 32–64. [15](#), [16](#), [17](#), [19](#)
- Endsley, M. and Rodgers, M. [1994], Situation awareness information requirements analysis for en route air traffic control, *in* ‘Proceedings of the Human Factors and Ergonomics Society Annual Meeting’, Vol. 38, SAGE Publications, pp. 71–75. [31](#)
- Fitzgerald, D., Walker, R. and Campbell, D. [2005], A vision based forced landing site selection system for an autonomous UAV, *in* ‘Intelligent Sensors, Sensor Networks and Information Processing Conference, 2005. Proceedings of the 2005 International Conference on’, IEEE, pp. 397–402. [15](#)
- Flint, M., Polycarpou, M. and Fernandez-Gaucherand, E. [2002], Cooperative control for multiple autonomous UAV’s searching for targets, *in* ‘Decision and Control, 2002, Proceedings of the 41st IEEE Conference on’, Vol. 3, pp. 2823–2828 vol.3. [11](#)
- Fowers, S., Lee, D.-J., Tippetts, B., Lillywhite, K., Dennis, A. and Archibald, J. [2007], Vision Aided Stabilization and the Development of a Quad-Rotor Micro UAV, *in* ‘Computational Intelligence in Robotics and Automation, 2007. CIRA 2007. International Symposium on’, pp. 143–148. [13](#)

## REFERENCES

---

- Geyer, C., Singh, S. and Chamberlain, L. [2008], ‘Avoiding collisions between aircraft: State of the art and requirements for UAVs operating in civilian airspace’, *Robotics Institute, Carnegie Mellon University, Tech. Rep. CMU-RI-TR-08-03*. 13, 22
- Grubbs, F. E. [1964], ‘Approximate Circular and Noncircular Offset Probabilities of Hitting’, *Operations Research* **12**(1), pp. 51–62. 112
- Hayles, J. [2006], ‘UK Police Aviation’, <http://www.aeroflight.co.uk/waf/uk/police/ukaf-police-home.htm>. Accessed: 01/11/2012. 31
- Hoekstra, J., Ruigrok, R. and Van Gent, R. [2001], ‘Free flight in a crowded airspace?’, *Progress in Astronautics and Aeronautics* **193**, 533–546. 22
- IEEE [2008], IEEE Standard for Floating-Point Arithmetic, Technical report, Microprocessor Standards Committee of the IEEE Computer Society. 144
- International Telecommunications Union [1996], Recommendation X25, Technical report, International Telecommunications Union. 145
- Ippolito, C. and Pisanich, G. [2005], Cognitive Emotion Layer Architecture for Intelligent UAV Planning, Behavior and Control, *in* ‘Aerospace Conference, 2005 IEEE’, pp. 1–16. 9
- Krozel, J. [2000], Intent inference for free flight aircraft, *in* ‘AIAA Guidance, Navigation, and Control Conf., Denver, CO’. 22
- Kuchar, J. and Drumm, A. [2007], ‘The Traffic alert and Collision Avoidance System’, *Lincoln Laboratory Journal* **16**(2), 277. 23, 68
- Kuwata, Y., Richards, A., Schouwenaars, T. and How, J. [2007], ‘Distributed Robust Receding Horizon Control for Multivehicle Guidance’, *Control Systems Technology, IEEE Transactions on* **15**(4), 627–641. 11
- Lacher, A., Zeitlin, A., Maroney, D., Markin, K., Ludwig, D. and Boyd, J. [2010], Airspace Integration Alternatives for Unmanned Aircraft, *in* ‘Association for Unmanned Vehicle Systems International (AUVSI)s Unmanned Systems Asia-Pacific’. 32

## REFERENCES

---

- Laminar Research [2012a], ‘Datarefs for X-Plane 1010’, <http://www.xsquawkbox.net/xpsdk/docs/DataRefs.html>. Accessed: 26/11/2012. 147
- Laminar Research [2012b], ‘X-Plane 10 Global — The World’s Most Advanced Flight Simulator — X-Plane.com’, <http://www.x-plane.com>. Accessed: 26/11/2012. 144
- Liu, W. and Hwang, I. [2011], ‘Probabilistic Trajectory Prediction and Conflict Detection for Air Traffic Control’, *Journal of Guidance Control and Dynamics* **34**, 1779–1789. 23
- Manning, S., Rash, C., LeDuc, P., Noback, R. and McKeon, J. [2004], The role of human causal factors in US Army unmanned aerial vehicle accidents, Technical report, DTIC Document. 12
- MAVLink [2012], ‘MAVLink Micro Air Vehicle Communication Protocol - QGroundControl GCS’, <http://qgroundcontrol.org/mavlink/start>. Accessed: 26/11/2012. 144
- Meier, L., Tanskanen, P., Fraundorfer, F. and Pollefeys, M. [2011], Pixhawk: A system for autonomous flight using onboard computer vision, *in* ‘Robotics and automation (ICRA), 2011 IEEE international conference on’, IEEE, pp. 2992–2997. 143
- Menon, R., Sweriduk, G. and Sridhar, B. [1999], ‘Optimal strategies for free-flight air traffic conflict resolution’, *Journal of Guidance Control and Dynamics* **22**, 202–211. 22
- Microsoft [2011], ‘Windows Sockets 2 (Windows)’, [http://msdn.microsoft.com/en-gb/library/windows/desktop/ms740673\(v=vs.85\).aspx](http://msdn.microsoft.com/en-gb/library/windows/desktop/ms740673(v=vs.85).aspx). Accessed: 26/11/2012. 148
- Moore, B. C. J. [1996], Information Extraction and Perceptual Grouping in the Auditory System, *in* ‘Proceedings of the Second International Workshop on Human and Machine Perception’. 10

## REFERENCES

---

- National Transportation Safety Board [2006], Annual Review of Aircraft Accident Data: US General Aviation, Calendar Year 2003, Technical report, NTSB. [13](#), [14](#), [25](#)
- Negnevitsky, M. [2005], *Artificial intelligence: a guide to intelligent systems*, Addison-Wesley Longman. [10](#), [21](#), [24](#)
- Pack, D., DeLima, P., Toussaint, G. and York, G. [2009], ‘Cooperative Control of UAVs for Localization of Intermittently Emitting Mobile Targets’, *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* **39**(4), 959–970. [11](#)
- Patel, J. and Read, C. [1982], *Handbook of the normal distribution*, Statistics, textbooks and monographs, M. Dekker. [121](#)
- Pfoser, D. and Tryfona, N. [2001], Capturing Fuzziness and Uncertainty of Spatiotemporal Objects, in A. Caplinskas and J. Eder, eds, ‘Advances in Databases and Information Systems’, Vol. 2151 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 112–126. [23](#)
- Rao, S. S. [2009], *Engineering optimization: theory and practice*, Wiley. [96](#)
- Reichard, K. [2004], ‘Integrating self-health awareness in autonomous systems’, *Robotics and Autonomous Systems* **49**(1), 105–112. [14](#)
- RTCA [2002], Minimum Aviation System Performance Standards for Automatic Dependent Surveillance-Broadcast (ADS-B), Technical report, DO-242A. [12](#), [32](#)
- Samek, J., Sislak, D., Volf, P. and Pechoucek, M. [2007], Multi-party Collision Avoidance among Unmanned Aerial Vehicles, in ‘Intelligent Agent Technology, 2007. IAT ’07. IEEE/WIC/ACM International Conference on’, pp. 403–406. [11](#)
- Seah, C. E. and Hwang, I. [2009], ‘Terminal-Area Aircraft Tracking Using Hybrid Estimation’, *Journal of Guidance, Control, and Dynamics* **32**(3), 836–849. [23](#), [35](#), [39](#), [50](#)

## REFERENCES

---

- Sinclair, A. J., Prazenica, R. J. and Jeffcoat, D. E. [2008], ‘Optimal and Feedback Path Planning for Cooperative Attack’, *Journal of Guidance, Control, and Dynamics (0731-5090)* **31**, 1708–1715. [11](#)
- Thom, T. [2003], *Radio Navigation & Instrument Flying*, Air Pilot Publishing. [25](#)
- Transportation Accident Investigation Commission [2008], Report 08-001, Cessna 152 ZK-ETY and Robinson R22 ZK-HGV, mid-air collision, Paraparamu, Technical report, Transportation Accident Investigation Commission. [125](#)
- Vachtsevanos, G., Tang, L., Drozeski, G. and Gutierrez, L. [2005], ‘From mission planning to flight control of unmanned aerial vehicles: Strategies and implementation tools’, *Annual Reviews in Control* **29**(1), 101–115. [8](#)
- Weil, H. [1954], ‘The Distribution of Radial Error’, *The Annals of Mathematical Statistics* **25**(1), 158–170. [108](#), [111](#)
- Weitz, L. A., Hurtado, J. E. and Sinclair, A. J. [2008], ‘Decentralized Cooperative-Control Design for Multivehicle Formations’, *Journal of Guidance, Control, and Dynamics (0731-5090)* **31**, 970–979. [11](#)
- Welch, G. and Bishop, G. [2001], *An Introduction to the Kalman Filter*, ACM. [19](#), [51](#)
- Wickens, C. [2002], ‘Situation awareness and workload in aviation’, *Current directions in psychological science* **11**(4), 128–133. [18](#)
- Wickens, C., Gordon, S. and Liu, Y. [2004], *An introduction to human factors engineering*, Pearson Prentice Hall Upper Saddle River, NJ. [19](#), [21](#)
- Wilson, E. B. and Hilferty, M. M. [1931], ‘The Distribution of Chi-square’, *Proceedings of The National Academy of Sciences* **17**, 684–688. [112](#)