Embedding Human Expert Cognition and Real-Time Trajectory Planning in Autonomous UAS

Pritesh Narayan

Australian Research Centre for Aerospace Automation

Queensland University of Technology

A thesis submitted for the degree of

Doctor of Philosophy (PhD)

2011 September

Declaration

This is to certify that the work presented in this thesis has not been previously submitted for an award at this or any other higher education institution. To the best of my knowledge and belief, the thesis comprises of only my original work towards the PhD except where indicated, and due acknowledgement has been made in the text to all other material used.

_____

Pritesh Praneet Narayan

_____

Date

Abstract

This thesis presents a new approach to compute and optimize feasible three dimensional (3D) flight trajectories using aspects of Human Decision Making (HDM) strategies, for fixed wing Unmanned Aircraft (UA) operating in low altitude environments in the presence of real time planning deadlines. The underlying trajectory generation strategy involves the application of Manoeuvre Automaton (MA) theory to create sets of candidate flight manoeuvres which implicitly incorporate platform dynamic constraints. Feasible trajectories are formed through the concatenation of predefined flight manoeuvres in an optimized manner.

During typical UAS operations, multiple objectives may exist, therefore the use of multi-objective optimization can potentially allow for convergence to a solution which better reflects overall mission requirements and HDM preferences. A GUI interface was developed to allow for knowledge capture from a human expert during simulated mission scenarios. The expert decision data captured is converted into value functions and corresponding criteria weightings using UTilité Additive (UTA) theory. The inclusion of preferences elicited from HDM decision data within an Automated Decision System (ADS) allows for the generation of trajectories which more closely represent the candidate HDM's decision strategies.

A novel Computationally Adaptive Trajectory Decision optimization System (CATDS) has been developed and implemented in simulation to dynamically manage, calculate and schedule system execution parameters to ensure that the trajectory solution search can generate a feasible solution, if one exists, within a given length of time. The inclusion of the CATDS potentially increases overall mission efficiency and may allow for the implementation of the system on different UAS platforms with varying onboard computational capabilities.

These approaches have been demonstrated in simulation using a fixed wing UAS operating in low altitude environments with obstacles present.

Thesis supervisor: D. Campbell

Title: Associate Professor

Contents

List of Figures **ix**

List of Tables **xiv**

Glossary **xv**

1 Introduction **1**

1.1 Integration of UAS into the NAS 2

1.1.1 Demonstrating ELOS 3

1.1.2 Compliance with existing aviation regulations 3

1.1.3 Appearing transparent to other airspace users 5

1.2 UAS Autonomy 5

1.2.1 Increasing onboard autonomy levels 7

1.2.2 Autonomous low altitude UAS operations in the NAS . . . 9

1.2.2.1 Incorporating platform dynamics and collision free guarantees 10

1.2.2.2 Trajectory optimisation 10

1.2.2.3 Real-time computation constraints on planning . 12

1.3 Research statement 12

1.3.1 Research questions 13

1.3.2 Research Objectives 14

1.3.3 Research Outcomes 14

1.3.4 Research Methodology 15

1.3.5 Research Contributions 17

1.4 Publications 18

1.5 Format of thesis 19

2 Literature Review **22**
2.1 Intelligent control architectures 23
2.1.1 Review of UAS intelligent control architectures 24
2.1.2 Summary of findings 27
2.2 Review of UAS trajectory planning methodologies 27
2.2.1 Spline based trajectory planning 28
2.2.2 Geometric trajectory planning 29
2.2.3 Summary of findings 33
2.3 Automated UA operations in partially known environments 34
2.3.1 Summary of findings 37
2.4 Candidate trajectory planning solution 38
2.4.1 Incorporation of platform dynamics 38
2.4.2 Real-time constraints on computation time 39
2.4.3 Trajectory optimisation to meet given mission requirements 40
2.5 Summary of Findings 41

3 Flight Trajectories for Fixed Wing UA using Manoeuvre Automaton (MA) Theory **42**
3.1 Manoeuvre automaton theory 42
3.1.1 Trim primitives 43
3.1.2 Manoeuvre primitives 44
3.2 Trim primitive formulation 44
3.2.1 Straight and level flight (cruise) 45
3.2.2 Coordinated turn 45
3.2.3 Constant climb 46
3.2.4 Controlled descent 47
3.2.5 Helical ascent 49
3.2.6 Helical descent 49
3.3 Manoeuvre primitive implementation 51
3.3.1 Roll rate constraints during pure rolling motion 51
3.3.2 Pitch rate constraints during pure pitching motion 54
3.3.3 Roll and pitch rate constraints during helical manoeuvres . 55

3.4 Trajectory tracking of feasible trajectories 58
3.5 Ensuring platform safety during trajectory planning 61
3.5.1 Safe states in 3D partially known environments 61
3.6 Generating optimised trajectories through concatenation 65
3.6.1 Dynamic programming applied to MA based trajectory planning 66
3.6.2 Application of DP to this research project 67
3.7 Summary of findings 68

4 Embedding Human Expert Cognition into Trajectory Planning 70
4.1 MCDA process 71
4.1.1 Determining relevant criteria and alternatives 72
4.1.1.1 Planning space 72
4.1.1.2 Alternatives 73
4.1.1.3 Criteria 74
4.1.2 Evaluating alternatives on all criteria 75
4.1.3 Eliciting preferences from candidate HDM decision data . 76
4.1.3.1 Overview of UTA theory 77
4.1.4 Determining a ranking of the alternatives 78
4.1.5 Summary of findings 80
4.2 MCDA and mission priorities 80
4.2.1 Expert knowledge capture and decision modeling strategies 80
4.2.2 Preference formulation using UTA 81
4.2.2.1 LC-2 83
4.2.2.2 UTA-2 84
4.2.2.3 UTA-4 86
4.2.3 Accuracy of UTA 86
4.3 Results 91
4.3.1 Simulation setup 91
4.3.2 Preference selection during online planning 93
4.3.3 Simulation results 94
4.3.3.1 Terrain simulation 1 94
4.3.3.2 Terrain simulation 2 94

4.4 Discussion 96

5 Computationally Adaptive Real-Time Trajectory Planning 99
5.1 Presence of real-time deadlines during MA trajectory planning . . 100
5.1.1 Simulation setup 101
5.1.2 FPW during online planning in partially known environments 103
5.1.3 Presence of real-time constraints during simulation 104
5.1.4 Summary of findings 106
5.2 Development of CATDS optimisation system 108
5.2.1 CATDS offline component 109
5.2.2 CATDS online component 110
5.2.3 Applying a minimum FPW 113
5.2.4 Summary of findings 116
5.3 Inclusion of HDM preferences during real-time planning 116
5.3.1 Results 117
5.3.1.1 Inclusion of HDM 2 Data via UTA-4 119
5.3.1.2 Inclusion of HDM 3 Data via UTA-4 120
5.4 Discussion 120

6 Conclusions 123
6.1 Thesis summary 123
6.2 Contributions 125
6.3 Future work 127

Bibliography 129

Appendices 142

A Human Expert Data Capture 143
A.1 HDM Information 143
A.2 Graphical User Interface 143
A.3 Decision Scenarios 145

B Preference elicitation from HDM decision data **149**
B.1 Alternative subset formulation 149
B.2 Selecting global preferences from multiple subsets 150
B.2.1 HDM 1 152
B.2.2 HDM 2 153
B.2.3 HDM 3 154
B.2.4 HDM 4 155

C Offline HDM and ADS Decision Analysis **156**
C.1 HDM 1 157
C.2 HDM 2 158
C.3 HDM 3 159
C.4 HDM 4 160

D Chapter 4 Simulation Results **161**
D.1 Terrain simulation 1 162
D.2 Terrain simulation 2 164

E Publications **166**

List of Figures

1.1 Aerosonde UA 2
1.2 Australian civilian NAS (June 2004) [1] 4
1.3 Australian NAS long term strategy [1] 5
1.4 Cognitive model of a pilot's decision mMaking process [2] 7
1.5 Global planning visualisation [3] 8
1.6 Proposed shift in control paradigms with increasing levels of system autonomy [4] 11

2.1 Definition of intelligent control discipline [5] 23
2.2 3T intelligent control architecture [6] 24
2.3 Boskovic's UAS decision making architecture [7] - image courtesy of Paul Wu 26
2.4 Extension of Dubins Paths 3D configuration space [8] 30
2.5 Example of RRT expansion during trajectory planning [9] 31
2.6 Schouwenaars MILP based trajectory planning algorithm with implicit safety guarantees [10] 32
2.7 Kuwata's MILP based trajectory planning algorithm [10] 33
2.8 Sherer's architecture for reactive collision avoidance [11] 35

3.1 SL trim primitive example 45
3.2 CT trim primitive example 46
3.3 CC trim primitive example 47
3.4 CD trim primitive example 48
3.5 HA trim primitive example 49
3.6 HD trim primitive example 50

3.7 Concatenation of two CT trim primitives without inclusion of platform attitude rate constraints 52
3.8 $\dot{\phi}$ modelled as a first order response for aerosonde UAS platform travelling at $V = 30m/s$ 52
3.9 ϕ during concatenation of CT trim primitives without and with inclusion of transition manoeuvre respectively 53
3.10 Concatenation of two CT trim primitives with inclusion of attitude rate constraints 54
3.11 θ during concatenation of SL to CC trim primitive without and with inclusion of attitude rates respectively where $V = 30m/s$ . . 56
3.12 Concatenation of SL to HC trim primitive with inclusion of attitude rate constraints where $V = 30m/s$ 57
3.13 ϕ and θ during concatenation of SL to HC trim primitive without and with inclusion of attitude rate constraints respectively 57
3.14 Inclusion of trajectory tracking layer within UAS guidance and control software architecture developed by Beard et. al [12] 58
3.15 Comparison of nominal, predicted (using MPC) and actual UA trajectory by Topsakal [13] 59
3.16 Comparison of Kuwata's MILP based UA trajectory planning algorithm without and with predicted position adjustment due to external disturbances [14] 60
3.17 Execution of fixed wing UA hold manoeuvres where attitude rate constraints are included via MA theory 62
3.18 Generation of hold manoeuvres for automaton 63
3.19 Valid states where the corresponding hold manoeuvres are executed within the finite horizon 64
3.20 Safe states within finite horizon 65

4.1 Discrete jump times for a coordinated turn trim primitive 74
4.2 Graphical User Interface (GUI) developed for HDM data capture 82
4.3 Example decision scenario presented to HDM 82
4.4 Decision sets completed by HDM 83
4.5 Normalised aggregated decision values for all criteria (LC-2) . . . 84

4.6 UTA value functions (UTA-2 with HDM 2 dataset) representing HDM preferences for sample decision scenario 85
4.7 Normalised aggregated decision values for all criteria (UTA-2 with HDM 2 dataset)) 85
4.8 UTA value functions (UTA-4 with HDM 2 dataset) representing HDM preferences for sample decision scenario 87
4.9 Normalised aggregated decision values for all criteria (UTA-4 with HDM 2 dataset) 88
4.10 Comparison of HDM and automated trajectory decisions (LC-2) (UTA-4) for sample decision scenario (Figure 4.3) 88
4.11 Average error comparison between human and automated trajectory decisions for all decision sets 89
4.12 Box plots comparing UAS platform ϕ for offline trajectories selected by HDM 2, ADS LC-2 and ADS UTA-4 solutions 90
4.13 Box plots comparing UAS platform $|g_z - s_z|$ for offline trajectories selected by HDM 3, ADS LC-2 and ADS UTA-4 solutions 91
4.14 Simulated mission environment (terrain simulation 1) 92
4.15 UAS platform altitude during simulation (UTA-4 with HDM 3 dataset) (terrain simulation 1) 95
4.16 Comparing trajectories from LC-2 solution and UTA-4 with HDM 3 dataset (terrain simulation 1) 95
4.17 Comparing UAS platform ΔAltitude at goal for LC-2 solution and UTA-4 with HDM 3 dataset (terrain simulation 1) 96
4.18 Comparing Trajectories from LC-2 solution and UTA-4 with HDM 2 Dataset (terrain simulation 2) 97
4.19 Comparing UAS Platform Mean and Maximum ϕ (per stage) for LC-2 solution and UTA-4 with HDM 2 Dataset (terrain simulation 2) 97

5.1 FPW during single stage MA based trajectory planning in cluttered environment 101
5.2 Simulated 3D environment representing low altitude urban terrain 102
5.3 Waypoint capture represented as spherical regions 102

5.4 Comparison between smaller ($1350A$) and larger($41800A$) fixed wing automatons . . . 103
5.5 Simulated 3D trajectory generated using $5700A$ parameters 106
5.6 Comparison of segment computation time and FPW for $5700A$ simulation on system 1 . . . 107
5.7 Time to compute offline solution with respect to p number and p resolution for system 1 . . . 110
5.8 Time to compute offline solution with respect to A resolution for system 1 . . . 111
5.9 Segment A resolution with CATDS enabled where $R_{sphere} = 250$m on System 1 and 2 respectively . . . 112
5.10 Comparison of trajectory compute time and FPW with CATDS enabled where $R_{sphere} = 250$ on System 1 . . . 113
5.11 Segment FPW with CATDS enabled and $FPW_{min} = 2$s on system 1 115
5.12 Segment computation and idle times with CATDS enabled and $FPW_{min} = 2$s on system 1 . . . 115
5.13 Waypoints representing desired path to goal . . . 117
5.14 Trajectory generated by LC-2 solution with CATDS enabled . . . 118
5.15 Segment computation and idle times with CATDS enabled and $FPW_{min} = 1$s on system 1 for LC-2 solution . . . 118
5.16 Trajectory generated by UTA-4 solution with CATDS enabled using HDM 2 dataset . . . 119
5.17 Comparing UAS Platform Mean and Maximum ϕ (per stage) for LC-2 solution and UTA-4 with HDM 2 Dataset . . . 120
5.18 Trajectory generated by UTA-4 solution with CATDS enabled using HDM 3 dataset . . . 121
5.19 Comparing UAS Platform relative Δ altitude at goal for LC-2 solution and UTA-4 with HDM 3 Dataset . . . 121

6.1 Illustration of trajectory planning research presented in this thesis 126

A.1 Data Capture . . . 144

B.1 Comparison of ($\Delta_{|g-s|}$), (Δ_{ϕ}) and (Δ_{psi}) for HDM 1 with respect to the number of A_R sets included in (B.1) for LC-2 and UTA-4 . 152
B.2 Comparison of ($\Delta_{|g-s|}$), (Δ_{ϕ}) and (Δ_{psi}) for HDM 2 with respect to the number of A_R sets included in (B.1) for LC-2 and UTA-4 . 153
B.3 Comparison of ($\Delta_{|g-s|}$), (Δ_{ϕ}) and (Δ_{psi}) for HDM 3 with respect to the number of A_R sets included in (B.1) for LC-2 and UTA-4 . 154
B.4 Comparison of ($\Delta_{|g-s|}$), (Δ_{ϕ}) and (Δ_{psi}) for HDM 4 with respect to the number of A_R sets included in (B.1) for LC-2 and UTA-4 . 155

C.1 Box plots comparing HDM 1 and corresponding automated decisions 157
C.2 Box plots comparing HDM 2 and corresponding automated decisions 158
C.3 Box plots comparing HDM 3 and corresponding automated decisions 159
C.4 Box plots comparing HDM 4 and corresponding automated decisions 160

D.1 Comparing UA trajectories generated for LC-2 solution and UTA-4 with HDM 2 dataset 162
D.2 Comparing UA trajectories generated for LC-2 solution and UTA-4 with HDM 3 dataset 163
D.3 Comparing UA trajectories generated for LC-2 solution and UTA-4 with HDM 2 dataset 164
D.4 Comparing UA trajectories generated for LC-2 solution and UTA-4 with HDM 3 dataset 165

List of Tables

1.1 Levels of automation of decision and action selection [15] 6

4.1 Primitive type, number and samples per primitive applied during online simulations 93

5.1 A applied during simulation 103
5.2 Computation results for system 1 where $R_{sphere} = 250$m 105
5.3 Computation results for system 2 where $R_{sphere} = 250$m 105
5.4 A resolutions which form matrix of expected times 110
5.5 Computation results for real-time MA based planning with CATDS enabled 112
5.6 Computation results with CATDS enabled and FPW_{min} present for System 1 114
5.7 Computation results with CATDS enabled and $FPW_{min} = 1$s on System 1 117

A.1 HDM aircraft operational experience information 143
A.2 Computation results with CATDS enabled and $FPW_{min} = 1$s on System 1 148

Glossary

ADS Automated Decision System

ADS-B Automatic Dependant Surveillance Broadcast

ARCAA Australian Research Centre for Aerospace Automation

ATC Air Traffic Control

B-spline Basis spline

CATDS Compuationally Adaptive Trajectory Decision optimisation System

CPU Central Processing Unit

DoD Department of Defence

DP Dynamic Programming

EASA European Aviation Safety Agency

ELOS Equivalent Level Of Safety

FAA Federal Aviation Authority

FDA Fault Detection and Accommodation

FPW Finite Planning Window

HDM Human Decision Maker

IFR Instrument Flight Rules

MA Maneouvre Automaton

MACBETH Measuring Attractiveness by a Categorical Based Evaluation Technique

MATLAB MATrix LABoratory

MAVT Multiple Attribute Value Theory

MCDA Multi-Criteria Decision Aid

MILP Mixed Integer Linear Programming

MPC Model Predictive Control

NAS National Airspace System

QUT Queensland University of Technology

RRT Rapidly Exploring Random Trees

SAS Stability Augmentation System

UA Unmanned Aircraft

UAS Unmanned Aerial System

US United States of America

UTA UTilité Additive

1

Introduction

The United States of America's (US) Department of Defence (DoD) defines [16; 17] an Unmanned Aircraft (UA) as *"A powered, aerial vehicle that does not carry a human operator, uses aerodynamic forces to provide vehicle lift, can fly autonomously or be piloted remotely, can be expendable or recoverable, and can carry a lethal or non-lethal payload."*

The DoD, US Federal Aviation Authority (FAA) and the European Aviation Safety Agency (EASA) have progressively adopted the term Unmanned Aerial System (UAS) to signify that UA are part of a complete system consisting of ground control systems, communication links and launch and retrieval systems in addition to the UA [18]. The UA or platform is considered as the primary component of the UAS [19]. Figure 1.1 presents an example of a UA designed by Aerosonde for civilian operations.

Shaefer [20] states that the promise of reduced manufacturing costs, reduced risk to human life and reduced support costs make UA extremely attractive alternatives to manned aircraft for "dull" or potentially dangerous tasks [21]. This statement is reflected by the employment of UAS in an increasingly diverse range of applications. Numerous UAS market forecasts portray a burgeoning future for the UAS, including predictions of a USD10.6 billion market by 2013 [22].

The UAS sector has had the most dynamic growth of the world aerospace industry last decade (2001 - 2010) according to Teal Group, with worldwide UAS expenditure estimated to double this decade to USD11.3 billion annually [23].

Figure 1.1: Aerosonde UA

Finnegan [23] states that "UAS have proved their value in Iraq and Afghanistan and will be a high priority for militaries in the United States and worldwide."

Whilst UAS have proven their capability within military fields, the benefits in civilian applications are only just beginning to be understood [24]. Geographically sparse countries, such as Australia, have great potential for utilisation of UAS in a wide range of civilian applications [25]. Such applications can include asset management, search and rescue, remote sensing, monitoring wildlife and atmospheric observation [26; 27].

To realise these civilian applications, seamless operation of UAS within the National Airspace System (NAS) will be required. The NAS refers to the network of a particular country's airspace, air navigation facilities, equipment and services, airports or landing areas [18]. The following section further details the integration of UAS into the NAS.

1.1 Integration of UAS into the NAS

Most literature [28; 29] indicate that an Equivalent Level Of Safety (ELOS) to that of a human pilot will be one of the requirements for integration of UAS into

the NAS. Operation of UAS in the NAS creates a new set of challenges that are not applicable to many military applications. From a regulatory perspective [30], UAS need to:

1. demonstrate an ELOS to that of a human piloted aircraft,
2. operate in compliance with existing aviation regulations and
3. appear transparent to other airspace users

The following sections elaborate on these requirements for the seamless integration and operation of UAS in the NAS.

1.1.1 Demonstrating ELOS

Taking the human pilot out of an aircraft removes much sensory and decision making capability. The absence of an observing, reacting and decision-making pilot onboard UA has resulted in higher loss rates when compared to manned aircraft [31]. Factors such as weather changes, errors in terrain databases, encountering previously unknown threats and the impacts of vehicle subsystem failures are difficulties that human pilots routinely deal with, but are beyond the capability of most current UAS [20].

To demonstrate an ELOS to that of piloted aircraft, UA must exhibit a high level of autonomy without a human in the loop. Thus, a higher degree of onboard autonomy is required to replicate some of the sensory and decision making capabilities of a human pilot. Furthermore, in the presence of communications failures, the inclusion of onboard autonomy can allow for the UA to safely continue operations or return to a predefined location.

1.1.2 Compliance with existing aviation regulations

Australia is a unique environment with an overall landmass comparable to that of the United States, but with a fraction of the population. Understandably, en route radar coverage is limited to the more populated coastal regions [32], leaving Air Traffic Control (ATC) with less capability to effectively monitor and prevent potential collisions in areas outside radar coverage (Figure 1.2).

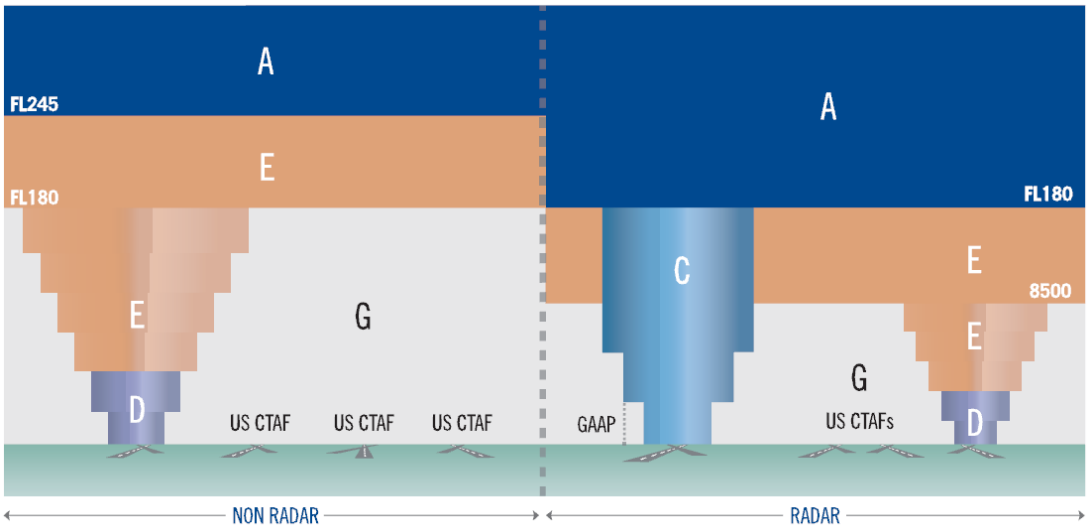


Figure 1.2: Australian civilian NAS (June 2004) [1]

The expected introduction of Automatic Dependent Surveillance Broadcast (ADS-B) [33; 34] and other enabling technologies [35; 36] in Australia's NAS will increase the overall capability of Australia's ATC. This is reflected in Australia's long term NAS strategy most importantly through the overall increase in Class E airspace (Figure 1.3).

At higher altitudes, suitably equipped aircraft generally fly in Instrument Flight Rules (IFR) mode in class A and E airspace; where separation services are provided by ATC. For a UAS operating at higher altitudes in the NAS, the UA must follow all commands directed by ATC to maintain separation between other aircraft in proximity.

Applications such as: traffic surveillance; response to emergency situations; and search and rescue may require aircraft to fly at lower altitudes, where the immediate environment present a hazard to the platform. For aircraft operations outside or below radar coverage such as class G airspace, no separation services are provided by the ATC [1]. Low altitude autonomous UAS operations in the NAS are expected to be more challenging as onboard automation must meet mission objectives whilst ensuring that the UA remains safe from the threat of collision with terrain or other aircraft (if present).

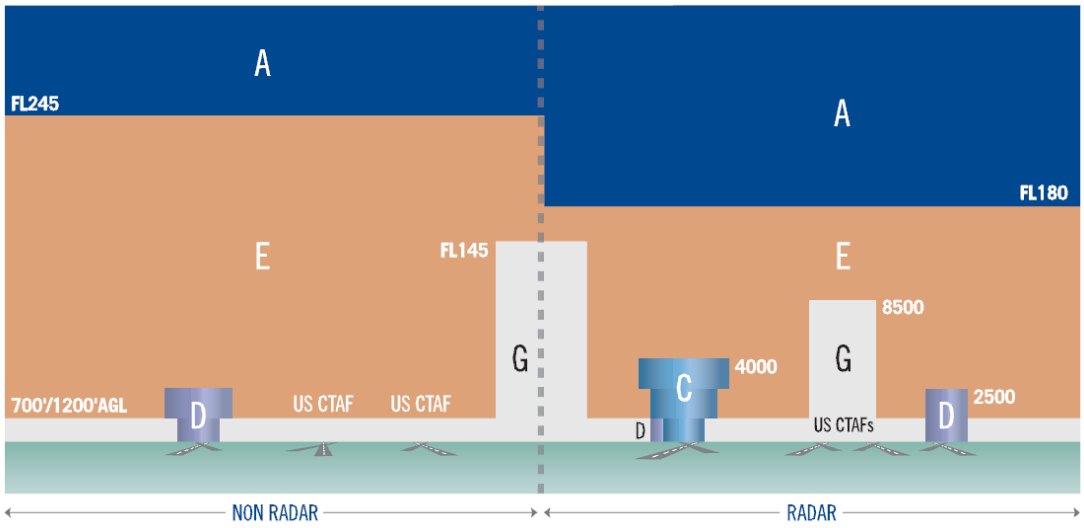


Figure 1.3: Australian NAS long term strategy [1]

1.1.3 Appearing transparent to other airspace users

The Office of the Secretary of Defense details its vision for the integration of military UA into the NAS. Transparency is deemed when no distinction is made between appropriately equipped UAS and manned aircraft by ATC authorities and airspace regulators [37].

Dalamagkidis [18] states that "Dealing with mixed UA/manned aircraft operations will present one of the greatest challenges to the air traffic system." This is a difficult problem which requires careful coordination between ATC, manned aircraft pilots and UAS (either human in the loop or potentially onboard automated systems). Research regarding mixed operations has been undertaken by Boeing Research and Technology in collaboration with the University of Sheffield and the Australian Research Centre for Aerospace Automation (ARCAA) as part of the Smart Skies project [38].

1.2 UAS Autonomy

Bruce [39] states that "Autonomous means that a system has a choice to make free of outside influence". A UA without any onboard autonomy present (e.g.

HIGH	10. The computer decides everything, acts autonomously, ignoring the human.
	9. informs the human only if it, the computer, decides to
	8. informs the human only if asked, or
	7. executes automatically, then necessarily informs the human, and
	6. allows the human a restricted time to veto before automatic execution, or
	5. executes that suggestion if the human approves, or
	4. suggests on alternative
	3. narrows the selection down to a few, or
	2. The compute offers a complete set of decision/action alternatives, or
LOW	1. The computer offers no assistance: human must take all decisions and actions.

Table 1.1: Levels of automation of decision and action selection [15]

a Radio Piloted Vehicle) requires a human in the loop at all times to perform all sensory and decision making tasks required during operations. Whilst a completely autonomous UAS has the freedom and capability to determine and execute missions in the most optimal manner without human intervention.

There are various methods proposed to provide a measure of system autonomy [15; 39; 40]. For example, Sheridan's [15] scale of autonomy (Table 1.1) outlines UAS autonomy requirements on a scale of 1 to 10. Sheridan's scale of autonomy states that autonomy levels up to 5 are suited to decision support where the UAS must have operator approval before performing any operation. Conversely autonomy levels above 8 are suited to onboard stability and control systems where the UAS performs all aspects of the operation and then informs the operator; the operator has no power to veto the decision.

Due to the potential risks of platform failure, UAS operations are continuously monitored by Human Decision Makers (HDMs), where UAS require human guidance to varying degrees and often through several operators [41]. For example military UA such as the predator (General Atomics) and shadow (AAI) require two HDMs to perform navigational and payload control functions during operations. This results in continuous operator workload and places an increased reliance on the communications link connecting the ground station to the UAS platform. One method to decrease operator workload is through increased levels of autonomy onboard UAS.

1.2.1 Increasing onboard autonomy levels

A higher degree of onboard autonomy allows for the replication some of the sensory and decision making capabilities of a human pilot. Most literature indicates that this capability can be realised through the implementation of an intelligent control architecture [42]. Replicating the capabilities of a human pilot is not a trivial task however. For example, during a routine manned flight in civilian airspace, the pilot uses available data (e.g. terrain maps), sensor readings and instructions from ATC to fly the aircraft safely to its destination. The pilot is capable of dealing with varying situations including and not limited to: turbulence, onboard failures (e.g. actuator, sensor, engine), performing a forced landing and avoiding potential collisions with terrain and other aircraft.

To encapsulate the qualities of a human pilot within UAS, the intelligent control architecture must accurately model a pilot's decision making process. An example of aircraft pilots cognitive process [2] during routine flight is shown in (Figure 1.4). The cognitive model is relatively complex but the reader should note that human pilots have their own sensors (e.g. vision, touch) and actuators (e.g. hands, feet). Pilots use their own perception (e.g. recognition of obstacles) in conjunction with memory (prior experiences) to take appropriate actions in a broad range of scenarios.

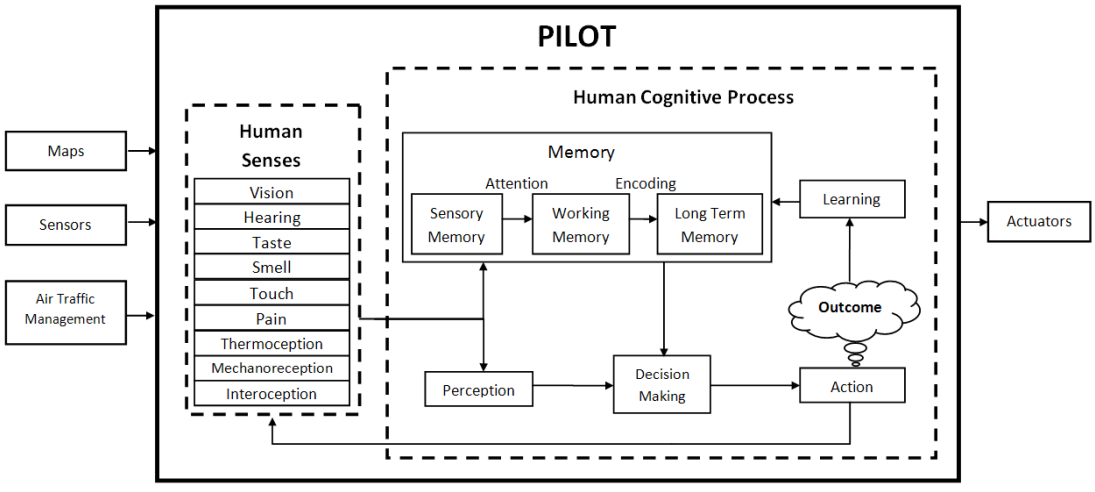


Figure 1.4: Cognitive model of a pilot's decision mMaking process [2]

Chandler [3] decomposes a generic aerial mission into a hierarchy of layers, each representing specific tasks undertaken by pilots or ATC. Figure 1.5 presents a simplified version of Chandler's global planning visualisation. These layers can be automated to provide UAS platforms with the onboard capability to operate with a higher level of autonomy. The following section discusses how the automation of the flight planning and navigation components of Chandler's global planning visualisation may provide UAS with greater capability to operate with greater autonomy in the NAS.

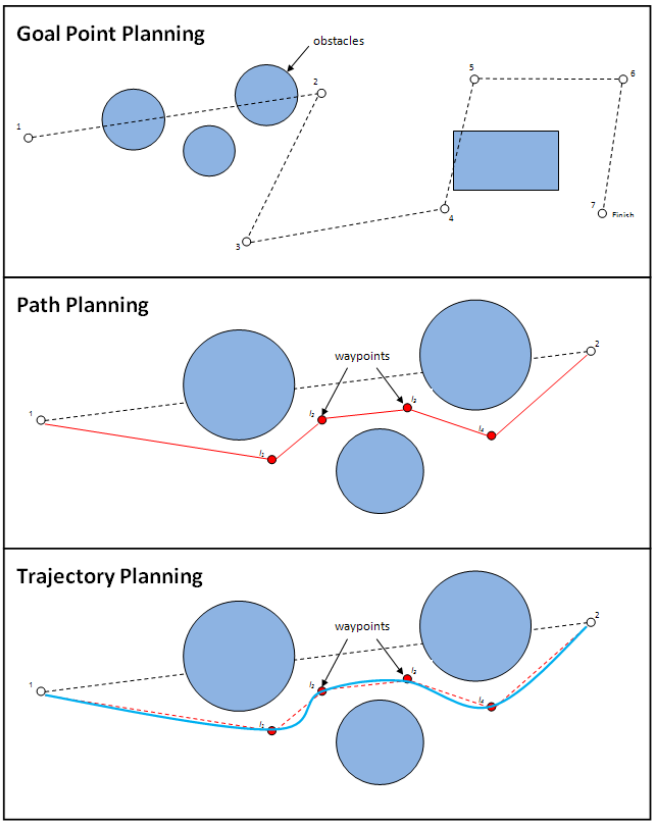


Figure 1.5: Global planning visualisation [3]

1.2.2 Autonomous low altitude UAS operations in the NAS

Automation of UAS flight planning and navigation can be broadly categorised into the following two areas of research; path and trajectory planning. Path planning is the generation of a path (represented as a set of waypoints from an initial position to the goal) to achieve mission success in some optimised manner [43].

The trajectory planner outputs the desired platform track, represented as a continuous collision free flight trajectory. The track represents the traversal through a waypoint set in an optimal manner whilst ensuring that the UAS meets platform performance bounds and non-holinomic constraints (for fixed wing UAS platforms). Trajectory planning algorithms are dependent on either a path planning algorithm or a human in the loop to provide a set of mission level waypoints for traversal (Figure 1.5.

Conducting UAS operations autonomously at lower altitudes in the NAS may present several challenges not encountered during high altitude flight (Section 1.1.2). Terrain and urban structures become hazards to the safety of the UAS operations and must be taken into consideration during flight planning and navigation. Assuming that optimised mission waypoints are available, the inclusion of an automated trajectory planning solution onboard UAS platforms would allow for safe autonomous flight through low altitude environments. Automating the trajectory planning process however, is non-trivial and some challenges include:

- incorporation of complex platform dynamics and the guarantee that trajectories generated are collision free,
- trajectory optimization to meet given mission requirements,
- real-time constraints on computation time imposed by obstacles in the flight path.

These challenges are discussed in further detail in the following sections.

1.2.2.1 Incorporating platform dynamics and collision free guarantees

Generally, there are two types of UA; rotary/helicopter UA that have the ability to brake and hover, and fixed wing UA which exhibit non-holinomic constraints during flight. Rotary UAS generally have shorter flight times while fixed wing UAS often have greater endurance but must always maintain some minimum (greater than stall) velocity. This research project focuses on fixed wing platforms due to their increased velocity and endurance capabilities, and their ability to operate at a greater range of altitudes than their rotary counterparts [26].

Inclusion of platform dynamics is required for a fixed wing UAS allows for the generation of flyable and trackable flight trajectories [13]. Platform dynamics refers to a mathematical model which allows for the inclusion of a platforms aerodynamic constraints such as; minimum and maximum cruising velocity, climb rate, minimum turn radius and attitude rates for fixed wing UA [44; 45].

In order to ensure UA safety during autonomous operations, the platform must remain in a collision free state. The platform must have the capability to enter a hold a manoeuvre during periods of operations when they are required to remain in a stationary location. Scenarios requiring the execution of hold manoeuvres are dependent on platform capability and mission requirements, and may include; awaiting instructions from HDM, conducting surveillance operations, communications loss, or sensor malfunction.

1.2.2.2 Trajectory optimisation

During the course of flight operations; the pilot/UAS operator may consider multiple criteria in order to achieve mission success. Examples of mission criteria may include: achieving mission goal/s; safety of the vehicle; the environment and the public at all times; mission efficiency (minimising time, fuel and/or cost); and/or limiting operations to within a specified altitude ceiling. Mission objectives and their priorities may also dynamically change at any point during UAS operations (usually at the discretion of the operator).

Decision making during autonomous trajectory planning requires the selection of the most optimal feasible collision free trajectory with respect to one or

more criteria. Therefore, the use of Multi-Criteria Decision Aid (MCDA) methodologies during autonomous trajectory planning may allow for convergence to a solution which better reflects overall mission requirements as determined by the candidate HDM. MCDA is a field of research for the development of multi-criteria decision tools to assist HDMs with formulating decisions whilst considering more than one criteria [46].

Franke [4] states that with increasing levels autonomy onboard UA, operators move away from direct control of the platform towards a management by exception control paradigm (Figure 1.6). Management by exception occurs when the UAS performs planning and execution and informs the HDM of its current and future actions. The operator has the option to veto or override the current plans and revert to a lower control paradigm if required (similar to level six of Sheridan's scale of autonomy presented in Table 1.1). Operating at higher autonomy levels requires the HDM to have a sense of trust with the automation, where he/she feels that the automated systems onboard UAS are making correct decisions.

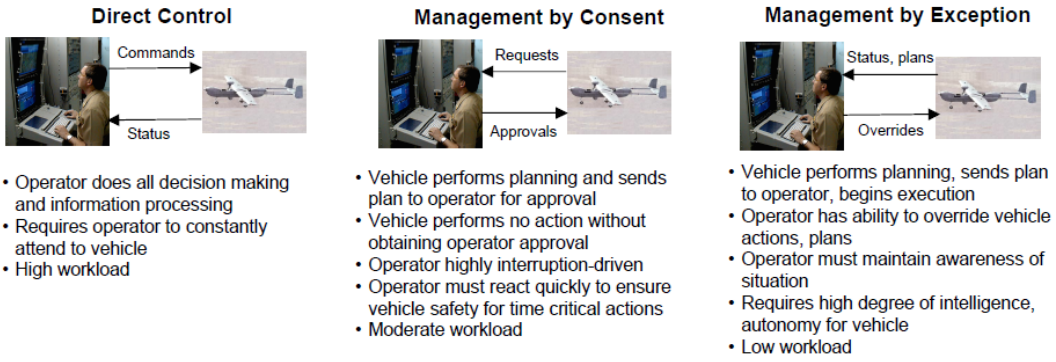


Figure 1.6: Proposed shift in control paradigms with increasing levels of system autonomy [4]

It is important to note, that during the decision making process, the HDM will apply his/her own values, priorities and preferences for a given decision problem [47]. Different human operators may possess varying viewpoints on whether a given solution is acceptable or to be vetoed. Supervising HDM maybe reluctant to allow a UAS which they are supervising to continue operations autonomously if they do not agree with the decisions being made by automated systems onboard.

The analysis of expert decision data gathered from a set of human operators may provide a deeper understanding of objectives considered and the preferences they apply during the decision making process. Incorporating this information into a multi-objective optimization process can potentially allow automated trajectory planners to better encapsulate mission criteria considered by supervising HDMs, and subsequently increase the acceptance of the autonomous solution.

1.2.2.3 Real-time computation constraints on planning

Conducting autonomous UAS operations at low altitude cluttered environments may present several challenges not encountered during high altitude flight. Terrain and urban structures become hazards to the safety of the UAS. Thus, the proximity of obstacles to the UAS may place real-time constraints on the time available to compute valid trajectory solutions.

Assuming that the trajectory planner is deterministic in nature, the time required to compute a solution is dependent on the onboard computational capabilities of the platform and the efficiency of the algorithms applied. Additionally, the time available to the planner to compute a solution is also dependent on platform velocity. Finally, the onboard computational capabilities and flight performance of individual UA can also vary.

For operations in environments where planning must be performed within a limited time frame; the UAS onboard processing capabilities may not be sufficient to generate a feasible optimised collision free trajectory plan before the imposed deadline. Thus, a computationally adaptive trajectory planning solution may allow for the computation of optimised solutions in the presence of real-time deadlines onboard UA with different computational and performance capabilities.

1.3 Research statement

This research investigates the development of a trajectory planning solution to provide UAS with the capability to operate with greater autonomy during low altitude operations in the NAS. Enabling autonomous low altitude UAS opera-

tions can allow for search and rescue, wildlife monitoring, fire fighting and aerial law enforcement assistance missions to be undertaken [26; 27].

1.3.1 Research questions

Most literature states that certain aspects of aircraft operations can be automated through the application of intelligent control architectures. Automated trajectory planning has been identified as potentially providing the onboard capability for autonomous UAS operations in low altitude partially known environments in the NAS. Whilst it is outside the scope of this research to implement and validate an intelligent control architecture, architectures or frameworks can be identified which may allow for the incorporation of trajectory planning as a specific module or layer within the overall architecture. This leads to the first research question:

1. Can trajectory planning be effectively automated for standalone autonomous operations on a UAS platform?

The ability to represent trajectory planning as specific standalone module allows for the development and validation of the trajectory planning aspect independently, whilst still providing the flexibility for integration of other layers (e.g. path planning or trajectory tracking) to form a functional architecture which can be integrated onboard UA to enable autonomous functionality. However, some of the challenges present during the automation of the trajectory planning process include; inclusion of complex platform dynamics, meeting HDM preferences and priorities and the presence of real-time constraints during planning. These challenges are encapsulated in the second research question:

2. Under what conditions can a flight management concept be developed to ensure that the supervisory HDM's mission criteria are successfully met during operations in low altitude environments with real time planning constraints present?

The analysis and incorporation of expert decision data into the decision making component of the trajectory planner, may result in better encapsulation of HDM's mission priorities and preferences during autonomous UAS operations. A

computationally adaptive trajectory planning solution may allow for the computation of optimised solutions in the presence of real-time constraints. Real-time constraints placed on a trajectory planner are dependent on; platform's onboard computational capabilities, efficiency of algorithms applied and platform dynamic capabilities. Thus, both components of this research question do not overlap and can be investigated in parallel as separate components and then integrated together to form the overall flight management concept.

1.3.2 Research Objectives

Based on the research questions posed in the previous section, the research objectives are defined as:

1. Identification of architecture/s which allow for the inclusion of trajectory planning as a standalone module.

2. Development of a methodology for the inclusion of HDM preferences during trajectory planning.

3. Development of a flight management concept to enable autonomous trajectory planning during low altitude operations in the presence of real time constraints.

1.3.3 Research Outcomes

The research outcomes are derived from the research objectives outlined in the previous section. The major outcomes of this research are:

1. A methodology for the capture and inclusion of human preference information, to meet HDM mission requirements during automated trajectory planning.

2. A computationally adaptive flight management concept to enable autonomous trajectory planning, in the presence of real time constraints, onboard

platforms with varying computational capabilities.

In order to verify the proposed methodologies and concepts, test scenarios were developed for validation of this research through simulation. Case studies were developed in simulation to demonstrate the applicability of this research to generate trajectories which allow for safe autonomous UA missions in the NAS, whilst meeting HDM mission requirements, in the presence of real-time constraints.

3. Proposed methodologies and concepts were validated through simulated test scenarios which represent potential low altitude missions in the NAS. This demonstrates the application of this research to autonomous low altitude UAS operations in the NAS.

Additionally, it is important to publish the findings of this research:

4. A number of peer reviewed papers resulting from this research were published in the form of conference proceedings and journal articles which encourages discussion and provides scope for the extension of this research.

1.3.4 Research Methodology

This section presents the methodology applied to meet the research objectives. The methodology can be divided into three sections; review of existing literature, development of proposed methodologies and concepts, and verification and validation of the proposed solutions.

A review of relevant literature is performed in the field of intelligent control architectures and UA trajectory planning (Section 2). The outcomes of this research results in the selection of a suitable standalone trajectory planning framework (Section 2.4) which allows for verification and validation of the proposed flight management concepts detailed in the Research Objectives (Section 1.3.2).

Whilst Manoeuvre Automaton (MA) [48] and safe state [49] theories implicitly guarantee trajectory feasibility and platform safety respectively, the implemented algorithms are validated through the generation of fixed wing trajectories in simulated low altitude cluttered environments. The resulting safe automaton

(Section 3.5) is inspected to ensure that dynamic and kinematic constraints are not exceeded (Section 3.3).

To develop a methodology for the capture and inclusion of human decision information to meet HDM mission requirements, data captured from a candidate human experts is first compared to a reference automated algorithm (LC-2) where preferences remain fixed (Section 4.2.2.1) . HDM preferences are elicited through the application of UTilité Additive (UTA) theory [50] (Section 4.2.2).

It is expected that if automated decisions applying preferences elicited from the candidate human expert data (UTA-4) consistently generates the same or similar decisions to the HDM, then the residual error between HDM decisions and UTA-4 will be less than that between HDM decisions and LC-2 (Section 4.2.3). This is verified by computing the UTA-4 and LC-2 automated solutions and comparing the output trajectories to the HDM trajectories selected for all decision scenarios (see Appendix A for list of decision scenarios completed by each HDM).

These results are validated through the comparison of UTA-4 and LC-2 trajectory solutions computed online in simulated low altitude decision scenarios, to demonstrate that UTA-4 allows for the generation of trajectories which incorporate aspects of the candidate HDM's decision style (Section 4.3.3).

To develop a computationally adaptive flight management concept, trajectory planning was performed in low altitude simulated urban environments using fixed length automatons. Due to variable trajectory segment lengths, real-time planning was only feasible for small automaton sizes using systems with high computational capabilities.

The Computationally Adaptive Trajectory Decision optimization System (CATDS) flight management concept was proposed to dynamically select automaton size, to ensure that, trajectory segments can be computed within the available window of time (Section 5.2). It is expected that by selecting automaton sizes resulting in the computation of each trajectory segment within a finite planning window, the resulting final trajectory can be computed in real-time.

This concept is validated via online trajectory planning in low altitude urban environments, to demonstrate that real-time planning and more efficient use of the planning window is possible through the inclusion of the CATDS. Additionally, simulations are conducted using two test systems with processor speeds, to

demonstrate the algorithms ability to generate solutions, in real-time, onboard platforms with varying computational capabilities (Section 5.2.2).

Finally, both concepts are integrated together to demonstrate the overall flight management concept proposed in the second research question (Section 1.3.1). The overall flight management concept is validated through planning in simulated low altitude partially known urban environments (Section 5.3).

1.3.5 Research Contributions

This section highlights the contributions of this research in the field of autonomous trajectory planning through the application of intelligent control and MCDA methodologies to autonomous UAS systems.

There are two primary contributions of this research, the first contribution is the inclusion of HDM preferences for the selection of flight manoeuvres which better encapsulate HDM mission requirements through UTA theory [50]. This concept can be divided into three components; data capture, preference elicitation and preference selection during decision making.

The algorithm uses a Graphical User Interface (GUI) for data capture where HDMs are tasked with the selection of desired flight manoeuvres, from a given set of alternatives, for unique decision scenarios. The elicitation of human expert decision data representing HDM flying styles into mathematical value functions is performed using UTA theory. HDM preferences are applied during automated selection of the desired flight manoeuvre segments using Multiple Attribute Value Theory (MAVT) [51; 52]. This research allows for the quantification of HDM or pilot decisions and provides a deeper understanding of the decisions considered by a candidate HDM during UAS operations. This demonstrates that the unique decision styles of individual HDMs can be better represented during automated trajectory planning through the inclusion of HDM preferences, via the UTA MCDA technique.

The second contribution is the CATDS flight management concept, which extends on the real-time planning component of Frazzoli's hybrid architecture [53], to allow for real-time replanning in environments with variable planning

deadlines by dynamically varying automaton size. This approach is divided into offline and online components.

The offline component benchmarks the times required to generate and select a safe feasible flight manoeuvres for a range of automaton sizes (alternative set resolutions). The online component selects the appropriate automaton sizes to ensure that a trajectory segment solution can be computed within the finite time available. Additionally, a buffer is applied to provide supervisory HDMs with a predefined period of time to veto decisions if operating in a management by exception control paradigm. This approach provides greater flexibility in providing real-time trajectory planning capabilities on platforms with different computational capabilities whilst allowing for more efficient use of the available time to plan.

Secondary contributions have resulted from extension of Shouwenaars' safe state theory [49] to guarantee platform safety during low altitude UAS flight in the NAS. Manoeuvre Automaton (MA) theory was applied to fixed wing platforms where attitude rate constraints were considered through the generation of manoeuvre primitives. This allowed for the inclusion of attitude rate constraints during the inclusion of implicit safety guarantees, through the application of safe state research in partially known 3D environments.

1.4 Publications

Papers published based on the research presented in this thesis are listed in chronological order below (see Appendix E for full copies of papers where the author is listed as first author).

- **Narayan, Pritesh P.**, Meyer, Patrick, and Campbell, Duncan (2011) "Embedding Human Expert Cognition into Autonomous UAS Trajectory Planning," Submitted for Review. *IEEE Transactions on Systems, Man and Cybernetics Part B: Cybernetics*

- **Narayan, Pritesh P.**, Campbell, Duncan A., and Walker, Rodney A. (2009) "Computationally adaptive multi-objective trajectory optimization

for UAS with variable planning deadlines," In *IEEE Aerospace Conference* 2009, 7-14 March 2009, Big Sky, Montana.

- **Narayan, Pritesh P.**, Wu, Paul P., and Campbell, Duncan A. (2008) "Unmanning UAVs Addressing Challenges in On-Board Planning and Decision Making," In Legras, Francois (Ed.) *First International Conference on Humans Operating Unmanned Systems (HUMOUS)*, 3 - 4 September 2008, Telecom Bretagne, Brest, France.

- **Narayan, Pritesh P.**, Campbell, Duncan A., and Walker, Rodney A. (2008) "Multi-Objective UAS Flight Management in Time Constrained Low Altitude Local Environments," In 46th *AIAA Aerospace Sciences Meeting and Exhibit*, 7 - 10 January 2008, Reno, Nevada, USA.

- **Narayan, Pritesh P.**, Wu, Paul P.Y., Campbell, Duncan A., and Walker, Rodney A. (2007) "An Intelligent Control Architecture for Unmanned Aerial Systems (UAS) in the National Airspace System (NAS)," In *2nd International Unmanned Air Vehicle Systems Conference*, 20th to 21st March, 2007, Grand Hyatt, Melbourne, Australia.

- Wu, Paul P.Y., **Narayan, Pritesh P.**, Campbell, Duncan A., Lees, Michael, and Walker, Rodney A. (2006) "A High Performance Fuzzy Logic Architecture for UAV Decision Making," In *IASTED International Conference on Computational Intelligence*, Nov 20-22, San Francisco.

1.5 Format of thesis

This section details the thesis format and outlines each individual chapter.

In Chapter 2, titled "Literature Review," intelligent control is defined and a literature review of intelligent control architectures is provided. Trajectory planning is defined and trajectory planning layers within UAS architectures are investigated. Literature regarding trajectory planning for mobile robots and UAS is presented and the candidate method applied to this research project for the generation of feasible UAS trajectories is presented.

In Chapter 3, titled "Generating flight trajectories for fixed wing UA using Manoeuvre Automaton (MA) theory," MA theory is defined [48]. Implementation of MA theory in simulation is presented for flight trim conditions commonly executed by fixed wing UA. Platform roll and pitch rate constraint inclusion is demonstrated through manoeuvre primitives. Additionally, the generation of feasible trajectories through concatenation is discussed and the application of collision detection methods to form collision free feasible trajectories is presented. Finally, a review of decision making algorithms to optimise trajectories based on MA theory is presented and the need for MCDA is discussed.

In chapter 4, titled "Embedding Human Expert Cognition into Automated Trajectory Planning," the inclusion of MCDM methodologies and HDM decision data during trajectory planning to better reflect HDM mission priorities is investigated. An overview on the application of MCDM methodologies is presented to demonstrate the inclusion of preferences which reflect HDM priorities. The need to use HDM decision capture methods to formulate mathematical preferences is discussed and the candidate HDM decision capture strategy is presented. Additionally, HDM data capture via a Graphical User Interface (GUI) and the formulation of preferences through UTilité Additive (UTA) theory is demonstrated. Finally, trajectories are generated in simulated low altitude 3D mission environments, to demonstrate that the inclusion of UTA based preferences, can allow for, better representation of the candidate HDM's decision strategies during autonomous UAS operations.

In Chapter 5, titled "Computationally Adaptive Real-Time Trajectory Planning," the safe operation of UAS in low altitude environments in the presence of real time planning requirements is investigated. The requirements for generating MA based flight trajectories in real time is demonstrated by simulating 3D missions in urban terrain using fixed automaton sets on two simulated platforms with different computational capabilities. A flight management system is proposed to dynamically moderate system parameters, to allow for the computation of a trajectory segment solution within a predefined window of time. The results for the inclusion of the Computationally Adaptive Trajectory Decision optimisation System (CATDS) is presented in simulated low altitude 3D urban environments.

In Chapter 6, titled "Conclusions," a summary of findings and discussion of this research project is given. Potential future research and the contributions of the research presented in this thesis are also discussed.

2

Literature Review

Operation of UAS in the NAS requires an equivalent level of safety to that of a human pilot [30]. Achieving higher levels of onboard autonomy helps to address this safety requirement. At the same time, it also reduces the susceptibility to communications failure (less reliance on a remote pilot), lowers the operational costs, and decreases operator or mission commander workload. However, taking the human pilot out of an aircraft removes much sensory and decision making capability. To demonstrate that a UAS still has an equivalent level of safety to a human piloted aircraft, this capability must be automated.

UAS will need to possess greater "intelligence" than they do today, aspiring to acquire the traits of the human pilot. The UAS will need to acquire the capacity to monitor the vehicle's internal systems and the outside world, and to detect any changes that affect the mission safety and mission outcome. With this information, the UAS must then make rational decisions and take the necessary actions to preserve safety and achieve mission objectives. Most literature indicates that this capability can be realised through the implementation of an intelligent control architecture [6].

Intelligent control is a multi disciplinary field that involves the use of techniques from the fields of Artificial Intelligence (AI) and control within the context of the operational requirements of the task (Figure 2.1). Intelligent control systems are generally structured in a hierarchical manner where high level (complex and abstract) tasks are decomposed into a series of time critical low level tasks (data rich and precise). This obeys the so called "principle of increasing precision

with decreasing intelligence" [5]. An overview of intelligent control architectures is presented in the following section.

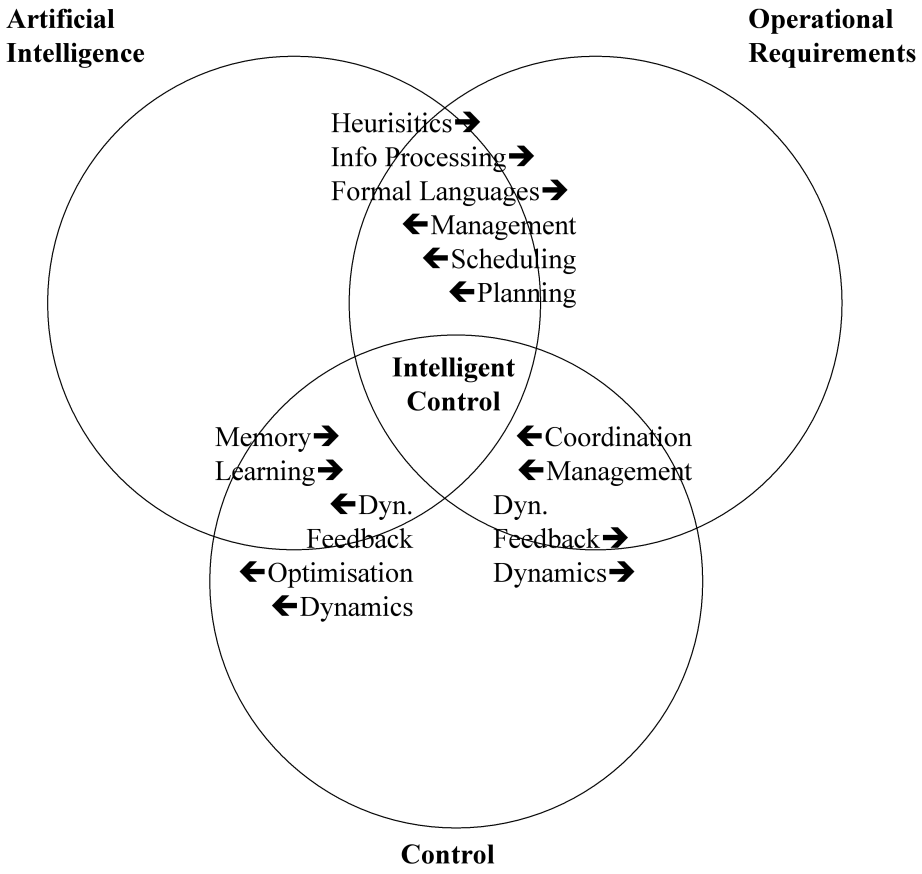


Figure 2.1: Definition of intelligent control discipline [5]

2.1 Intelligent control architectures

From the literature review, it was found that the vast majority of architectures were hierarchical. This approach was often used to separate slower, deliberative planning processes from faster, time-critical hardware control systems [6]. Additionally, it allows for abstraction of complexity from one layer to the next; this is useful not only in reducing subsystem complexity, but also helps in software

reusability [54]. The vast majority of architectures employed some variation of Bonasso's [6] three tiered (3T) hierarchy which has separate layers for deliberation, sequencing of actions and control execution (Figure 2.2).

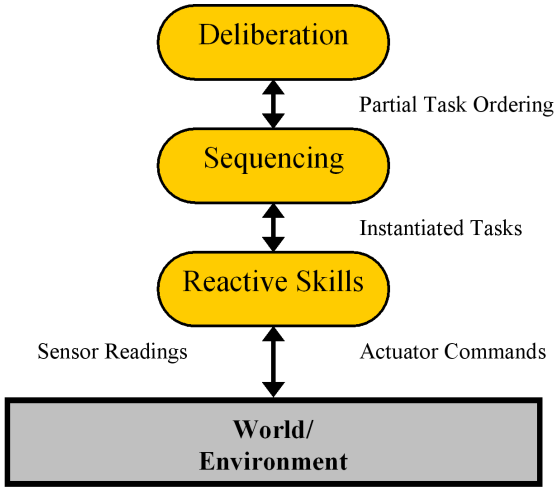


Figure 2.2: 3T intelligent control architecture [6]

Intelligent control architectures implemented onboard UAS are generally extensions of architectures found in robotics. The following section provides an overview of intelligent control architectures implemented onboard UAS platforms.

2.1.1 Review of UAS intelligent control architectures

A UAS platform can be thought of as a mobile robot with the primary directive being to move to a given spatial location within a certain period of time. However, many robotics architectures cannot be directly implemented in UAS.

In comparison to ground based robots, UA operate in highly dynamic environments where atmospheric changes can occur almost instantaneously; therefore, the agent's response must meet real time constraints. This is further compounded by the fact that aircraft typically travel at much greater velocities than ground based robots. Additionally, aircraft dynamics can be highly non-linear and thus require careful consideration during controller design and how this will interface to other subsystems onboard the UA. Finally, failures onboard UA can be catastrophic and result in loss of the platform, property damage, and in the worst

case, loss of human life as these aerial robots are exposed to the general public. Therefore the architecture should be capable of robustly dealing with not only changes in the UAS internal state, but also with changes in the environment.

It was found that in many UAS (and spacecraft based architectures) that an important capability was a method for monitoring the agent's internal state (i.e. the health of the vehicle) and its impact on vehicle performance. This was implemented as a form of Fault Detection and identification/Accommodation (FDA) in Technologies for Reliable Autonomous Control (TRAC), remote agent, Open Control Platform (OCP) and in Boskovic's architectures [7; 20; 55; 56].

Ideally, a human operator should only need to interact with the high level deliberative layer. In this scenario, the operator performs high level tasks such as specifying mission goals and the schedule associated with these goals. In these instances, there is need for a communications subsystem that provides the link between the remote agent and the ground station. Such communications modules are incorporated into the Autonomous Science-craft Experiment (ASE), Architecture for Procedure EXecution (APEX) and Remote agent architectures [54; 55; 57].

Various architectures have been proposed that are specifically targeted at UAS [7; 20; 54; 56; 58]. Schaefer [20] for example, presents a multi-layered UAS decision making architecture known as "Technologies for Reliable Autonomous Control (TRAC)". This is a variation of the 3T architecture pioneered by Bonasso [6] that has been augmented with another layer known as the meta-executive layer. The meta-executive layer is used to coordinate and synchronise interactions between the deliberative (which is goal driven e.g. performing a set of tasks based on accomplishing a particular goal) and execution (which is event driven e.g. performing a set of tasks based on a schedule) layers.

Boskovic [7] presents an architecture (Figure 2.3) which is optimised for UA navigation; the upper layers are arranged in similar fashion to that of the AUtonomous Robot Architecture (AURA) [59]. The layers within this architecture are defined with respect to specific UAS functionalities rather than generic form used for other architectures such as TRAC. The highest layer in this hierarchal four layered model is the Decision Making layer. This layer uses *apriori* information in conjunction with information obtained from sensors to make appropriate

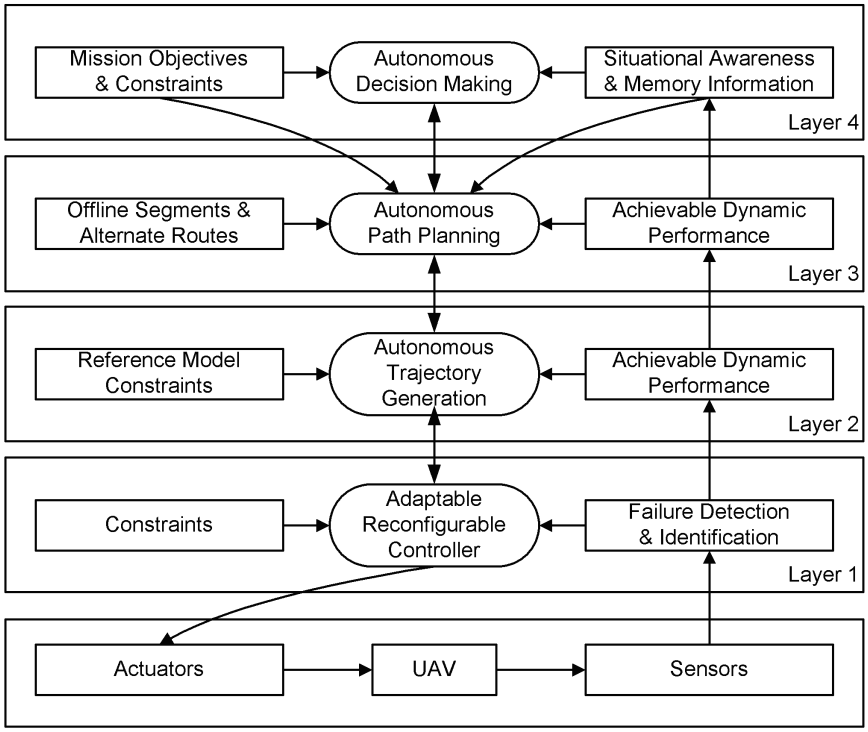


Figure 2.3: Boskovic's UAS decision making architecture [7] - image courtesy of Paul Wu

decisions to achieve mission goals. The next level is the path planning layer which generates an optimised set mission waypoints representing a path from the current location to the goal. The trajectory generation layer generates a smooth flight track which takes into account platform constraints and current achievable dynamic performance.

The desired trajectory is then forwarded to the adaptable reconfigurable controller, which is essentially a set of low level controllers coupled to some form of FDA. The low level controllers are designed to ensure aircraft stability at all times [60] whilst FDA algorithms are used to detect sensor or actuator malfunctions, and accommodated by recalculating achievable performance dynamics to account for any faults identified [61].

2.1.2 Summary of findings

Through an investigation of existing architectures in unmanned aircraft, space based systems and robotics, it was found that various intelligent control architectures have been successfully implemented onboard a multitude platforms to perform navigation and other tasks traditionally only a human operator was capable of performing [42].

As UAS are inherently an airborne mobile robots, intelligent control architectures have been successfully implemented as stand-alone navigational modules denoting aspects of flight planning and execution [7; 62]. For example, automated trajectory planning is treated as a standalone module which receives inputs from the mission path planner (mission waypoints) and the achievable dynamics performance module (platform aerodynamic and kinematic constraints).

The first question that this research project will attempt to answer is whether trajectory planning can be effectively automated for standalone autonomous operations on a UAS platforms. Boskovic's architecture [7] has demonstrated that the trajectory planning layer can be automated and operated as a standalone module if it is being provided with a set of mission waypoints and has knowledge of the achievable performance of the current platform. It is also expected that the trajectory planner will require knowledge of the external environment in order to generate feasible and collision free trajectories.

The following section provides a review of existing research conducted in the field of UAS trajectory planning in cluttered environments. The main objective is to investigate trajectory planning methodologies which allow for UAS to operate with greater autonomy during low altitude operations in the NAS.

2.2 Review of UAS trajectory planning methodologies

A smooth nominal feasible and collision free trajectory is required for safe guidance of the UA from its current position to the desired goal. It should be noted that there is a vast number of trajectory algorithms have been published to date, the majority of which have been applied to the field of robotics [43; 63]. This

section presents an assessment of the suitability of key trajectory algorithms to autonomous low altitude UAS operations.

2.2.1 Spline based trajectory planning

A spline curve (piecewise by polynomials) is a smooth curve which is defined by a set of control points in space. Polynomial or spline based techniques [64; 65] place control points in a particular order to generate the desired trajectory. Spline curves commonly used in trajectory generation algorithms are cubic splines (3rd order polynomial) [66] and Basis splines (B-splines) [67]. Basis splines are commonly chosen since every spline function of a given degree, smoothness and domain partition can be represented as a linear combination of B-splines of that same degree and smoothness [65].

Singh [68] presents a 2D local path planning algorithm using a combination of classical spline based trajectory planning, differential flatness and Model Predictive Control (MPC) techniques. The nominal trajectory is initially generated as a spline containing vehicle coordinates expressed as a polynomial series. If at anytime the generated trajectory falls outside the feasible convex space, an additional waypoint is placed in the middle of the feasible set and the spline based trajectory is recomputed using an MPC based control formulation. A nominal control sequence to track the trajectory is then produced using the assumption that the UAS has differential flatness characteristics. The simulation takes into account a two dimensional (2D) environment which is known *apriori* and the trajectory is calculated offline using an MPC based trajectory generation system. Offline processing is only feasible however, if the environment remains completely static.

Rathbun [69] uses an evolutionary based path and trajectory planning algorithm to converge to a feasible collision free trajectory in a 2D environment with static and dynamic obstacles present. The evolutionary search algorithm initially starts with a finite population of path functions. The population is then randomly placed through a set of genetic mutation operations, with the resulting path segments being evaluated with respect to fitness objectives. This process is iteratively performed until an acceptable solution is obtained.

Nikolos [64] presents an evolutionary based online local path planning system designed for low altitude UAS navigation in three dimensional (3D) environments. The algorithm uses acquired information from onboard sensors to generate a collision free trajectory through unknown environments. The online replanning component of the algorithm has a reduced number of initial path functions in comparison to the offline planner; this decreases the overall replanning time but inversely affects the probability of the algorithm converging to an acceptable solution. The probability of converging to a solution increases as the number of initial path functions increases; this inversely affects processing time. Furthermore, the probabilistic nature of genetic algorithms, makes it difficult to accurately predict the time required to converge to an acceptable solution; if a solution will be found at all.

Koyuncu [70] applies a parameterized modal decomposition to convert the spline trajectory initially formed into a set of concatenated aircraft flight manoeuvres in a 3D known environment. The use of concatenated manoeuvres to form a trajectory is referred to as geometric trajectory planning. The following section provides a literature review of key geometric trajectory planning methods applied to robotics and UAS applications.

2.2.2 Geometric trajectory planning

Geometric trajectory generation methods use a combination of straight line and curved paths to create a smooth optimal pathway through a set of waypoints for vehicles with non-holonomic constraints (e.g. cars or fixed wing UA). Dubins [71] addressed the problem of constructing optimal planar paths (referred to as Dubins paths) to move a ground based vehicle with non-holonomic constraints from an initial location to the goal.

Dubins algorithm [71] creates a 2D trajectory composed of either curves of maximum curvature (C) or straight lines (S), where the optimal solution is shown to be of a bang-bang form (a type of control implementation where the controller begins operations only when a discrete threshold value is exceeded). The geometric construction of the solution consists of at most three segments, either the form CCC or CSC where it is assumed that the vehicle can effectively transition

between path types instantaneously. The algorithm does not take obstacles into account or cost functions other than minimum time [72].

Dubins research has been extended by numerous researchers to generate smooth optimised trajectories for robotics [43; 73] and UAS applications [8; 10; 48; 74; 75].

Anderson [74] has extended Dubin's algorithm [71] to perform geometric trajectory generation for UAS in 2D environments. It is assumed that the path planner has created a collision free path as the algorithm does not take into obstacles into account.

Hwangbo [8] extends Dubins curves [71] to 3D configuration space. The algorithm uses a depth first forward propagating Dynamic Programming (DP) algorithm to select the most desirable manoeuvre from a set of possible manoeuvres for each segment. Hwangbo takes advantage of planning in 3D by generating Dubins paths which allow the aircraft to climb, descend and perform helical manoeuvres (Figure 2.4). However, Dubins algorithms was initially developed for wheeled robots in mind, therefore Dubin's paths may not accurately represent the aerodynamic motion of fixed wing UA.

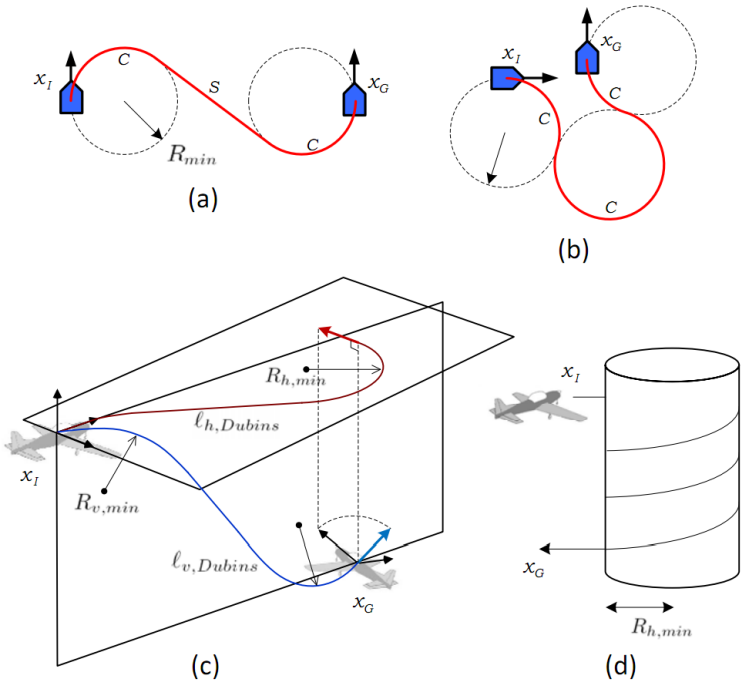


Figure 2.4: Extension of Dubins Paths 3D configuration space [8]

Frazzoli [48; 53; 76; 77] extends Dubins research to form smooth feasible trajectories through the concatenation of motion primitives. Motion primitives represent aircraft manoeuvres which are generated off-line and stored into either of two classes: manoeuvre and trim primitives. Trim primitives represent flight manoeuvres where the platform is in a state of equilibrium (trim state where attitude rates remain zero). Manoeuvre primitives represent flight manoeuvres where the platform transitions to a state outside of equilibrium (attitude rates are non-zero). The generation of feasible trajectories is possible using Manoeuvre Automaton (MA) theory because Frazzoli has mathematically proven [48] that any two manoeuvres can be concatenated together given that there is a trim primitive of finite length to separate the two manoeuvre primitives.

To decrease the computational complexity and resulting time to plan, Frazzoli et al. apply a hybrid architecture to the motion planning problem for rotary aircraft [53; 77]. The hybrid architectures, involve integration of dynamic programming optimisation [43] with other planning methodologies including Rapidly Exploring Random Trees (RRT) [78] (Figure 2.5) and MPC [49]

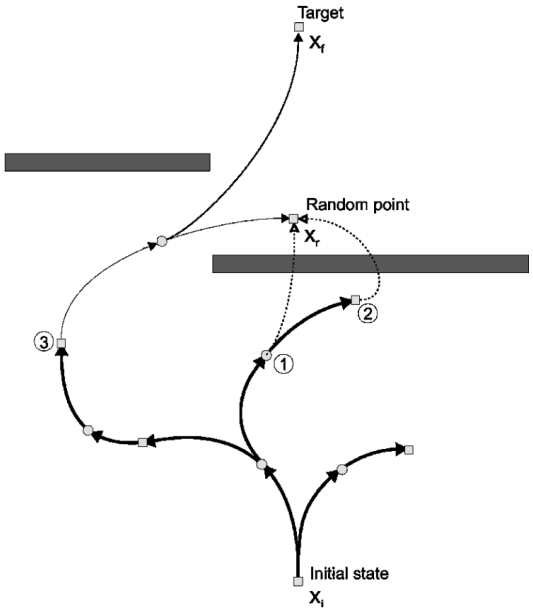


Figure 2.5: Example of RRT expansion during trajectory planning [9]

Schouwenaars [79; 80] presents a 2D MPC based trajectory planning algorithm

with implicit safety guarantees. Firstly, a cost map is generated using Dijkstra's graph search algorithm [81]. Mixed Integer Linear Programming (MILP) toolsets in conjunction with MPC control formulations are then used to generate collision free trajectory segments for each time step. The trajectory is represented as a set of discrete points within a planning horizon (MPC receding horizon strategy). To guarantee platform safety for a fixed wing UA platform, only feasible states where the platform can enter a safe state (i.e. loiter manoeuvre) are considered valid (Figure 2.6).

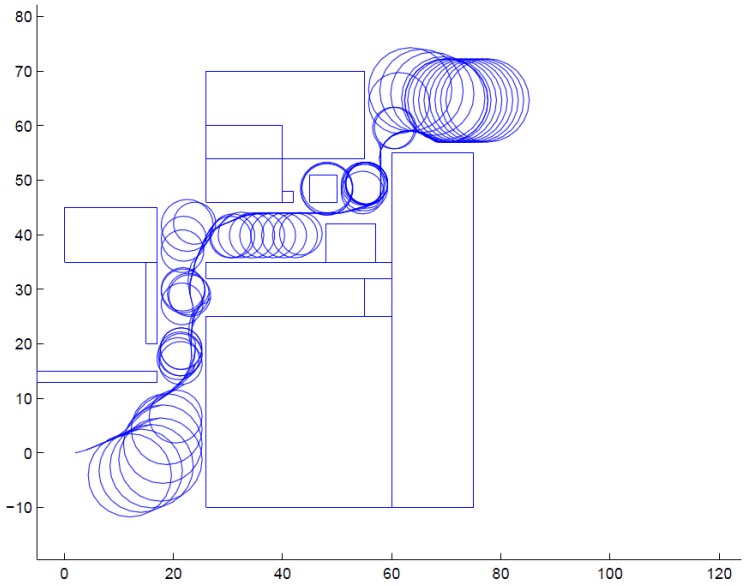


Figure 2.6: Schouwenaars MILP based trajectory planning algorithm with implicit safety guarantees [10]

Kuwata [10] presents a 2D MPC based trajectory planning algorithm which ensures that a solution can be generated in real-time during planning. The cost map and trajectory optimisation using MILP toolsets with MPC control formulations is performed in a similar manner to Schouwenaars' research [79; 80]. To ensure that planning can be performed in real-time, the trajectory executed (execution horizon) is shorter than the trajectory generated (planning horizon) (Figure 2.7). Once the UA reaches the execution horizon, replanning is performed.

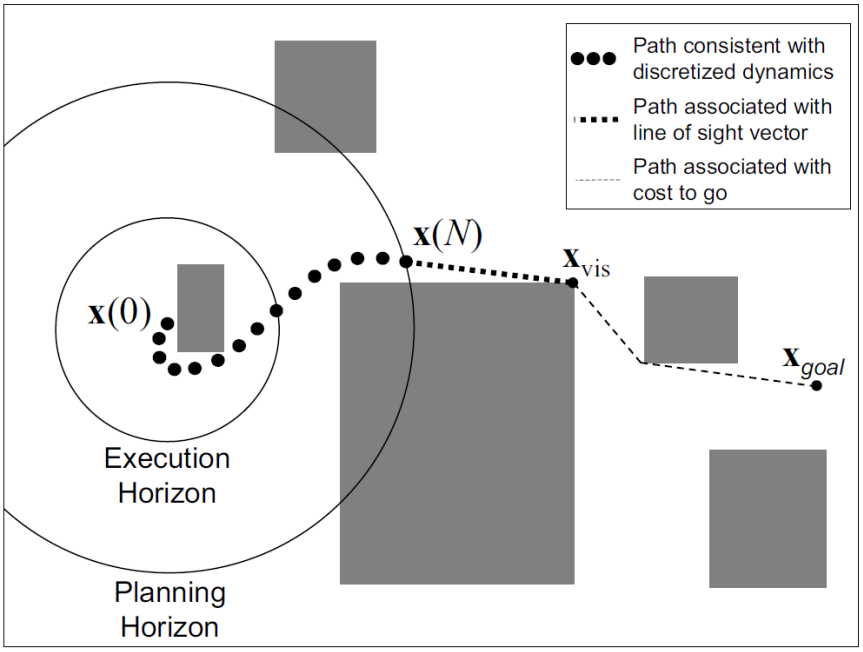


Figure 2.7: Kuwata's MILP based trajectory planning algorithm [10]

2.2.3 Summary of findings

The trajectory planning methods presented can be divided into two areas with respect to the type of environment abstraction applied to the planning problem; known and partially known environments. Known environments refer to trajectory planning in environments where the location of all obstacles is known *apriori*. Partially known environments refers to planning in environments where only a limited knowledge of the surroundings is available.

Planning in known static environments alleviates the issue of planning with real-time constraints as the onboard system can compute a solution offline or compute the full trajectory at the beginning of operations. Practical implementations of onboard trajectory planning in known environments implies the use of precomputed high resolution maps of the entire area of operations [82]. This may not be feasible for some forms of UA (e.g. mini or micro variants) due to cost, computational, or payload limitations. Additionally, maps are required to be routinely updated to reflect any changes to the operational environment. Singh [68], Rathbun [69], Koyuncu [70] and Pettersson et al. [82] apply trajectory planning

algorithms to known environments.

An alternative is to use active or passive onboard sensors to perform online mapping; this is generally referred to as planning in partially known environments. The NAS can be considered as a partially known environment, where onboard sensors are required for the sensing of terrain and/or other aircraft.

2.3 Automated UA operations in partially known environments

Planning in partially known environments limits the onboard knowledge of the operational environment, thus all trajectory planning must be computed online during operations. In environments without complete abstraction, potential field algorithms [83] offer an efficient way to guide a moving vehicle from current location to the goal.

This approach was first developed by Khatib [83] for online collision avoidance for an agent with onboard proximity sensors. An artificial repulsive potential field is applied around obstacles, thereby repulsing the robot away from obstacles. There are special cases where the robot may become stuck in a local minimum if surrounded by obstacles. Techniques such as simulated annealing [84] have been developed to increases the algorithms capability to avoid local minima.

The major issue with potential field algorithms and their suitability to UA guidance in partially known environments is that they are susceptible to becoming trapped in local minima. Scherer, Shim and Griffiths [11; 85; 86] implement additional collision avoidance techniques in conjunction with laplacian algorithms to overcome this issue. Laplacian is a type of potential field algorithm, where the goal is represented as an attractive potential.

Scherer et al. [11] have developed an algorithm which reactively avoids obstacles in 3D space; the overall architecture is separated into planning, reactive and control layers. LAser Detection And Ranging (LADAR) data is processed and composed into an evidence grid; this is a type of configuration space where the obstacle density of regions is represented in logarithmic form (Figure 2.8). As the UAS heads in the direction of the goal, if an obstacle is detected on the

current path, the 3D dodger is initiated to perform reactive collision avoidance. The reactive avoidance module then commits to performing either horizontal or vertical competing manoeuvres.

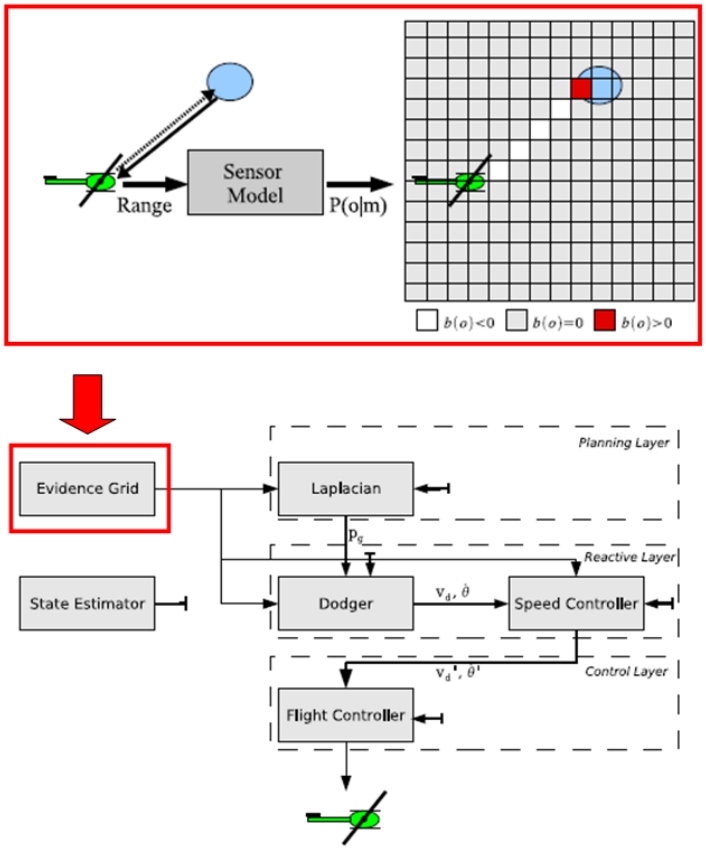


Figure 2.8: Sherer's architecture for reactive collision avoidance [11]

Shim et al. [85] from the BErkeley AeRobot team (BEAR) at the University of California present a conflict-free navigation system for exploration of unknown urban environments using a rotary UA. A 3D local map is generated online using detected obstacle data from an onboard scanning LADAR sensor. An MPC solver attempts to find the optimal control sequence by penalising the proximity of the UAS to the nearest obstacles detected using LADAR over a finite time horizon. Finally, the generated trajectory is executed by the onboard flight management system (responsible for planning and control of the UAS) [87].

Griffiths et al. [86] from the Brigham Young University demonstrate an ob-

stacle and terrain avoidance approach for Miniature Aerial Vehicles (MAVs). An RRT algorithm is used to generate a nominal path through an known urban environment modelled using *a priori* data. During flight, obstacle and terrain avoidance is performed online using data from a static LADAR in conjunction with optic flow sensing cameras. The static LADAR is used to detect if an obstacle is in front of the UAS. Once an obstacle is detected, the UAS performs collision avoidance through the execution of left or right coordinated turn manoeuvres.

Schouwenaars [80; 88] and Kuwata [10] perform planning in partially known environments using a receding horizon strategy to represent a known region. The receding horizon simulates an onboard active sensor which provides the UA with environment abstraction within a finite region (Figure 2.7). Schouwenaars [79] guarantees the safety of the platform by only considering future states which allow the platform to execute collision free hold manoeuvres. Safe states can be represented as hold or loiter manoeuvres for fixed wing UA (Figure 2.6) and hover modes for rotary platforms. A safe state can be initiated as long as the UAS is the minimum distance required from the nearest obstacle to execute a loiter manoeuvre without collision; this is usually the minimum turning radius for fixed wing UAS.

Planning in 2D partially known environments with finite receding horizon strategies can lead to scenarios where the UAS becomes trapped in local minima. Anisi [89] extends the safe state approach with the addition of a 3D safety manoeuvre (essentially a vertically guided spiral manoeuvre). In situations where there is no solution available, executing the safety manoeuvre allows the UAS to gain altitude and recompute a new plan; this ensures the possibility of task completion.

Kuwata [10] and Frazzoli [9] take real-time replanning considerations into account during trajectory planning. Kuwata uses a MILP optimisation methodology to generates trajectories using a dubin's style algorithm [71] for 2D environments with a finite planning horizon implementation. Kuwata ensures that the trajectory planner can converge to a solution within a finite planning window by only partially traversing the planned trajectory segment before the next trajectory segment is computed (Figure 2.7).

Frazzoli [9] applies real-time replanning constraints during the computation of feasible trajectories using a hybrid architecture where DP optimisation is integrated with RRT planning techniques for rotary UA in known environments (Figure 2.5). During online planning, the UA starts traversing the known trajectory whilst the RRT tree composed of feasible primitives are expanded and concatenated together. A finite period of time (τ safe) was introduced to overcome the tendency for randomised algorithms to drive the UA towards a dead end as a result of finite computation times. The planner only selects future primitive segments as feasible if it provides the planner with a minimum amount of time (τ safe) to continue expanding future nodes and is collision free.

2.3.1 Summary of findings

It was found that different methodologies have been applied to trajectory planning and guidance of UA in partially known environments. Sebastian, Shim and Griffiths [11; 85; 86] used some implementation of potential fields to drive the UAS towards the intended destination; alternatively, Griffiths [86] uses an RRT algorithm. As the UAS progresses towards the goal, any detected obstacles within proximity, force the UAS to execute collision avoidance algorithms to ensure platform safety. The use of reactive collision avoidance architectures with limited manoeuvre options may not take complete advantage of the manoeuvrability of the UAS platform. Furthermore, the limited manoeuvre set may not completely encapsulate the range of decisions a HDM is capable of selecting from during Radio Piloted Vehicle (RPV) flight.

Kuwata, Schouwenaars and Frazzoli [9; 10; 88] apply techniques which do not require the inclusion of a reactive collision avoidance module. Kuwata and Schouwenaars [10; 88] apply MILP algorithms to generate trajectories in partially known environments where the known environment is represented as a receding horizon. Frazzoli applies MA theory to discretise platform dynamics and uses a hybrid architecture in conjunction with RRT to generate trajectories which are modelled on rotary flight manoeuvres executed by human pilots. Frazzoli's research [48] allows for the generation of an automaton which contains manoeuvres

typically executed by pilots. This may allow decision making algorithms to select and execute manoeuvres which better represent HDM and mission requirements.

This concludes the review of trajectory planning methods for UA operating in partially known environments. The following section details the selection of the candidate trajectory planning solution which best allows for investigation of the second research question (Section 1.3.1).

2.4 Candidate trajectory planning solution

After reviewing intelligent control architectures and trajectory planning methodology literature, a candidate trajectory planning method was selected which was best suited in allowing for the investigation of the second research question (Section 1.3.1) "Under what conditions can a flight management concept be developed to ensure that the supervisory HDM's mission criteria are successfully met during operations in low altitude environments with real time planning constraints present?".

The incorporation of Manoeuvre Automaton (MA) theory [48] as the underlying component of trajectory planner is expected to allow for autonomous trajectory planning in low altitude environments in the presence of real-time planning constraints. The basis for selection of MA theory as the candidate planning solution is stated in greater detail with respect to the set of autonomous trajectory planning challenges outlined in Section 1.2.2.

2.4.1 Incorporation of platform dynamics

The inclusion of vehicle dynamics during the trajectory planning process, allows for the generation of flight trajectories which take platform constraints into account. Vehicle dynamics are used to calculate the performance envelope which the aircraft must remain within to ensure that the platform does not operate outside performance bounds. In the presence of a Stability Augmentation System (SAS) onboard, the UAS will continue to operate within platform stability boundaries. However, executing trajectories which do not consider platform performance bounds may lead to poor tracking [13].

The actuator control power available on fixed wing platforms is finite; this leads to a non instantaneous period where the vehicle does not remain in a state of equilibrium while the platform transitions between different states of trim. While the platform remains in a state outside equilibrium (trim conditions), attitude rates will be non-zero.

Geometric trajectory planning methodologies applying Dubins curves [10; 74; 90] require the concatenation of aircraft trim flight manoeuvres to form a smooth flight track. However, these flight manoeuvres are usually limited to cruise and constant radius turns trim primitives which are only a small subset of the flight manoeuvres that are capable of being executed by fixed wing aircraft. Furthermore, during the concatenation of Dubins curves, it is assumed that the vehicle can transition between curve segments instantaneously.

During periods when the platform is not in a state of equilibrium, the trajectory planner must account for UAS platform attitude rates as a component of the overall aircraft performance envelope. Inclusion of attitude rate limitations allows for the generation of trajectories which more accurately represent vehicle dynamics, this potentially allows for greater trajectory tracking performance [91]. MA theory [48; 53] allows for the inclusion of attitude rates as a component of overall performance bounds through manoeuvre primitives. Low altitude operations in the presence of terrain could benefit as improved tracking would allow for operations in more cluttered environments as the UA is expected to track the desired trajectory more closely.

2.4.2 Real-time constraints on computation time

Kuwata [10] and Frazzoli [9] have demonstrated that it is possible to take real-time constraints into consideration during planning with geometric trajectory planning techniques. Frazzoli [9] ensures that real-time planning considerations are met by only considering future primitive segments, as feasible, which provide the planner with a minimum amount of time to continue expanding future nodes (τ safe) and are collision free. However, Frazzoli states [53] *"The selection of the trajectory primitives is currently done manually: it would be desirable to obtain formal criteria defining the optimal choice of primitives, trading off the complexity*

of the resulting automaton with the achievable performance. A dynamic resizing of the automaton is also conceivable: in critical situations, when a decision has to be taken in a very short time, the automaton could be reduced to a few maneuvers, whereas in a more secure situation the set of possible maneuvers could be expanded."

During secure situations, the use of a higher set of manoeuvres would allow for planning at higher resolutions, whilst in time constrained scenarios, the automaton could be reduced to ensure that a solution can still be computed within real-time constraints. To the authors knowledge, this research has not been explicitly considered in literature previously.

2.4.3 Trajectory optimisation to meet given mission requirements

MA theory is a technique to discretise the system dynamics rather than state space (as generally present in most graph search planning methods [43]). Therefore, careful selection of automata (trim and manoeuvre primitives sets) can provide a discretised approximation of an aircrafts aerodynamic performance capabilities.

During the course of manned operations, the pilot is responsible for steering the aircraft to achieve mission success whilst taking into account multiple mission criteria. The use of automata which represent common flight manoeuvres provides autonomous onboard trajectory planners with the capability to generate trajectory solutions which emulates the candidate Human Decision Maker's (HDM's) flying styles.

However, the analysis of expert decision data is required in order to have deeper understanding of objectives considered and the preferences they apply during the decision making process. The inclusion of HDM or pilot decision data may allow an automated UAS trajectory planner to better encapsulate human expert decision styles and subsequently increase the acceptance of the autonomous solution [4]. To the authors knowledge, this research has not been explicitly considered in literature previously.

2.5 Summary of Findings

Literature regarding intelligent control architectures and trajectory planning methods was presented in chapter 2. Key trajectory planning methods and their applicability to meeting autonomous low altitude operational challenges were discussed. An overview of the candidate trajectory generation method, MA theory, was presented and its potential for the inclusion of UAS dynamic constraints for accurate platform tracking was highlighted.

Research has previously been presented on the generation of common flight manoeuvres for rotary UA [9; 49; 72; 91; 92; 93; 94]. Fixed wing UAS have a completely different set of dynamics in comparison to rotary platforms, thus a different set of trim and transition manoeuvres must be generated through the application of MA theory. The following chapter presents an overview of the implementation of motion primitives for fixed wing platforms.

3

Flight Trajectories for Fixed Wing UA using Manoeuvre Automaton (MA) Theory

The inclusion of vehicle dynamics during the trajectory planning process, allows for the generation of flight trajectories which take platform constraints into account. Vehicle dynamics are used to calculate the performance envelope which the aircraft must remain within to ensure that the platform does not operate outside performance bounds. Manoeuvre Automaton (MA) theory allows for implicit inclusion of platform dynamics within the automaton, thus any combination manoeuvres from the automaton can be concatenated together to form a feasible trajectory. The following section presents an overview of the implementation of motion primitives for fixed wing platforms.

3.1 Manoeuvre automaton theory

Manoeuvre Automaton (MA) theory, proposed by Frazzoli et al. [48; 53] can be used in the generation of feasible flight trajectories through the sequential concatenation of predefined motion primitives. MA employs two types of primitives; trims and manoeuvres. Trim primitives represent the vehicle during a state of equilibrium whilst manoeuvre primitives characterise the vehicle operating outside of equilibrium. Primitives are generated using a dynamic model of the

vehicle, thus platform stability can be implicitly guaranteed through generation of primitives which ensure that the vehicle remains within performance bounds.

For this research project, MA theory is used to describe a time-invariant non-linear, dynamical system , described as a set of Ordinary Differential Equations (ODE) as:

$$\dot{x} := \frac{d}{dt}x(t) = f(x(t), u(t)) \tag{3.1}$$

where u is the control input $(\tau, primitive)$ = (execution time, manoeuvre type) and x is the state vector $([x, y, z], [\phi, \theta, \psi], [(\dot{\phi}), (\dot{\theta})])$ = ([position], [attitude], [attitude rate]).

3.1.1 Trim primitives

Trim primitives represent the UAS platform operating in a state of equilibrium. Using MA theory, trim primitives can be generated by placing the body fixed roll $(\dot{\phi})$ and pitch $(\dot{\theta})$ rates to zero and maintaining a Constant (C) velocity (V), roll (ϕ) and pitch (θ) angle for the duration (τ_q) of the primitive execution.

Trim primitives were generated using a six Degree Of Freedom (DOF) flight dynamics model based on the Aerosonde UAS platform data set available in the aerosim blockset.The aerosim blockset is executed using simulink, which is a component of the MATrix LABoratory (MATLAB) programming environment. Six predefined trim primitives have been implemented in simulation to characterise fixed wing UA during a state of equilibrium including: cruise, coordinated turn, climb, descent, helical climb and helical descent.

The initial platform state $(x(t_i) = x_i)$ reaches a final state $(x(t_f) = x_f)$ due to the execution of a given trim primitive (q); this can be represented as:

$$\begin{aligned} x_f &= x_i + \tau_q \dot{x_q} \\ t_f &= t_i + \tau_q \end{aligned} \tag{3.2}$$

where $\{V, \phi, \theta\} = \{\mathrm{C}, \mathrm{C}, \mathrm{C}\}$ are constant and $\{\dot{\phi}, \dot{\theta}\} = \{0, 0\}$.

It is of importance to note, that for a platform to enter a state of equilibrium (execution of a trim primitive), the initial platform attitude must equal the attitude requirements of the trim primitive to be executed; $\{\phi, \theta\}_i = \{\phi, \theta\}_q$. If the

initial platform attitude does not equal the attitude required to execute the given trim primitive, a manoeuvre primitive must be inserted to ensure that body fixed attitude rate constraints are taken into account.

3.1.2 Manoeuvre primitives

During the execution of a manoeuvre primitive, the UAS does not have to remain in a state of equilibrium. For a fixed wing platform, the body fixed attitude rate constraints become $\{\dot{\phi}, \dot{\theta}\} = \{\dot{\phi}_{max}, \dot{\theta}_{max}\}$. In this research, manoeuvre primitives (p) are employed to connect two trim primitives, if required, in the formation of feasible trajectories. This allows for the consideration of attitude rates as an additional platform constraint during periods where the UAS is not in a state of equilibrium. If $\{\phi, \theta\}_i \neq \{\phi, \theta\}_q$, the UAS platform dynamic model is propagated until the platform reaches the desired state configuration, ensuring that platform attitude rate considered are considered during trajectory generation.

While

$$\begin{aligned} \{\phi, \theta\}_k &\neq \{\phi, \theta\}_q \\ x_{k+1} &= x_k + \dot{x_p}\Delta T \\ t_{k+1} &= t_k + \Delta t \end{aligned} \tag{3.3}$$

where $\{\dot{\phi}, \dot{\theta}\} = \{\dot{\phi}_{max}, \dot{\theta}_{max}\}$.

3.2 Trim primitive formulation

The trim primitives implemented, represent common fixed wing aircraft manoeuvres in which the platform remains in a state of equilibrium. The trim primitives implemented include: cruise (straight and level flight), coordinated turn, constant climb, constant descend, helical climb and helical descent. This section outlines the formulation and implementation of the trim primitive set.

3.2.1 Straight and level flight (cruise)

Straight and Level (SL) flight (Figure 3.1) is commonly performed by the UAS platform where the aircraft maintains a Constant (C) velocity; heading and altitude for a finite period of time. SL flight can be represented for a finite period of time (t) in the Body Fixed (BF) coordinate system where $t \in [t_i, t_f]$ as:

$$\begin{aligned} V &\in [V_{min}, V_{max}] \\ [\theta_{SL}(t), \phi_{SL}(t), \psi_{SL}(t)]_{BF} &= [\mathrm{C}, 0, \mathrm{C}] \\ [\dot{\theta}_{SL}(t), \dot{\phi}_{SL}(t), \dot{\psi}_{SL}(t)]_{BF} &= [0, 0, 0] \end{aligned} \tag{3.4}$$

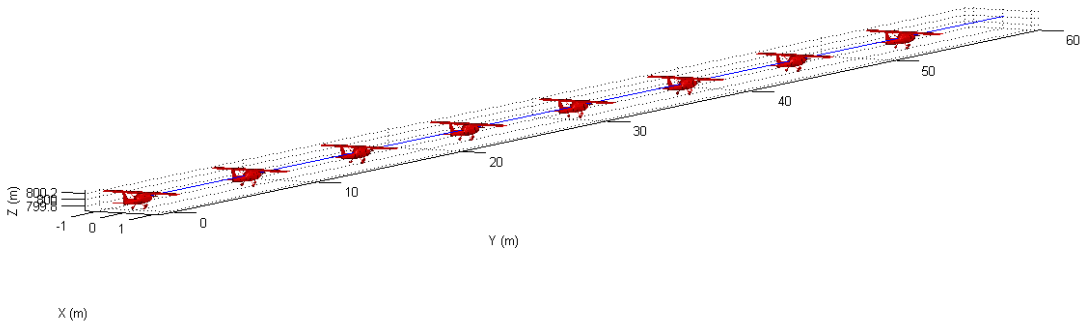


Figure 3.1: SL trim primitive example

3.2.2 Coordinated turn

A Coordinated Turn (CT) (Figure 3.2) is a fixed wing manoeuvre where the platform performs a turn without side-slip through the activation of onboard control surfaces. The aircraft maintains a constant bank angle (ϕ) throughout the duration of the manoeuvre. This manoeuvre has been previously captured in geometric trajectory planning techniques [10; 71]. CT flight can be represented for finite duration where $t = [t_i, t_f]$ as:

$$\begin{aligned} [\theta_{SL}(t), \phi_{SL}(t)]_{BF} &= [\mathrm{C}, \mathrm{C}] \\ [\dot{\theta}_{SL}(t), \dot{\phi}_{SL}(t), \dot{\psi}_{SL}(t)]_{BF} &= [0, 0, \mathrm{C}] \end{aligned} \tag{3.5}$$

The maximum bank angle (ϕ_{max}) is dependent on the wing loading factor (n) which can be placed on the platform. The Federal Aviation Administration (FAA) [95] lists the typical maximum value of n as 3.8G, this research project uses an more conservative arbitrary value of n as 2.2G. Only CT trim primitives which do not exceed ϕ_{max} are generated to ensure platform integrity. It is important to note that as the UAS platform executes CL trim primitives which approach ϕ_{max}, the risk of platform instability due to factors such as onboard controller performance and external factors (e.g. wind) increase (3.6). CT flight constraints applied are presented in Formula (3.6) and include n and the minimum turn Radius (R_{min}) of the platform [96].

$$\begin{aligned} n &= \sec(\phi) \\ R_{min} &= \left(\frac{u^2}{\tan((|\phi_{max}|)\times g}\right) \end{aligned} \tag{3.6}$$

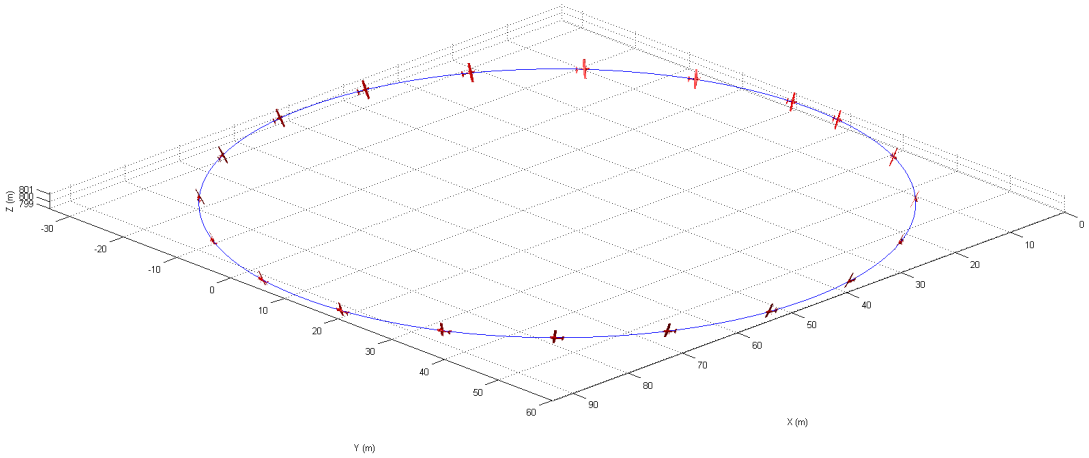


Figure 3.2: CT trim primitive example

3.2.3 Constant climb

Climbing manoeuvres are generally performed for the UAS platform to reach a waypoint which is at a higher altitude then the current vehicle location or during take-off. To perform a Constant Climb (CC) manoeuvre (Figure 3.3),

the platform maintains a constant climb rate (Vc) for the duration of the trim executed.

The maximum climb rate (Vc_{max}) allows for the calculation of the aircraft pitch angle ($\theta_{c(max)}$) which gives the greatest altitude gain in the shortest horizontal distance (3.7). The maximum climb rate (Vc_{max}) is calculated using the excess engine power available (EP_{max}) [96]. EP_{max} is formulated by subtracting the drag (D) experienced by the UA from maximum platform thrust (T_{max}) which is then multiplied by the current platform forward velocity (u_0). T_{max} is dependant on the type of engine present on the UA.

For a single propellor model such as the aerosonde UA, the mathematical approximation for T_{max} is a function of air density (ρ), propellor radius (R_{prop}), propellor rotation speed (Ω_{max}) and the coefficient of maximum thrust and power (CTJ_{max}). Formula (3.7) presents the constraints applied to model CC flight trajectories.

$$\begin{aligned} \theta_{c(max)} &= \sin^{-1}\left(\frac{Vc_{max}}{V}\right) \\ Vc_{max} &= \frac{EP_{max}}{W} \\ EP_{max} &= (T_{max} - D)u_0 \\ T_{max} &= \frac{4\rho {R_{prop}}^4 \Omega_{max}^2 CTJ_{max}}{\pi^2} \end{aligned} \tag{3.7}$$

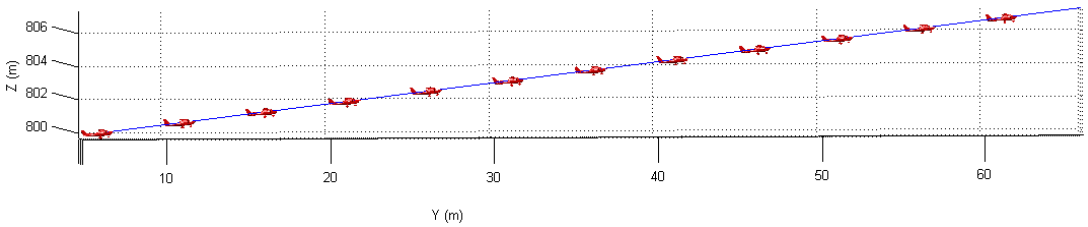


Figure 3.3: CC trim primitive example

3.2.4 Controlled descent

Descent manoeuvres are generally performed by the UAS platform to reach a waypoint which is at a lower altitude then the current vehicle location or during

the landing phase of the mission. To perform a Controlled Descent(CD) manoeuvre (Figure 3.4), the platform maintains a constant descent rate for the duration of the trim primitive execution. Typical descent rates for aircraft operating in the NAS range between 500 to 1500 feet/min [97]. Based on this information, a value of 5 m/s (1000 feet/min) was selected as the maximum platform descent rate (Vd_{max}) for this research project.

Vd_{max} allows for the calculation of the UA descent angle (θ_{dmax}) which provides the steepest angle of descent. During descent, the lift generated (L) is less than the aircraft weight (W). For a constant V, this results in a reduction of the platform angle of attack (α) in comparison to SL flight.

Platform α during CD flight is calculated using the non-dimensional coefficients for subsonic flight including: maximum lift ($(C_L)_{max}$), lift at $\alpha = 0$ ($(C_L)_0$) and derivative of lift with respect to α ($(C_L)_a$) [96]. $(C_L)_{max}$ is computed by dividing L (for CD flight) by the dynamic pressure (Q) experienced by the UA. Formula (3.8) presents the constraints applied during the CD flight mode.

$$\begin{aligned} \theta_{dmax} &= \tan(\tfrac{Vd_{max}}{u_0}) \\ L &= W\cos(\theta_d) \\ C_{Lmax} &= \tfrac{L}{Q} \\ \alpha &= \left(\tfrac{(C_L)_{max}-(C_L)_0}{(C_L)_a}\right) \end{aligned} \tag{3.8}$$

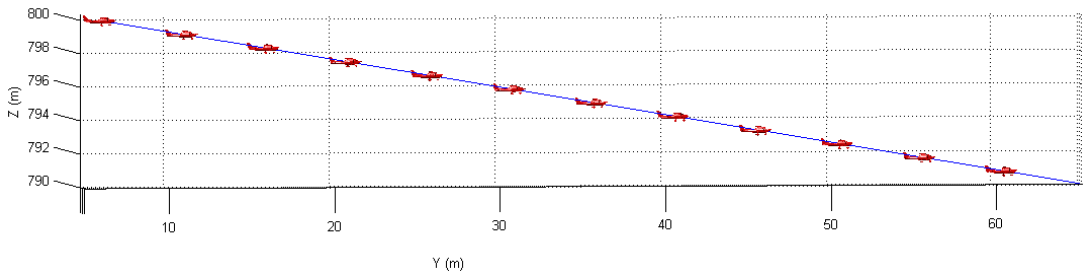


Figure 3.4: CD trim primitive example

3.2.5 Helical ascent

Helical Ascent (HA) occurs when the fixed wing platform performs a constant climb manoeuvre with a fixed banked angle ($|\phi_{HA}| > 0$). HA manoeuvres allow fixed wing platforms to reach waypoints at higher altitudes in a more efficient manner than possible with just the execution of CC manoeuvres alone. Additionally, HA manoeuvres (Figure 3.5) have been shown to be useful in the escape of local minima in 3D partially known environments [89].

The minimum turn radius ($R_{(min)_{HA}}$) equations incorporate maximum angle of climb ($\theta_{c(max)}$) to take into account climbing constraints. Formula set (3.9) outlines the constraints applied during the formulation HA manoeuvres [96].

$$\begin{aligned} u_{HA} &= \frac{u_0}{\cos(\theta)} \\ R_{(min)_{HA}} &= \left(\frac{{u_{HA}}^2}{g\tan(abs(\phi))\cos(\theta_{c(max)})}\right) \end{aligned} \tag{3.9}$$

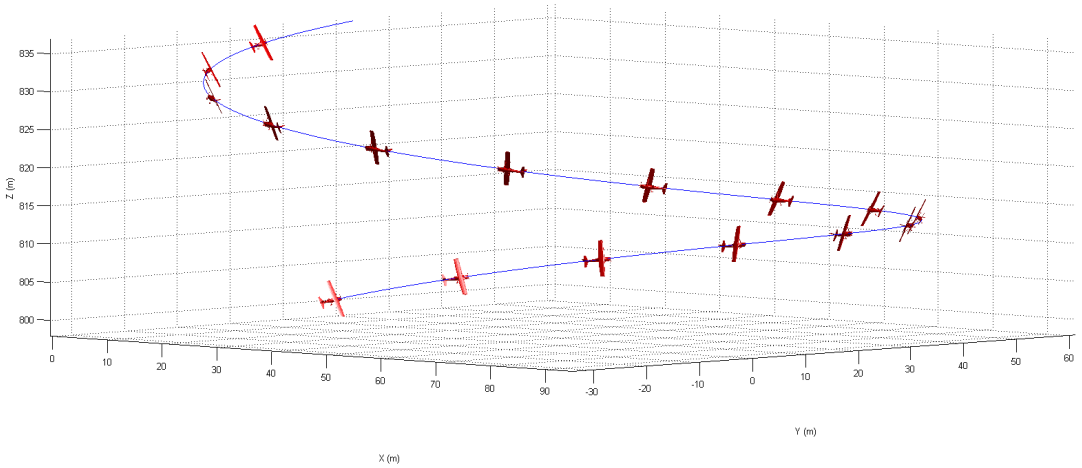


Figure 3.5: HA trim primitive example

3.2.6 Helical descent

Helical Descent (HD) takes place when the fixed wing platform performs a descent manoeuvre with a fixed banked angle ($|\phi_{HA}| > 0$). HD manoeuvres (Figure 3.6) allow fixed wing platforms to reach waypoints at lower altitudes in a more efficient

manner than possible with just the execution of CD manoeuvres alone. HD manoeuvres can also be used during the landing phase of flight.

Due to the introduction of a non-zero bank angle, the vehicle turn radius will be different to that of a CD. In a similar manner to HA, the minimum HD turn radius ($R_{(min)_{HD}}$) equations incorporate maximum angle of descent ($\theta_{d(max)}$) to take into account descent constraints. Formula set (3.10) outlines HD flight constraints formulation applied [96].

$$\begin{aligned} u_{HD} &= \frac{u_0}{\cos(\theta_d)} \\ R_{(min)_{HD}} &= \left(\frac{{u_{HD}}^2}{g\tan(|\phi|)\cos(\theta_{d(max)})}\right) \end{aligned} \tag{3.10}$$

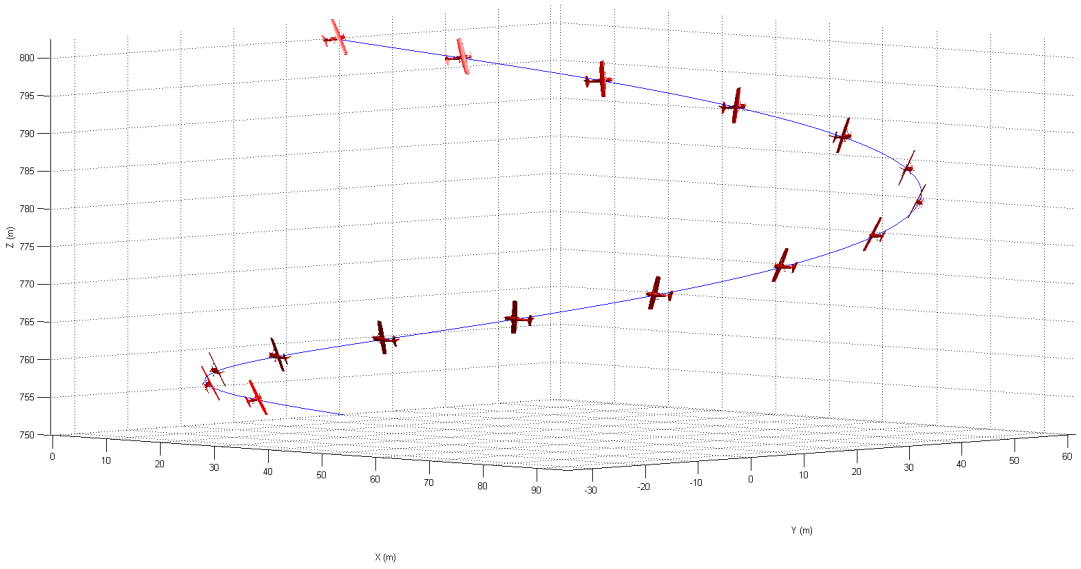


Figure 3.6: HD trim primitive example

This concludes the implementation details of a predefined set of trim primitives for fixed wing aircraft. The trim primitives implicitly guarantee vehicle stability whilst the platform is in a state of equilibrium. Switching between trim primitives may result in the vehicle moving outside the state of equilibrium.

Previous literature has represented switching between piecewise linear trajectory components [10; 71; 74] as instantaneous. Fixed wing UAS platforms

have attitude rate bounds, therefore switching between trim primitives will not be instantaneous if the difference between the current trim primitive attitude and the next trim primitive attitude is non-zero. To ensure that the platform remains within performance bounds, manoeuvre primitives are inserted between the current and following trim primitives. The following section details the implementation of manoeuvre primitives.

3.3 Manoeuvre primitive implementation

Manoeuvre primitives are a representation of aircraft flight while the platform is operating outside the state of equilibrium. Due to finite control power, fixed wing aircraft cannot instantaneously transition from its current attitude to a commanded attitude. The resulting attitude rate bounds are dependent on aircraft performance and physical parameters. Thus, the inclusion of fixed wing aircraft attitude rate constraints allows the generation of flight trajectories which model fixed wing UA flight with better accuracy. Subsequently, the inclusion of attitude rate constraints can potentially lead to the tracking of trajectories with greater precision [91].

This section is separated into three sections to outline the attitude rate formulations during pure rolling motion, pure pitching motion, and when both aircraft rolling and pitching motions are coupled.

3.3.1 Roll rate constraints during pure rolling motion

Most geometric trajectory planning methods [43; 71] assume instantaneous attitude changes between manoeuvres. Figure (Figure 3.7) demonstrates the concatenation of two CT trim primitives assuming that the attitude change between trims is instantaneous.

Inclusion of roll rate constraints allows for the consideration of aileron actuator control power effects during the execution of CT manoeuvres (Section 3.2.2). The fixed wing platform roll rate constraint can be modelled as a first order response (3.11) where the roll rate ($\dot{\phi}$) eventually reaches a steady state roll rate value (P_{ss}) [98]. The parameter (τ_p) represents the time constant of the first order system

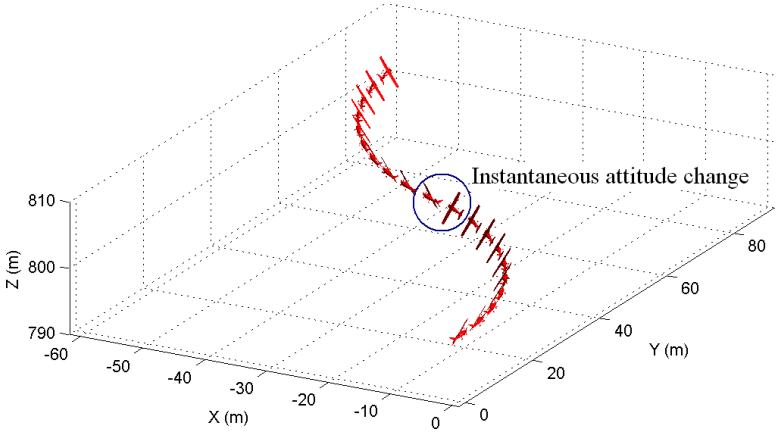


Figure 3.7: Concatenation of two CT trim primitives without inclusion of platform attitude rate constraints

modelling the platform roll rate. Figure 3.8 presents the roll rate constraint modelled for the aerosonde UAS platform operating at a constant velocity of $30m/s$.

$$\dot{\phi} = P_{ss}\left[1 - e^{\left(\frac{-t}{\tau_p}\right)}\right] \tag{3.11}$$

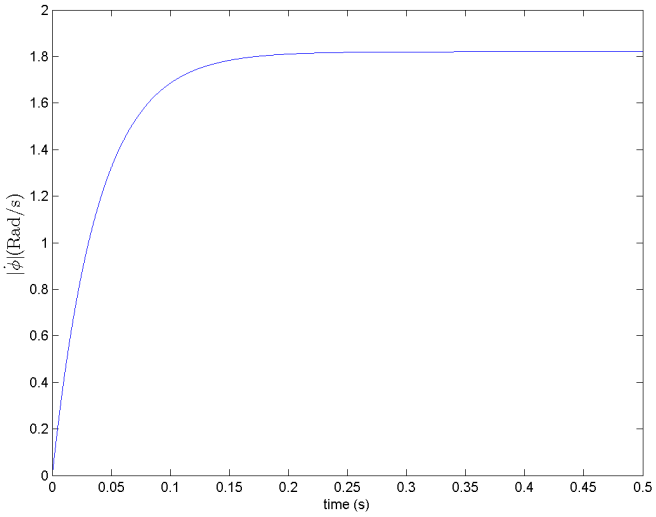


Figure 3.8: $\dot{\phi}$ modelled as a first order response for aerosonde UAS platform travelling at $V = 30m/s$

P_{ss} is a function of the platform roll control derivative coefficient (C_{lda}), roll rate derivative coefficient (C_{lp}), platform forward speed (u_0), platform wingspan (b) and aileron deflection (δ_a). An arbitrary value of $\frac{\pi}{12}$ is used for δ_a. The time constant (τ_p) is inversely proportional to the aerodynamic rolling moment (L_p). L_p is calculated using C_{lp}, b, u_0, Q, wing surface area (S) and the rolling moment of inertia (I_x). The steady state roll rate formulations (3.12) are available in [98]:

$$\begin{aligned} Pss &= -\frac{C_{l_{\delta a}}}{C_{l_p}}\frac{2u_0}{b}\delta_a \\ \tau_p &= -\frac{1}{L_p} \\ &\text{where} \\ L_p &= \frac{C_{l_p}(\frac{b}{2u_0})QSb}{I_x} \end{aligned} \tag{3.12}$$

During transition between trim primitives, $\dot{\phi}$ reaches and maintains steady state roll rate (P_{ss}) until the platform reaches the desired ϕ value of the second trim primitive (Figure 3.9). Figure 3.10 provides a visual representation of the trajectory generated with the inclusion of roll rate constraints during the concatenation of two coordinate turn manoeuvres.

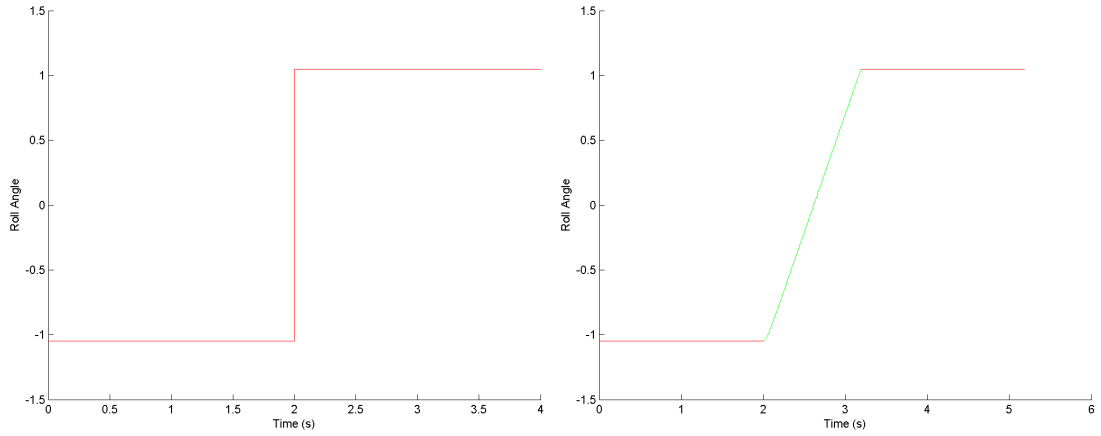


Figure 3.9: ϕ during concatenation of CT trim primitives without and with inclusion of transition manoeuvre respectively

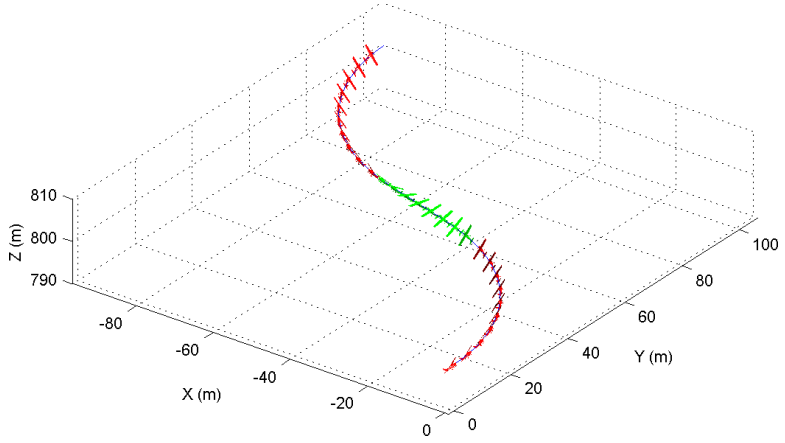


Figure 3.10: Concatenation of two CT trim primitives with inclusion of attitude rate constraints

3.3.2 Pitch rate constraints during pure pitching motion

Pure pitching motion is present during the execution of CC and CD trim primitives. Inclusion of pitch rate constraints allows for the consideration of elevator actuator control power effects during the execution of CC and CD manoeuvres (Section 3.2.2). The parameter (τ_q) represents the time constant of the first order system modelling platform pitch rate (Figure 3.13).

$$\dot{\phi} = Q_{ss}\left[1 - e^{\left(\frac{-t}{\tau_q}\right)}\right] \tag{3.13}$$

The steady state pitch rate (Q_{ss}) was derived using longitudinal pure pitching motion equations [98]. Q_{ss} is found by taking the limit of t approaching infinity where $e^{\left(\frac{-t}{\tau_q}\right)}$ is essentially zero. The formulation applies the assumption that no external disturbances such as wind are present ($\alpha = \theta$), and assumes that α is small ($\alpha \approx 0$) to further simplification. Formula (3.14) presents the Q_{ss} derivation where M_q and $M_{\dot{\alpha}}$ represent aerodynamic moment pitch stability derivatives. $M_{\delta e}$ represents the moment exerted by the platform elevator and $\Delta\delta_e$ is the elevator

deflection angle.

$$\begin{aligned} \Delta\ddot{\alpha} - (M_q + M_{\dot{\alpha}})\Delta\dot{\alpha} - M_\alpha\Delta\alpha &= M_{\delta e}\Delta\delta_e \\ \Delta\dot{q} - 2M_q\Delta q &= M_{\delta e}\Delta\delta_e \\ \lim_{t\to+\infty} Q_{ss} &= \tfrac{M_{\delta e}\Delta\delta_e}{2M_q} \end{aligned} \tag{3.14}$$

Q_{ss} can be calculated after converting M_q and $M_{\delta e}$ to non-dimensional coefficient form. The steady state pitch rate equation using non-dimensional coefficients is presented (3.15), where $C_{m\delta e}$ refers to the pitch control derivative coefficient, C_{mq} refers to the pitch rate derivative coefficient and $\overline{c}$ represents the mean aerodynamic chord length of the wing. An arbitrary value of $\frac{\pi}{12}$ is used for δ_e.

$$\begin{aligned} Q_{ss} &= -\tfrac{C_{m\delta e}}{C_{mq}}\tfrac{2u_0}{\overline{c}}\delta_e \\ \tau_q &= -\tfrac{1}{2M_q} \\ &\text{where} \\ M_q &= \tfrac{C_{mq}(\frac{\overline{c}}{2u_0})QS\overline{c}}{I_y} \end{aligned} \tag{3.15}$$

The limited maximum pitching angle (θ_{max}) generated by the aerosonde UA at $30m/s$ during a steady state climb manoeuvre results in a shorter manoeuvre primitive required to transition between SL and CC flight modes. Figure 3.11 presents the platform θ during the transition between SL and CC trim flight modes, without and with the inclusion of pitch rate constraints respectively. Please note that the x axis of Figure 3.11 has been limited to between 1.8 and 2.2 seconds in order to highlight the manoeuvre primitive execution.

3.3.3 Roll and pitch rate constraints during helical manoeuvres

During transition to a helical manoeuvre, $\dot{\theta}$ and $\dot{\phi}$ constraints may both be present. Adjusted steady state roll P_{ssA} and pitch rates Q_{ssA} are used to ensure that the UA reaches the correct state (θ_{k+1}, ϕ_{k+1}) without exceeding attitude

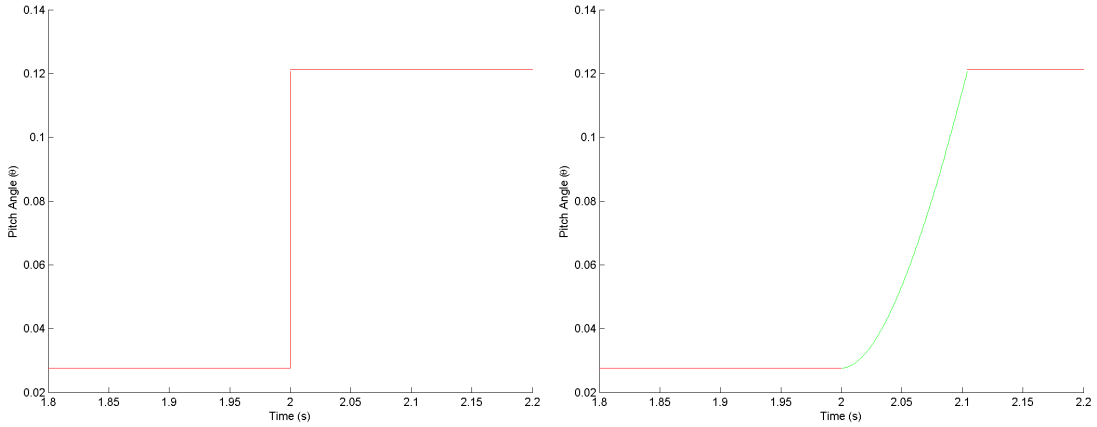


Figure 3.11: θ during concatenation of SL to CC trim primitive without and with inclusion of attitude rates respectively where $V = 30m/s$

rate constraints (3.16). t_{max} represents the maximum time required to transition between two primitives. This is calculated by comparing the time required to the rotate the UA from the current state to the desired state in the roll (t_ϕ) and pitch (t_θ) axes.

$$\begin{aligned} t_\phi &= \frac{\Delta\phi}{P_{ss\,max}} \\ t_\theta &= \frac{\Delta\theta}{Q_{ss\,max}} \\ t_{max} &= \vee(t_\phi, t_\theta) \\ Q_{ssA} &= \frac{\Delta\theta}{t_{max}} \\ P_{ssA} &= \frac{\Delta\phi}{t_{max}} \end{aligned} \tag{3.16}$$

The concatenation of SL and HA flight modes is presented to illustrate the how the steady state attitude rates are adjusted to ensure that the UAS platform correctly reaches the desired state without exceeding the maximum attitude rate constraints (Figure 3.12). In this example scenario, $\Delta\theta$ is constrained in comparison to $\Delta\phi$. This results in $t_{max} = t_\phi$ and a reduction in Q_{ssA} (Figure 3.13).

This section presented an overview of the constraints applied to generate flight

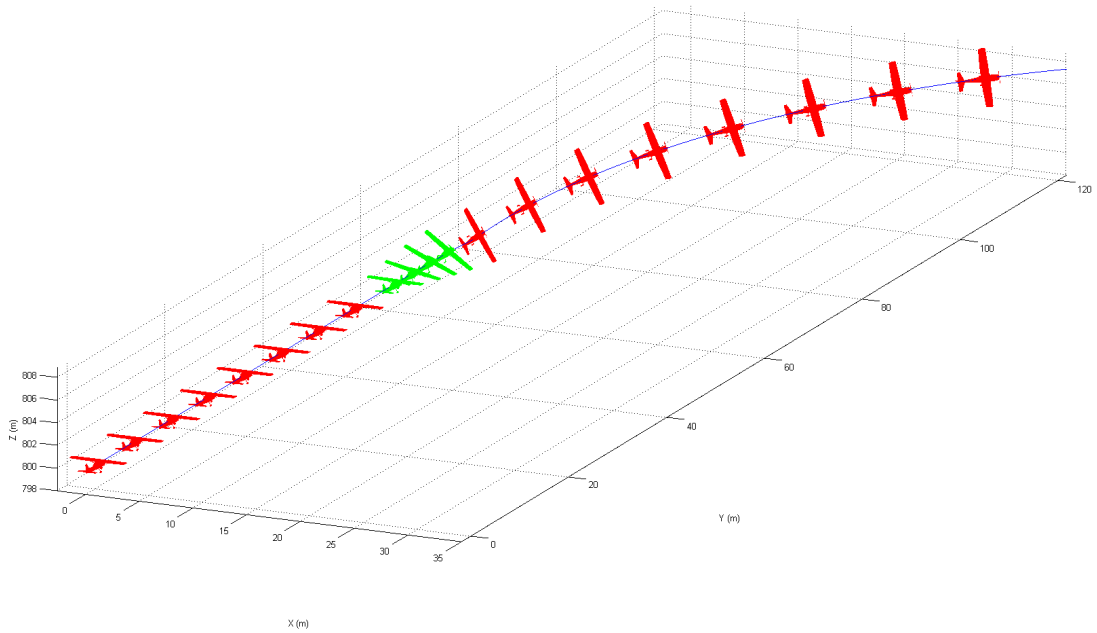


Figure 3.12: Concatenation of SL to HC trim primitive with inclusion of attitude rate constraints where $V = 30m/s$

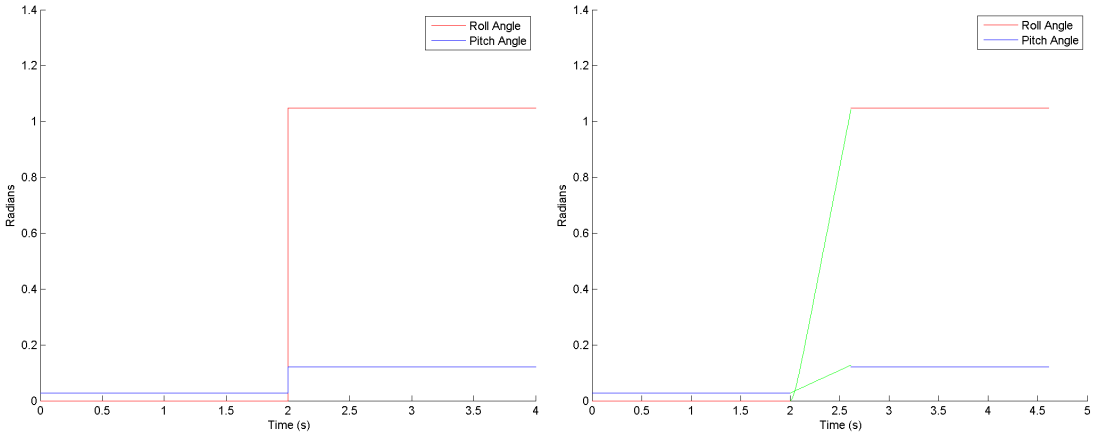


Figure 3.13: ϕ and θ during concatenation of SL to HC trim primitive without and with inclusion of attitude rate constraints respectively

manoeuvres for fixed wing aircraft using MA theory [48]. Trajectory segments are generated by concatenating individual trim primitive segments together to form a smooth feasible trajectory. It was shown that the inclusion of manoeuvre primitives ensured that the trajectory generated met platform performance

bounds and dynamic constraints; including attitude rate constraints. This allows for the generation of trajectories which incorporate the actual flight dynamics of fixed wing platforms with greater accuracy. This can allow for the tracking of trajectories with greater precision [91]. Whilst outside the scope of this research, a brief overview of trajectory tracking for UA is presented in the following section.

3.4 Trajectory tracking of feasible trajectories

The trajectory planning layer (Figure 2.3) is essentially a feed-forward controller and does not explicitly take external disturbances (e.g wind) into account. A trajectory tracking layer or guidance component is necessary for accurate tracking of the trajectory solution in the presence of dynamic external disturbances.

Numerous trajectory tracking algorithms have been presented in literature for accurate tracking of UA trajectories [12; 13; 99; 100]. Beard et. al [12] present a guidance and control software architecture which illustrates the inclusion of a trajectory tracking layer which minimises the tracking error between the desired trajectory solution and current platform state (Figure 3.14).

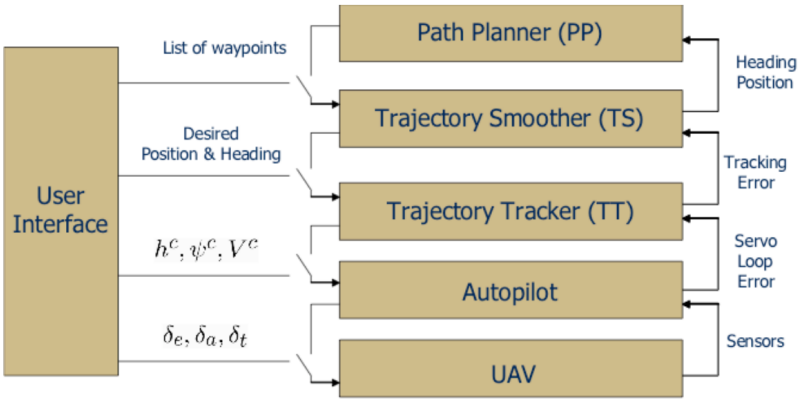


Figure 3.14: Inclusion of trajectory tracking layer within UAS guidance and control software architecture developed by Beard et. al [12]

Park [100] states that two approaches for trajectory tracking exist; implementation of the outer guidance loop and inner control loops separately or through the

use of an integrated approach where both loops are designed simultaneously. Designing both guidance and control loops synchronously can be achieved through the use of different modern control techniques including receding horizon MPC, differential flatness or neural network based adaptive controllers. Topsakal [13] applies the receding horizon MPC control technique to reduce tracking errors for primitives generated using MA theory in the presence of external disturbances (Figure 3.15).

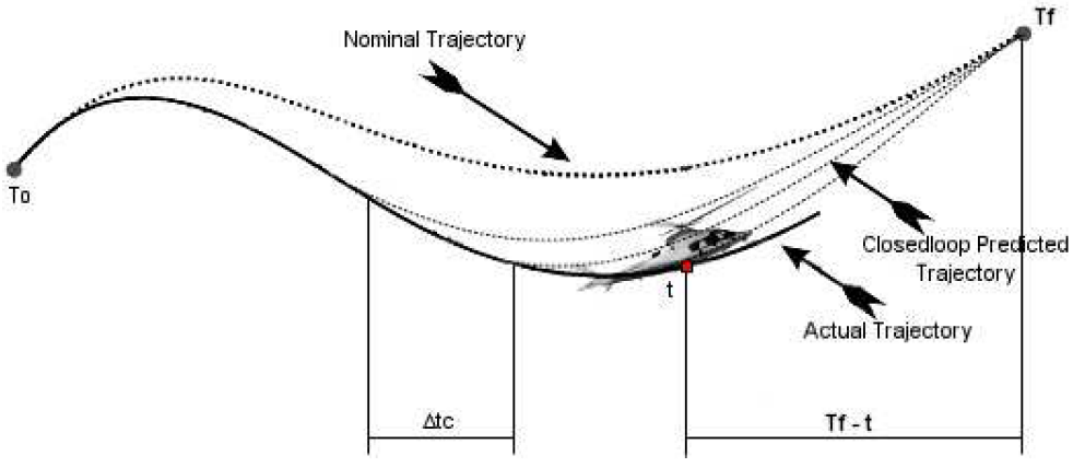


Figure 3.15: Comparison of nominal, predicted (using MPC) and actual UA trajectory by Topsakal [13]

During the concatenation of motion primitives being traversed in the presence of external disturbances, can result in an error between the actual final state of the current primitive ($a(p_k)$) and initial desired state of the next primitive ($d(p_k)$) (assuming a hybrid architecture is applied where primitives are selected sequentially using single stage DP optimisation). Topsakal has shown that receding horizon MPC can be applied to provide a more accurate prediction ($p(p_k)$) of $d(p_k)$. Kuwata [14] demonstrated using a MILP based UA trajectory planning algorithm that tracking error can be decreased by updating the UA position (due to the effect of external disturbances) after each iteration (Figure 3.16).

The inclusion of a trajectory tracking layer as a component of the onboard intelligent control architecture is necessary to minimise platform trajectory tracking error during actual autonomous UAS operations in the NAS. Additionally,

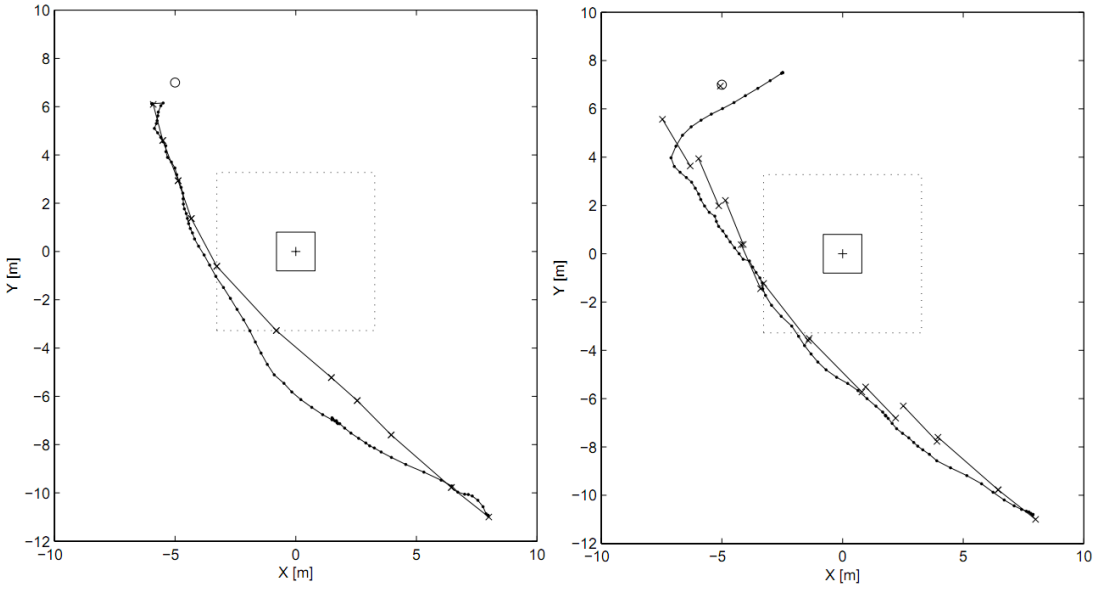


Figure 3.16: Comparison of Kuwata's MILP based UA trajectory planning algorithm without and with predicted position adjustment due to external disturbances [14]

the implementation of an MPC control methodology which iteratively adjusts $d(p_k) = p(p_k)$ in a sequential manner, could potentially allow for improved tracking of MA based trajectories in the presence of external disturbances. However, this research is focused on the development of a automated trajectory planning solution to provide UAS with the capability to operate with greater autonomy during low altitude operations in the NAS. Thus, the inclusion of a trajectory tracking layer is not investigated in detail as it expected that the trajectory tracking component can be implemented as a separate module.

In order to operate safely in low altitude environments, in proximity to terrain and obstacles, autonomous UAS require the onboard capability to ensure that trajectories generated are not only feasible, but also collision free. The following section presents the inclusion of safe states [79] to guarantee platform safety during autonomous operations in low altitude partially known environments.

3.5 Ensuring platform safety during trajectory planning

Safe UAS operations in cluttered environments requires the generation of collision free trajectories. This can be achieved by discretising the continuous flight track generated, and testing sampled points for collisions. However, the inclusion of trajectory collision detection alone does not provide the capability to undertake autonomous operations in a safe manner.

During autonomous operations where a management by exception control paradigm [4] is applied, in order to maintain full authority, the HDM should be able to veto UAS decisions and safely enter a holding pattern at any time. Schouwenaars [79] has applied a safe state formulation which allows a platform with non-holonomic constraints to enter a loiter state if the vehicle becomes trapped in local minima, thus preserving platform safety.

Schouwenaars [79] applies the safe state formulation (Figure 2.6) to two dimensional (2D) partially known environments. It was found that planning can be conducted in three dimensions (3D) [85; 89] to reduce the possibility of becoming trapped in local minima as the UA has the additional option of traversing over obstacles at a higher altitude. Furthermore, the generation of hold manoeuvres using MA theory allows for the inclusion of attitude rate constraints to provide greater trajectory trackeability. The following section provides the results of generating hold manoeuvres for fixed wing UA using MA theory, and the extension of the safe state formulation to 3D partially known environments.

3.5.1 Safe states in 3D partially known environments

Left and right coordinated turn trim primitives are performed in an alternate manner at sampled points along each trim primitive. Attitude rate constraints are implicitly considered through the inclusion of a transition manoeuvre between the current primitive and the loiter manoeuvre executed (Figure 3.17).

Using a finite receding horizon strategy, a safe state is deemed to be a state which is collision free and capable of safely executing a minimum turn loiter

manoeuvre, where the loiter manoeuvre does not traverse outside the finite horizon. This research project applies a spherical region of radius R_{sphere}, centered at the current platform position, to represent the known part of the operational environment.

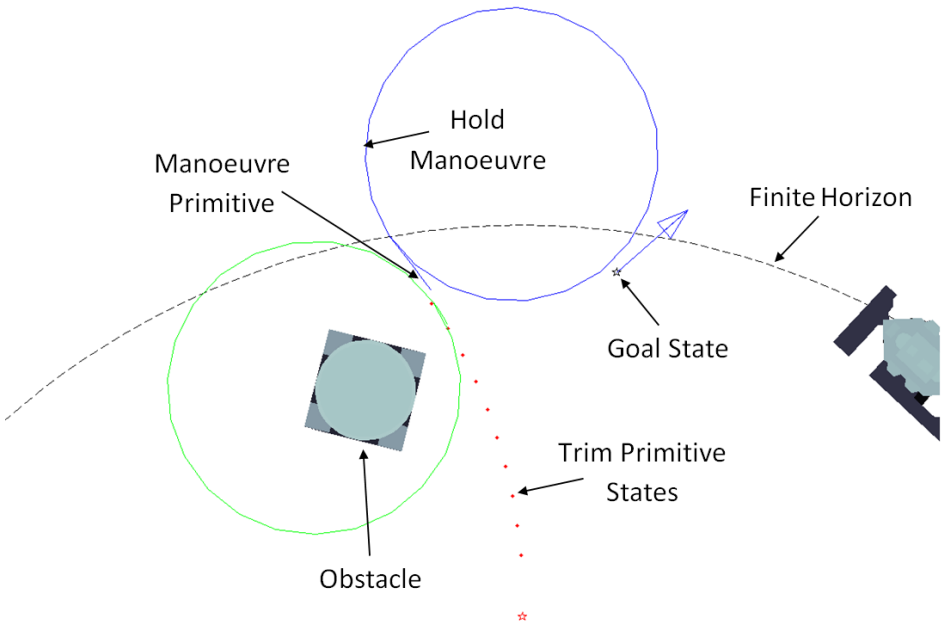


Figure 3.17: Execution of fixed wing UA hold manoeuvres where attitude rate constraints are included via MA theory

A fixed resolution automaton is generated using platform dynamics from the current state location (s_k). Left and right hold manoeuvres are then generated from automaton state locations as initial states (Figure 3.18). The use of alternating hold manoeuvres was applied to reduce overall computation time and algorithm memory footprint.

All fixed wing hold manoeuvres traversing outside the finite horizon cannot be considered safe unless full environment knowledge is available. Thus, only states of hold manoeuvres which can be executed within R_{sphere} are considered valid (s_{valid}) (Figure 3.19). The algorithm calculates the maximum euclidean distance between the furthest sampled point (xy_{max}) of each hold manoeuvre executed, and the centre point (C_{sphere}) of known environment abstraction. All states where

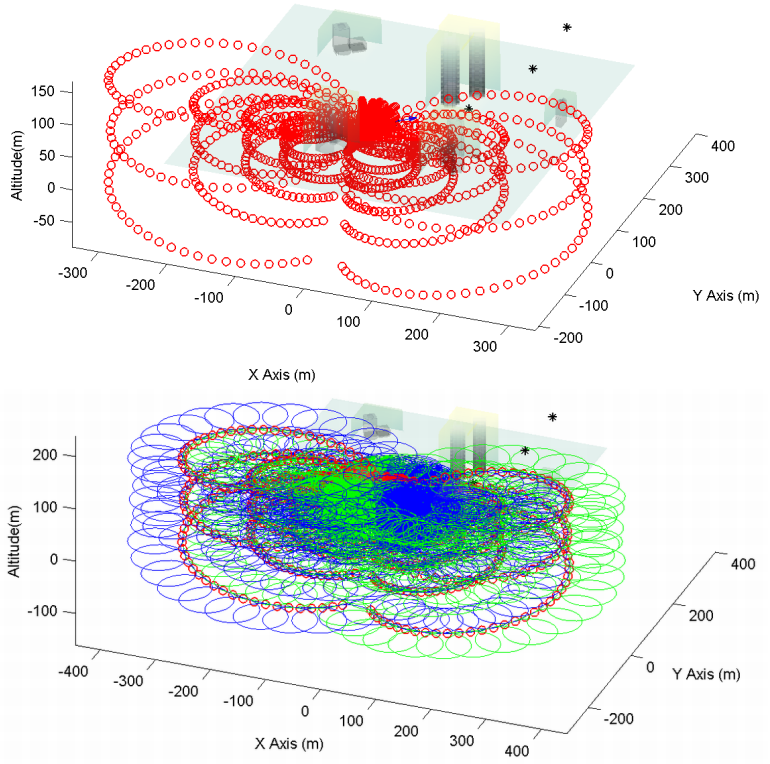


Figure 3.18: Generation of hold manoeuvres for automaton

the corresponding hold manoeuvres maximum euclidean distance from C_{sphere} is greater than R_{sphere} are culled (3.17).

$$\begin{aligned} x_{rel} &= x_H - x_{C_{sphere}} \\ y_{rel} &= y_H - y_{C_{sphere}} \\ xy_{euc} &= \sqrt{(x_{rel})^2 + (y_{rel})^2} \\ xy_{max} &= \max(xy_{euc}) \\ s_{valid} &= xy_{max} < R_{sphere} \end{aligned} \tag{3.17}$$

The valid states (s_{valid}) and corresponding hold manoeuvres resulting in collisions with obstacles and terrain (including the ground) are then culled (Figure 3.20) to ensure that the trajectory is safe and collision free. The remaining

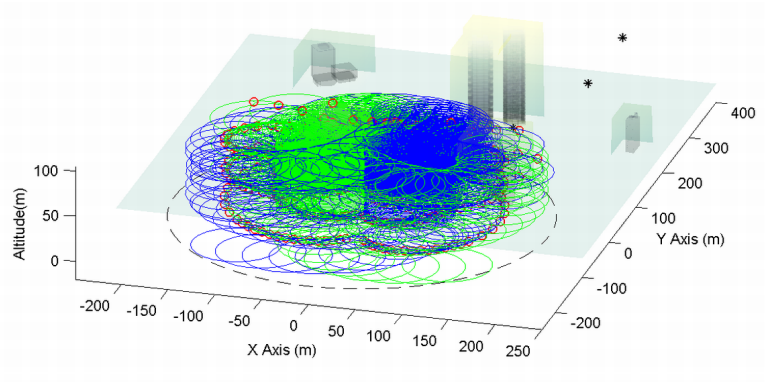


Figure 3.19: Valid states where the corresponding hold manoeuvres are executed within the finite horizon

states form the safe feasible collision free automaton (s_{safe}). This is achieved by testing to see if the altitude of each hold manoeuvre sample point z_H is above the terrain or obstacle at the corresponding lateral position (3.18).

$$\begin{aligned} \text{Threshold} &= \max\left(z_{obstacle}, z_{terrain}\right) \\ s_{safe} &= z_H > \text{Threshold} \end{aligned} \tag{3.18}$$

Without inclusion of states which result in loiter manoeuvres outside of the finite horizon, it was found that the minimum finite horizon radius R_{sphere} is limited by the UAs manoeuvrability. To ensure platform safety and to provide the HDM with the authority to veto and safely execute a hold manoeuvre at anytime, R_{sphere} is required to be greater than the minimum platform turn radius R_{min}.

The following section presents the concatenation of primitives to form a smooth trajectory in an optimised manner. An overview of previous optimisation methods applied to trajectories generated using MA theory are presented.

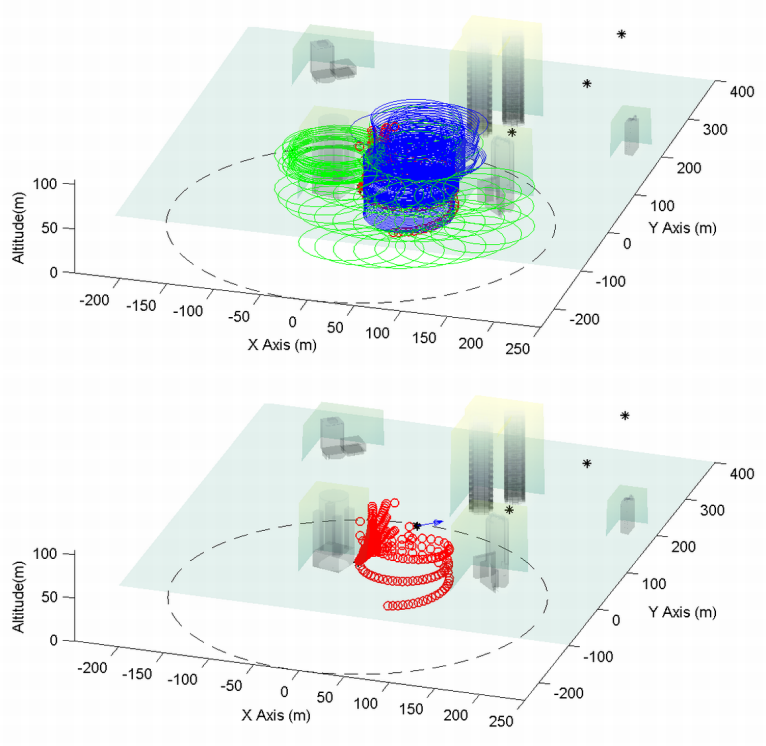


Figure 3.20: Safe states within finite horizon

3.6 Generating optimised trajectories through concatenation

MA theory [48], like geometric trajectory generation methods [71] requires the sequential selection of individual segments to form a smooth continuous trajectory. The final trajectory is formed through sequential concatenation of a set of selected trim primitives (and corresponding manoeuvre primitives, if required) where each trim primitive selected for execution can be considered as a stage.

Without an optimisation strategy in place, there is no guarantee that trajectories generated will meet HDM and mission criteria, thus an optimisation strategy is required in order to generate trajectories which best meet one or more mission objectives. Dynamic Programming (DP) [101] has been previously employed in related research [49; 53; 92; 102] for the optimisation of feasible trajectories generated through the application of MA theory. DP is a sequential optimisation

method which finds the least cost solution (optimal) from a set of alternatives over one or more stages. The following section presents an overview of the application of DP to MA based trajectory planning.

3.6.1 Dynamic programming applied to MA based trajectory planning

DP is a sequential optimisation process which finds the least cost solution (optimal) from a set of alternative solutions. The application of DP to the motion planning problem differs from the use of DP to graph search methods [43]. DP theory states that an optimal solution can be found if the exact cost of each stage is known and included in the optimisation process [43]. In comparison to the application of DP to trajectory planning with respect to a generic graph search implementation, the current UAS platform position can be treated as the current node. Each possible state the platform can reach through the execution of currently stored trim primitives must be treated as neighbouring nodes.

Expanding each neighbouring node would cause the algorithm to grow exponentially in computational complexity for each additional stage considered in the overall optimisation process.

MA theory is a method of discretising platform dynamics rather than planning space; an accurate representation of platform motion requires a higher automaton resolution resulting in larger number of neighbouring nodes. Therefore, the use of DP in autonomous motion planning can become very computationally expensive as the number of stages in the optimisation process increases.

To decrease the computational complexity and resulting time to plan, Frazzoli [53][48] applies a hybrid architecture to the motion planning problem for rotary aircraft. The hybrid architectures, involve integration of DP (optimised over single stage) with RRT [53] and with MPC [49].

Frazzoli [53] applies DP to the motion planning problem for rotary aircraft

over a single stage in the following manner (Eqn 3.19).

$$J^{\star}(q,h) = \min_{(\tau',q')} \left[\Gamma_T(q,h,\tau') + \Gamma_M(q',h') + J^{\star}(q'',h'')\right]$$

where

τ is the delay before the commanded transition to $q' \in Q_M$.
h' represents the position and heading at the start of the manoeuvre.
q'' and h'' represent the new state at the inception of the new trim trajectory.
Γ_T and Γ_M indicate the cost associated with the trim and manoeuvre portions of the commanded transition.
The optimal control $(\tau', q')^{\star}$ is the minimiser of the overall cost function.
The optimisation requires the solution of a mixed-integer program, with one continuous variable (τ'), and one discrete variable (q').

(3.19)

The DP formulation requires selection of the optimal manoeuvre and corresponding jump time from a predefined set of manoeuvres. Section 4.1.1.2 describes how each discrete jump time (for a given manoeuvre) can be represented as a unique alternative as it will result in a different final state if executed. The optimal manoeuvre and corresponding jump time will have the least cost, where the cost is an aggregated value representing the desirability of a given alternative with respect to a given criteria set (Section 4.1.1.3). Frazzoli et. al [53] have applied two specific criteria during manoeuvre and jump time selection: minimising euclidean distance of current (s) and goal (g) states (criterion $\text{crit}_{(|g-s|)}$); and minimising platform yaw (psi_s) and goal yaw (psi_g) angles (criterion $\text{crit}_{(|\Delta\psi|)}$) during optimal manoeuvre selection.

3.6.2 Application of DP to this research project

This research uses the DP search algorithm, but limits the search to single stage optimisation. This converts the DP algorithm to a greedy search implementation, which essentially chooses the most optimal trim primitive, trim execution time and manoeuvre execution time required to execute the optimal trim primitive over a single stage. The UA position after execution of the optimal trim primitive is taken as the next node for expansion, and continues until the goal is reached.

Executing a DP search algorithm iteratively over a single stage without explicit consideration for future stages ensures that the computational complexity and resulting time to plan remains comparatively lower than generating a solution over multiple stages. However, not considering all stages during the optimisation process means that global trajectory solution optimality and completeness cannot be guaranteed. Global path solution optimality and completeness can be guaranteed through the application of an intelligent control architecture with a mission/path planning layer which uses a deterministic search algorithm to generate an optimal set of waypoints from the current state to the goal location [43].

During DP optimisation over a single stage, scenarios may exist where the platform becomes trapped in local minima (e.g. in the presence of pop-up obstacles). UAS motion planning in 3D space allows for the execution of certain motion primitives (e.g. helical ascent) to escape local minima and continue operations [89]. In addition, during operations in dynamic and partially known environments, a greedy motion planning implementation can suffice as it may not be possible to find a globally optimal trajectory solution (e.g. due to limited environment representation).

3.7 Summary of findings

This chapter presented the generation of feasible collision free trajectories for fixed wing platforms using MA theory. Planning in 3D environments was possible through the formulation of common aircraft flight modes and attitude rate constraints were included through manoeuvres primitives.

The inclusion of safe states [79] in 3D partially known environments allow for HDM authority to veto and safely loiter at any time. Furthermore, it provides the platform with the capability to continue autonomous operations in partially known environments in the presence of communications link failure as the platform can safely enter a loiter state if the planner cannot compute a valid plan in the time available.

The computational complexities of applying the DP optimisation to MA based trajectory generators is discussed. It was shown that previous research applied

hybrid architectures [49; 53; 92; 102] where DP is limited to optimisation over a single stage to compute feasible trajectories in a computationally efficient manner.

During DP optimisation, Frazzoli [53] applies two specific criteria during manoeuvre and jump time selection: minimising euclidean distance of current (s) and goal (g) states ($\text{crit}_{(|g-s|)}$); and minimising platform yaw (psi_s) and goal yaw (psi_g) angles ($\text{crit}_{(|\Delta\psi|)}$) during optimal manoeuvre selection.

During UAS operations where a HDM is responsible for the planning and navigation aspects, the HDM applies their unique decision making style and preferences to form decisions for a given mission scenario. The criteria applied by Frazzoli may not accurately represent mission requirements as the candidate HDM may have their own perceptions on which criteria's are relevant to the current mission scenario and the preference given to each relevant criteria. Furthermore, if ($\text{crit}_{(|g-s|)}$) and ($\text{crit}_{(|\Delta\psi|)}$) do not completely encapsulate HDM decision strategies, then this may suggest the presence of additional relevant criteria. The following chapter investigates application of potential decision making methodologies to generate feasible trajectories based on MA theory which may represent HDM and mission requirements more accurately.

4

Embedding Human Expert Cognition into Trajectory Planning

Many problems can be solved through the application of decision analysis and decision aid techniques. The decision aid process generally provides a HDM with the most appropriate solution from a given set of alternatives. Each alternative will have one or more characteristics (criteria) which represent different dimensions in which an HDM can view the desirability of a given alternative by.

During the course of flight operations, the pilot/UAS operator may have to consider multiple criteria in order to achieve mission success. Examples of mission criteria generally include: achieving the mission goal/s; safety of the vehicle, the environment and the public at all times; mission efficiency (minimising time, fuel and/or cost); and/or limiting operations to below a specified altitude ceiling. Mission objectives and their priorities can dynamically change at any point during UAS operations (usually at the discretion of the operator).

Decision making during autonomous trajectory planning requires the selection of the most optimal feasible collision free trajectory with respect to one or more criteria. Gigerenzer et al. [103] have shown that HDMs do consider multiple criteria during real-life decision making processes. Therefore, the use of Multi-Criteria Decision Aid (MCDA) methodologies during autonomous trajectory planning may allow for convergence to a solution which better reflects overall

mission requirements.

MCDA is a category of decision aid methods in which decisions are formulated through the comparison of alternatives with respect to multiple criteria. Many MCDA techniques [46] have been published to date which can be used to determine the most suitable alternative, or to sort or rank a set of alternatives. MCDA techniques can roughly be divided into two categories: on the one hand Multiple Attribute Value Theory (MAVT) [51; 52], which aims at aggregating the multiple points of view into a unique synthesis criterion, and, on the other hand outranking methods [104] which aim at comparing the decision alternatives pairwisely and accept incomparability.

MCDA allows for the encapsulation of the HDM's decision style through the inclusion of preference information and a relevant set of criteria. Preference information can take various forms, among which for example the relative importance of each criterion to the HDM. The capture of these human preferences is called preference elicitation and depends on the HDM's individual decision experiences and training that he/she may have received. The following section presents a generic overview of the MCDA process and its use in the context of automated flight operations.

4.1 MCDA process

The MCDA process requires the implementation of algorithms which attempt to mimic aspects of the HDM's decision making style and take into account his/her preferences. Classically, an MCDA process can be divided into the following four steps [105]:

1. *Determining the relevant criteria and alternatives;*
2. *Evaluating the alternatives on all the criteria;*
3. *Eliciting the HDM's preferences related to the current decision problem;*
4. *Combining the evaluations and the preferential information to solve the decision problem and produce a decision recommendation.*

In the following sections, each of these steps is detailed in context of this research.

4.1.1 Determining relevant criteria and alternatives

The automated decision process selects the most desirable solution from a given set of alternatives. For the most optimal solution to be determined, the decision algorithms must rank the alternatives from the most desirable to the least desirable. The most optimal solution is the alternative which best meets a criteria or a set of criteria, where these criteria represent attributes which are relevant to mission objectives. During the decision process, the problem requires translation to a theoretical planning space. The following section provides an overview of planning space and its implementation for this research project.

4.1.1.1 Planning space

The planning space is an approximation of the real world environment where the path planning or motion planning problem can be solved. The difference between path planning and motion planning is that path planning is commonly referred to as discrete planning space (or configuration space) whilst motion planning utilises continuous planning space.

UAS path planning applications are generally represented in two dimensional (2D) or three dimensional (3D) configuration space (x, y, z) where possible platform states are represented as discrete nodes. A search can be conducted on this configuration space using graph search techniques [43] to generate a list of nodes which represent most optimal solution.

For the inclusion of platform dynamic constraints (in particular fixed wing variants), there is a need to represent the planning space in six DOF $(x, y, z, \theta, \phi, \psi)$. Increasing planning space dimensionality can greatly increases the number of nodes which represent the planning space. Planning in higher dimensionality may result in a time to plan which is not feasible, thus may not be suitable for real-time onboard re/planning.

Frazzoli [53] has formulated a hybrid architecture which allows for the generation of feasible trajectories using MA theory without resorting to planning with

higher dimensionality. The path planning and trajectory generation aspects are treated as separate components. The path planning component finds an optimised set of waypoints through 2D/3D configuration space using a graph search algorithm from the current location to the final goal location. The trajectory planning component applies DP optimisation methods to MA theory to select the most suitable trajectory solution from a set of decision alternatives. The optimal set of trajectory manoeuvres are concatenated to form a feasible trajectory. The following section details the formulation of the decision alternatives.

4.1.1.2 Alternatives

The decision making process can be defined as the selection of the most appropriate solution from a set of alternatives (A). The motion planning problem is defined as the selection of the most optimal manoeuvre and corresponding jump time which will allow the platform to reach the desired goal state.

This research project applies the DP search algorithm, but limits the search to single stage optimisation. This requires selection of the optimal manoeuvre and corresponding jump time from a predefined set of manoeuvres for each stage. Each trim primitive can be executed for any given length of time ($t_p = [t_{min}, t_{max}]$). Each discrete jump time, for a given manoeuvre, can be represented as a unique decision alternative, as it will result in a different final state if executed (Figure 4.1). Let A be the set of such alternatives.

For this research problem, the decision alternatives generated are sets of trim primitives (p) which can be safely executed by the platform (Figure 3.20). The automaton represents a set of sampled states (i) which the UAS can reach from its current state. Thus, the total number of alternatives for (m) trim primitives is:

$$|A| = \sum_{n=1}^{m} p_n i_n \tag{4.1}$$

HDM's select the alternative which appears most desirable by viewing how well each alternative meets their set of examining criteria. During RPV operations, the UAS operator may consider multiple criteria including: platform safety; successful completion of the mission; minimising fuel, time, and/or distance; or

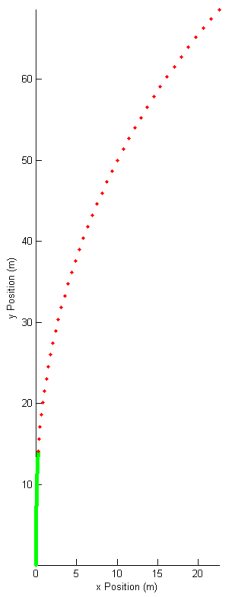


Figure 4.1: Discrete jump times for a coordinated turn trim primitive

minimising deviation from the current path. Gigerenzer [103] has shown that HDMs do consider multiple criteria, however it was found that they generally consider only a limited number of criteria during the decision making process. Thus, the application of MCDA to the automated trajectory planning may generate solutions which better encapsulate aspects of the candidate HDMs decision making process. The following section presents an overview on the application of criteria to a set of alternatives in order to form decisions.

4.1.1.3 Criteria

The criteria represent different dimensions with which an alternative can be viewed by. In literature, it was found that Frazzoli et. al [53] applied two such criteria: minimising euclidean distance of current (s) and goal (g) states (criterion $\text{crit}_{(|g-s|)}$); and minimising platform yaw (ψ_s) and goal yaw (ψ_g) angles (criterion $\text{crit}_{(|\Delta\psi|)}$) during optimal manoeuvre selection.

If $\text{crit}_{(|g-s|)}$ and $\text{crit}_{(|\Delta\psi|)}$ do not completely encapsulate the HDM's decision strategies, then the inclusion of additional criteria allows the onboard trajectory planner to take into account certain aspects of the mission which cannot be

considered using only the current two criteria. For example, executing very sharp turns (high bank angles) can lead to platform instability [106].

Platform safety can be implicitly considered through the inclusion of $\text{crit}_{(|\phi|)}$ which focuses on the minimisation of high platform roll angles (ϕ_s). It is important to note that through the implementation of MA theory, platform safety can be increased without penalising platform manoeuvrability.

The second additional criterion ($\text{crit}_{(|g_z - s_z|)}$) considers the minimisation of the altitude of the goal (g_z) and current state (s_z). For decision scenarios where the goal is not at the same altitude as the platform, this criterion captures how focused a HDM is on reaching the required altitude.

4.1.2 Evaluating alternatives on all criteria

In order to perform decisions on the set of alternatives (e.g. generating the most optimal decision or ranking/sorting), an evaluation scale needs to be attached to each criteria. Each alternative is then evaluated by placing a cost to go (from current state to the alternate state) on all attached criteria.

Whilst Frazzoli has not explicitly defined the criteria applied in literature, $\text{crit}_{(|g-s|)}$ can be expressed in 3D planning space as the euclidean distance between the goal location (g) and the current location (s). A lower cost ($c_{(|g-s|)}$) is placed on alternatives which drive the UAS platform closer to the goal (4.2).

$\text{crit}_{(|\Delta\psi|)}$ allows for greater control of the heading of the platform. For this research (ψ_g) represents the direction to next goal. The cost ($c_{(|\Delta\psi|)}$) can be calculated by taking the absolute difference between the desired (ψ_d) and absolute platform headings (ψ_a). Alternatives with a resulting ψ_a closer to ψ_d will have a lower cost placed on them (4.3).

$$c_{(|g-s|,i)} = |g - s| \in [\min|g - s|, max|g - s|] \tag{4.2}$$

$$c_{(|\Delta\psi|,i)} = |\psi_d - \psi_a| \in [0, \pi] \tag{4.3}$$

The evaluation of $\text{crit}_{(|\phi|)}$ has been performed by placing a greater cost ($c_{(\phi,i)}$) on trim primitives which are executed with higher roll angles (4.4). Finally, a greater cost $c_{(|g_z - s_z|)}$ is placed on trim primitives which do decrease the relative

vertical distance between the platform state (s_z) and goal state (g_z) for $\text{crit}_{(|g_z - s_z|)}$ (4.5).

$$c_{(|\phi|,i)} = \phi \in [-\phi_{max}, \phi_{max}] \tag{4.4}$$

$$c_{(|g_z - s_z|,i)} = |g_z - s_z| \in [\min |g_z - s_z|, max|g_z - s_z|] \tag{4.5}$$

Each candidate HDM may have their own perception of the relative importance of each criteria and thus the desirability of the alternatives presented. If an automated onboard trajectory planner applies multiple criteria without accounting for the relative importance placed on each criteria by the candidate HDM, the trajectory solution maybe quite different from what the UAS operator expects. The following section provides an overview of methods present in literature which formulate preferences through the analysis of HDM decision data.

4.1.3 Eliciting preferences from candidate HDM decision data

Roughly speaking, this elicitation can be performed either by questioning the HDM directly on the values of the various preferential parameters, or by extracting this information via a disaggregation technique from an order on some alternatives which the HDM is able to express.

To capture such expert knowledge in a direct way, one can use the MACBETH technique (Measuring Attractiveness by a Categorical Based Evaluation Technique) [107]. MACBETH's goal is to build a cardinal scale measuring the attractiveness of options through a learning process involving an interactive software. The HDM is asked to perform qualitative pairwise comparisons regarding his preferences between various evaluation levels and express himself on a scale reaching from *very weak* to *extreme*.

A well-known disaggregation approach is UTA (UTilité Additive) [50]. Here the HDM is tasked first with ranking a few well-known alternatives. Linear Programming (LP) techniques are then used to perform an ordinal linear regression in order to determine a preference model which is consistent with the HDM's

overall preferences. Both MACBETH and UTA approaches generate value functions and weighting vectors which correspond to the HDM's preferences. These can then be used in MAVT based decision algorithms.

UTA has been selected as the candidate method for the conversion of HDM decision strategies to preference parameters as it allows for more intuitive capture if the alternatives are presented to the HDM visually through a Graphical User Interface (GUI) (Figure 4.2). This research uses the UTA algorithm written for the MATLAB compiler by Bous [108].

The following section provides an overview of the formulation of preferences through the application of UTA theory.

4.1.3.1 Overview of UTA theory

UTA theory [50] uses the preference aggregation-disaggregation principle to infer global HDM preference models (utility functions and corresponding weighting vectors) from given preferential structures (HDM decisions). Let $A = \{x, y, z, \ldots\}$ be the set of alternatives and $J = \{g_1, \ldots, g_n\}$ be the set of n criteria. Each criterion can be seen as a real-valued function on the set A. Let $g(x)$ be the vector of evaluations of alternative x of A on the criteria of J. Each criterion can be represented as a non-decreasing real valued function defined on A [46], as

$$g_i : A \rightarrow [g_{i^*}, g_i^*] \subset \Re / x \rightarrow g(x) \in \Re \tag{4.6}$$

where $[g_{i^*}, g_i^*]$ is the criterion evaluation scale, g_{i^*} and g_i^* are the worst and best level of the i-th criterion respectively, $g_i(x)$ is the evaluation or performance of action x on the i-th criterion and $g(x)$ is the vector of performances of action x on the n criteria. From the definitions given, the following preferential scenarios can be deduced:

$$\begin{cases} g_i(x) > g_i(y) \Leftrightarrow x \succ y \text{ (x is preferred to y)} \\ g_i(x) = g_i(y) \Leftrightarrow x \sim y \text{ (x is indifferent to y)} \end{cases} \tag{4.7}$$

UTA [50] structures a set of actions using weak-order preferences, where pairwise comparisons of alternatives are performed and one alternative is given a preference or indifference over the other. The additive value functions (based on

multiple criteria) are adjusted so that the resulting structure would be as consistent as possible as the initial structure. In the aggregation context, the criteria model is known whilst the global preferences are unknown. Conversely, disaggregation refers to the inference of preference models from a given set of global preferences. The given set of global preferences may be [46]:

1. a set of past decision alternatives (A_R: past actions)
2. a subset of decision actions, if A is large ($A_R \subset A$)
3. a set of fictitious actions, consisting of performances on the criteria, which can be judge the HDM to perform global comparisons (A_R: fictitious actions)

As the number of A increases, the number of pairwise comparisons (and subsequently, memory and processing time) required increases exponentially. Thus, a smaller subset of A was pseudorandomly selected from the full set for faster convergence whilst retaining a high resolution set for the HDM to interact with during the knowledge capture process. To overcome the possibility that the downsampled subset does not accurately represent the full set of A, multiple pseudorandom subsets are generated (see Appendix B for further details).

This concludes the overview of formulating preferences using UTA theory. The following section outlines the inclusion of preferences formulated using UTA within automated decision making algorithms to rank the alternative set.

4.1.4 Determining a ranking of the alternatives

In order to determine which alternative is the most attractive for the HDM, a ranking of all the alternatives is computed. This allows the HDM direct access not only to the "best" solution, but to the remaining solutions and corresponding rankings. Many MCDA techniques [46] have been published to date which can be used to sort and rank a set of alternatives. For example, outranking and Analytic Hierarchy Process (AHP) approaches use pairwise comparisons to sort the alternatives in order of preference. Unlike some outranking methods, for example ELimination and Choice Expressing REality (ELECTRE) which form

a hierarchy of preferences by classifying alternatives as; preferred, indifferent, incomparable; AHP uses numerical measures attached to multiple criteria to form a ranking order with relative degree of preference presented.

Alternatively, MAVT can be applied to the multi-criteria decision problem, where multi-attribute preference functions can be decomposed into multiplicative or additive forms. Value functions are used to represent HDM preferences for each criteria. In order to represent the HDM's priorities, weightings are applied to scale value functions. Computational decision aids such as UTA and MACBETH can be applied for preference elicitation from expert decision knowledge to form value functions and corresponding weightings.

The aggregation technique used here is based on MAVT and requires the value functions and the weights obtained by the UTA technique. Consequently, the aggregation formula (4.8) is applied on the set of feasible alternatives. Thus, each of the alternatives gets an overall value (u), which allows to rank them from the most to the least attractive one. The criteria aggregation model in UTA is assumed to be an additive value function of the following form:

$$u(g(x)) = \sum_{i=1}^{n} w_i u_i(g_i(x)) \quad \forall x \in A \tag{4.8}$$

where u_i $(i = 1, \ldots, n)$ are real-valued functions called marginal value functions which are normalised between 0 and 1, w_i is the weight of criterion i, and u is the overall value function. A higher value of u_i is associated with a better alternative on criterion i.

In UTA, the ranking given by the HDM on a subset of alternatives is transformed into a set of linear constraints on u, which are added to the UTA disaggregation LP (see [50] for further details). The objective of this LP is to minimise the gap between the initial ranking given by the HDM and the one produced by the aggregation model. The output of the UTA LP is a set of value functions and associated weights which represent the HDM's preferences, based on the input ranking that he/she provided.

4.1.5 Summary of findings

This section presented a brief overview of MCDA and outlines the MCDA process to generate feasible trajectories which applied aspects of the candidate HDM's decision styles. Alternatives were defined as unique feasible sampled states which could be reached by the UAS platform. Criteria represented different dimensions with which a HDM could view the desirability of each alternative by.

The UTA disaggregation technique was selected to formulate preference information to represent HDM preferences and priorities for each criteria. An additive, MAVT decision strategy was then applied to incorporate HDM preferences during the aggregation of value functions representing mission criteria. The following section details the application of the proposed MCDA process to the current research problem to generate trajectory solutions which more accurately represent HDM and mission objectives.

4.2 MCDA and mission priorities

This section applies the MCDA process to the current research problem to formulate preferences which represent HDM's mission priorities. The following section details an overview of the HDM data capture process.

4.2.1 Expert knowledge capture and decision modeling strategies

One way of viewing the trajectory planning problem using single stage optimisation is that the candidate HDM is presented with unique decision scenarios, where they must select the most appropriate trajectory segment in an iterative manner until the mission is completed. During trajectory selection, the HDM's preferences may vary depending on the decision scenario presented to them, for example, the HDM may have a different set of preferences in mind when the UA is closer to the goal as opposed to decision scenarios where the UA position is farther from the goal.

A decision scenario can be defined as the relative difference between the goal and UA positions $((x_g - x_p), (y_g - y_p), (z_g - z_p))$ and the relative orientation of

the UA with respect to the desired direction at the goal ($(\psi_g - \psi_p)$). In addition, the automaton generated will be unique to the platform roll angle (ϕ_p) due to the inclusion attitude rate constraints; this results in a unique set of A for the HDM to consider. Thus, each unique decision scenario can be represented as $((x_g - x_p), (y_g - y_p), (z_g - z_p), ((\psi_g - \psi_p)), \phi_p)$. Figure 4.3 shows an example scenario presented to the candidate HDM.

The capture of HDM decision data for each unique decision scenario only provides a discrete snapshot of the candidate HDM's decision preferences for that particular scenario. In order to perfectly model a HDM's decision style would require data capture over an extremely large (approaching infinity) set of unique decision scenarios; this is not feasible. Thus, a sampled set of unique scenarios (which represent a discrete approximated subset of unique decision scenarios) are presented to the HDM via the GUI during data capture.

In order to elicit human expert decision preferences, a GUI was developed to generate a set of simulated decision scenarios, and to capture the corresponding candidate HDM's decision patterns (Figure 4.2). The HDM uses the GUI to intuitively select what they consider to be the most suitable decision from a set of alternatives (discrete sample points along each trim primitive) for each unique decision scenario. The trim primitives include straight and level flight, climb, descend, coordinated turn, helical turn and helical descent manoeuvres.

120 unique decision scenarios are completed by each HDM to form a bank of HDM decisions (Figure 4.4). The HDM decisions are then used to form preferences, for inclusion into a MAVT based Automated Decision System (ADS), that generates trajectories which incorporate aspects of HDM decision strategies. The following section provides an overview of the formulation of preferences through the application of UTA theory to the current research problem (see Appendix A for further details).

4.2.2 Preference formulation using UTA

UTA is applied to all decision sets completed by the HDM to form a selectable bank of preference data (Figure 4.4). The following sections present three experiments on three different problem formulations. A least cost formulation (LC-2)

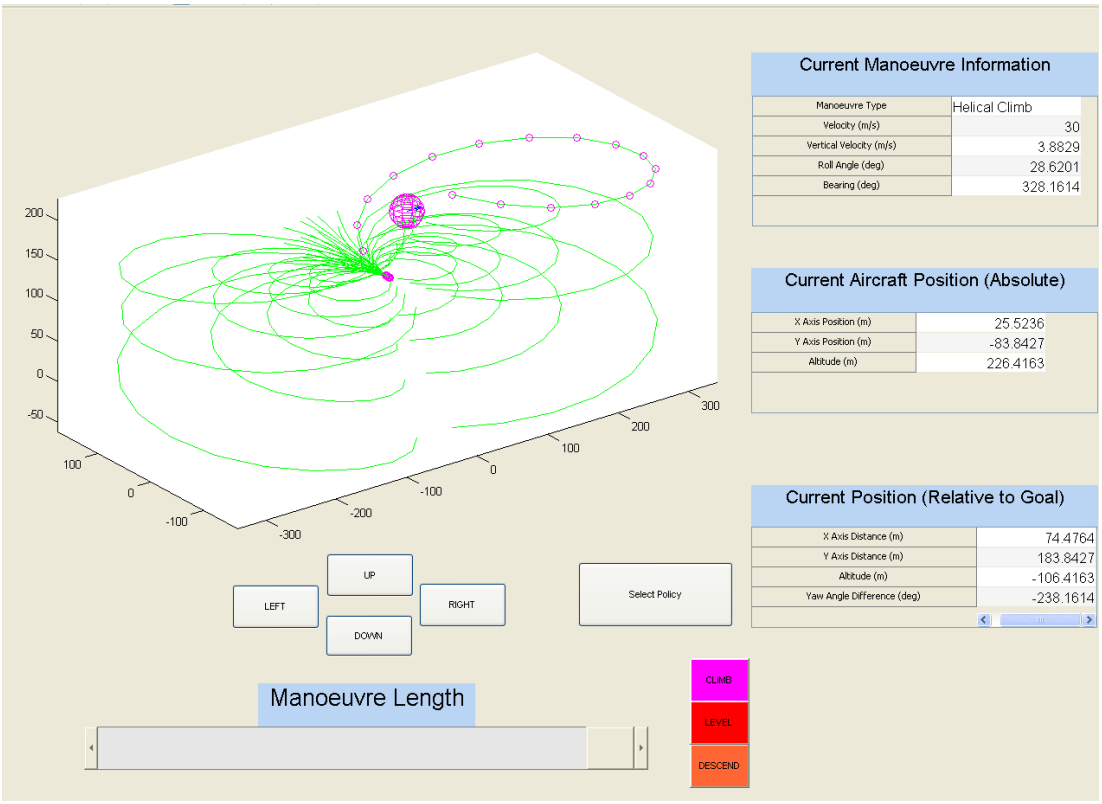


Figure 4.2: Graphical User Interface (GUI) developed for HDM data capture

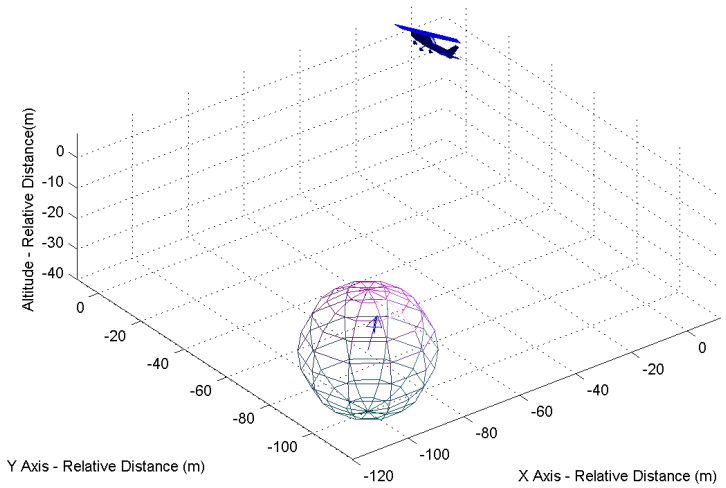


Figure 4.3: Example decision scenario presented to HDM

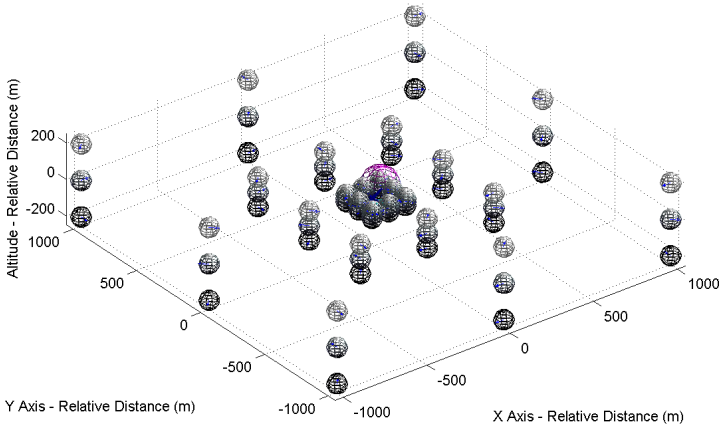


Figure 4.4: Decision sets completed by HDM

represents the inclusion of $\text{crit}_{(|g-s|)}$ and $\text{crit}_{(\Delta\psi)}$ with equal preference weighting as the reference solution. UTA-2 represents the inclusion of $\text{crit}_{(|g-s|)}$ and $\text{crit}_{(|\Delta\psi|)}$ where UTA is applied to generate value functions and weighting values using the candidate HDM's decision data. UTA-4 describes the inclusion of all four criteria presented in Section 4.1.1 where value functions and weighting values are again generated from candidate HDM decision data using UTA.

4.2.2.1 LC-2

An ADS applying the LC-2 decision algorithm generates trajectories where $\text{crit}_{(|g-s|)}$ and $\text{crit}_{(|\Delta\psi|)}$ are given equal preference. The cost functions $c_{(|g-s|)}$ and $c_{(|\Delta\psi|)}$ can be equivalently represented as value functions $\mu_{(|g-s|)}$ and $\mu_{(\Delta\psi)}$ respectively (4.9)(4.10). Value functions are an alternate way of representing cost functions where the most desirable alternative/s receive a value of 1 and the least desirable alternative/s receive a value of 0.

$$\mu_{(|g-s|)} = 1 - \left(\frac{c_{(|g-s|)}}{\max(c_{(|g-s|,1..n)}} \right) \tag{4.9}$$

$$\mu_{(\Delta\psi)} = 1 - \left(\frac{(c_{\Delta\psi})}{\pi} \right) \tag{4.10}$$

LC-2 may not accurately represent mission requirements as the candidate HDM may have their own perceptions on which criteria's are relevant to the

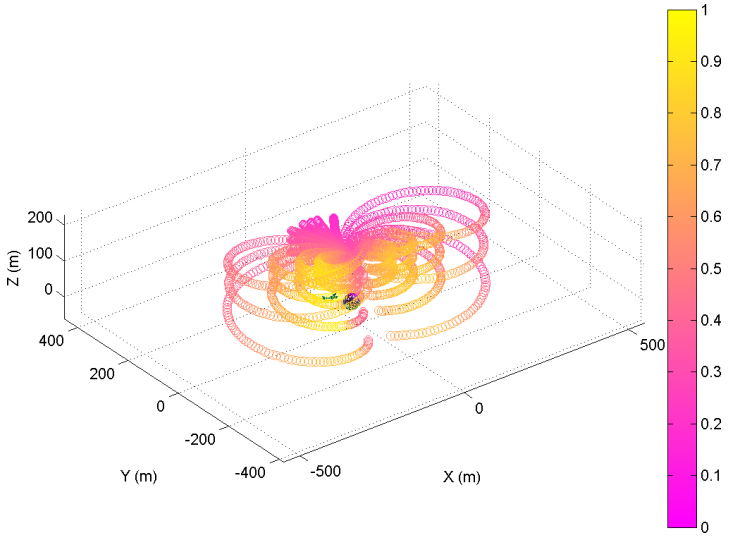


Figure 4.5: Normalised aggregated decision values for all criteria (LC-2)

current mission scenario and the preference given to each relevant criteria.

4.2.2.2 UTA-2

UTA theory is applied to the HDM decision sets to generate value functions and weighting values for $\text{crit}_{(|g-s|)}$ and $\text{crit}_{(|\Delta\psi|)}$ which provide a mathematical representation of the HDM's decision style for each given scenario. Figure 4.6 shows the value functions generated using UTA theory for the sample decision scenario when $\text{crit}_{(|g-s|)}$ and $\text{crit}_{(|\Delta\psi|)}$ are applied (Figure 4.3). Note that the weighting value is embedded within each value function (the maximum value of the value function corresponds to the weight coefficient of Formula 4.8).

The aggregate decision values generated by the ADS, with the application of UTA-2 (Figure 4.7) for the set of A shows that the desirable alternatives are concentrated into a singular region. In comparison, the aggregate decision values generated by LC-2 (Figure 4.5), UTA-2 shows a region focused near the goal state where alternatives have the highest utility decision values. This is due to the value functions generated using UTA placing a higher preference for $\text{crit}_{(|g-s|)}$ during optimisation (Figure 4.6).

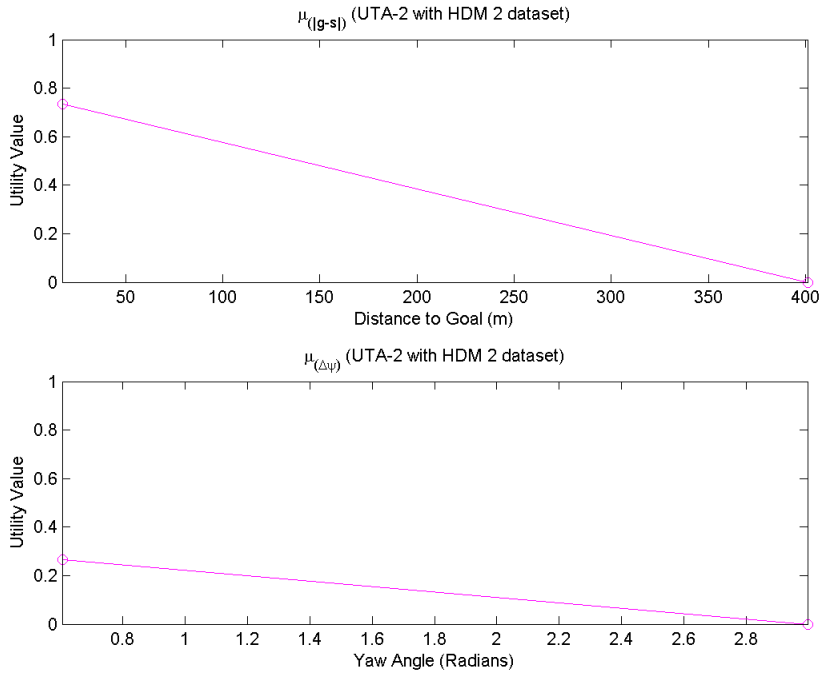


Figure 4.6: UTA value functions (UTA-2 with HDM 2 dataset) representing HDM preferences for sample decision scenario

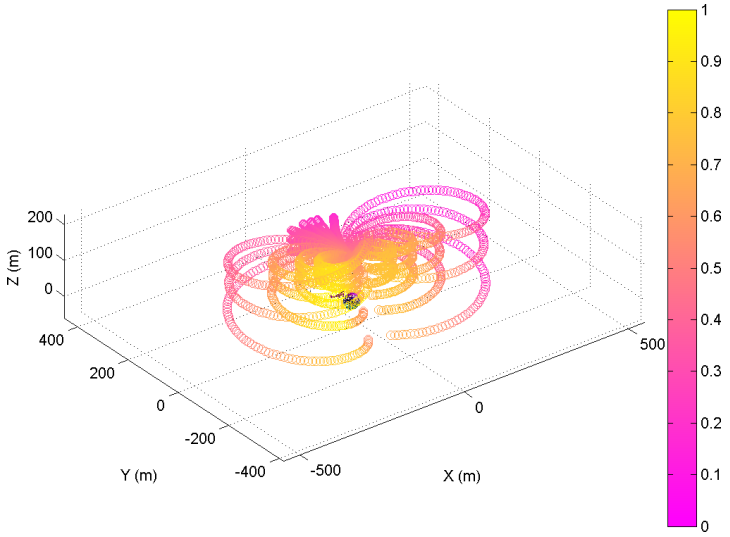


Figure 4.7: Normalised aggregated decision values for all criteria (UTA-2 with HDM 2 dataset))

4.2.2.3 UTA-4

In order to investigate if the inclusion of additional criteria can allow UTA to represent HDM decisions with further accuracy, UTA-4 applies two additional criteria ($\text{crit}_{(|\phi|)}$ and $\text{crit}_{(|g_z - s_z|)}$) during preference formulation using UTA theory. Figure 4.8 shows the value functions generated using UTA theory for the sample decision scenario (Figure 4.3) when the two previous and two additional criteria are applied.

The aggregate decision values generated by the ADS (UTA-4 with HDM 2 dataset) show several regions which are near optimal (Figure 4.9). It can be seen on Figure 4.8 that $\text{crit}_{(|\Delta\psi|)}$ has the greatest effect, thus all alternatives which have a low $c_{(|\Delta\psi|)}$ appear as near optimal solutions.

The following section compares UTA-4 and UTA-2 against LC-2 (reference least cost solution) to investigate the HDM decision modelling accuracy of UTA theory using HDM datasets captured.

4.2.3 Accuracy of UTA

The average results for all decision sets were compared to the trajectories selected by the HDM to determine how accurately the decisions were modelled by calculating the difference between the human and the ADS solutions for; roll angles (Δ_ϕ), euclidean position between goal and current state ($\Delta_{|g-s|}$) and platform yaw angles (Δ_{psi}). Figure 4.10 presents a graphical representation comparison UTA-4 and LC-2 decisions generated by the ADS for the example decision scenario (Figure 4.3).

The application of UTA-2 generated decisions which had a lower average Δ_ϕ, $\Delta_{|g-s|}$ and Δ_{psi} in comparison to the automated generation of decisions using LC-2 (Figure 4.11). This implies that $\text{crit}_{(|g-s|)}$ and $\text{crit}_{(|\Delta\psi|)}$ are relevant and considered by the HDM during the decision making process. Furthermore, the inclusion of additional criteria (UTA-4) generated decisions which were even closer to the HDM decisions captured than with just the inclusion of two criteria (UTA-2) (Figure 4.11).

Further analysis of each individual HDM's offline decision set shows how the UAS platform is expected to perform during autonomous operations with the in-

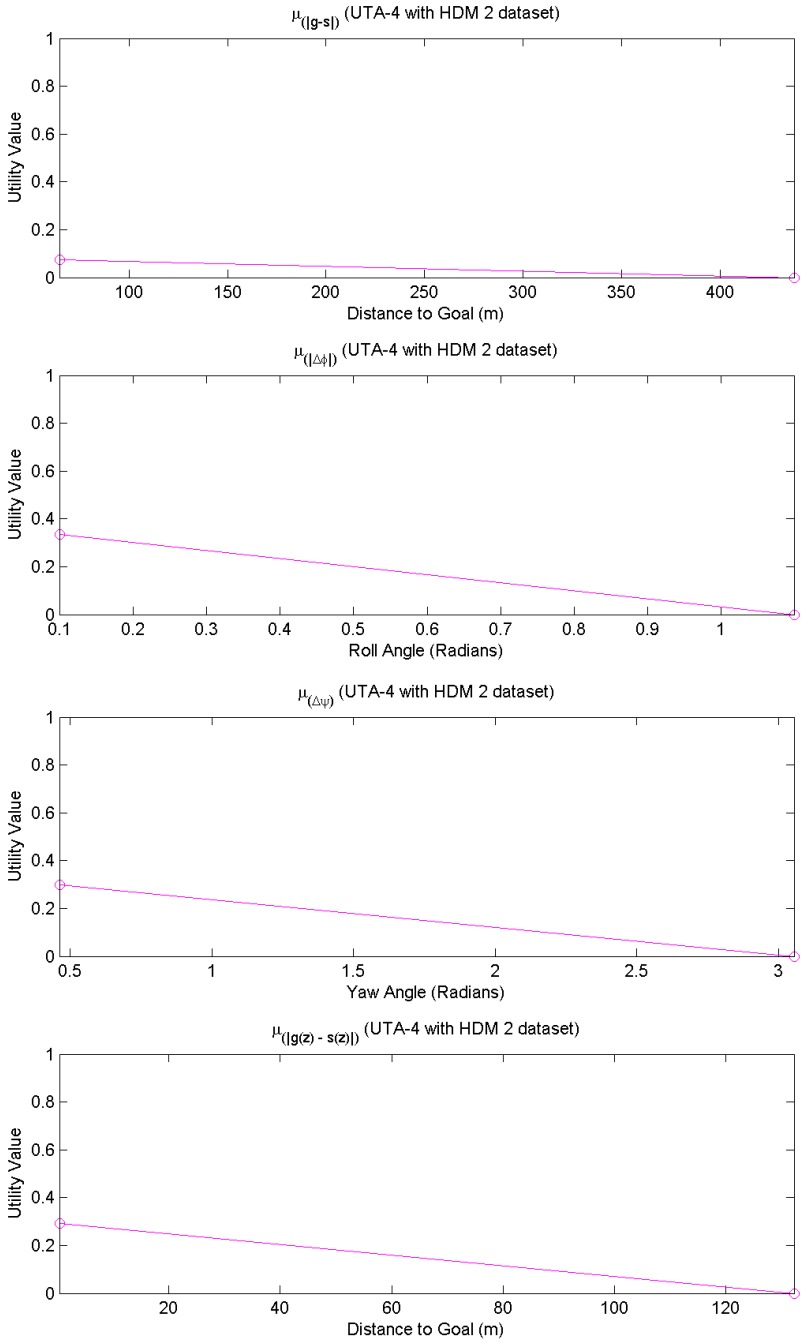


Figure 4.8: UTA value functions (UTA-4 with HDM 2 dataset) representing HDM preferences for sample decision scenario

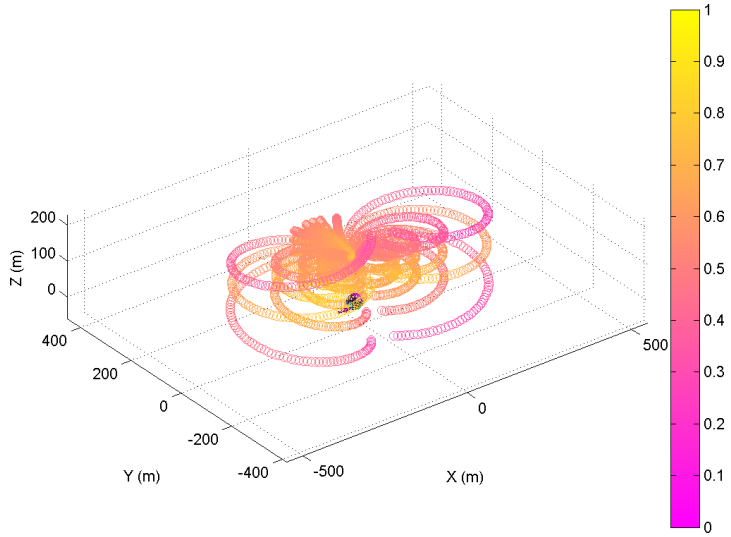


Figure 4.9: Normalised aggregated decision values for all criteria (UTA-4 with HDM 2 dataset)

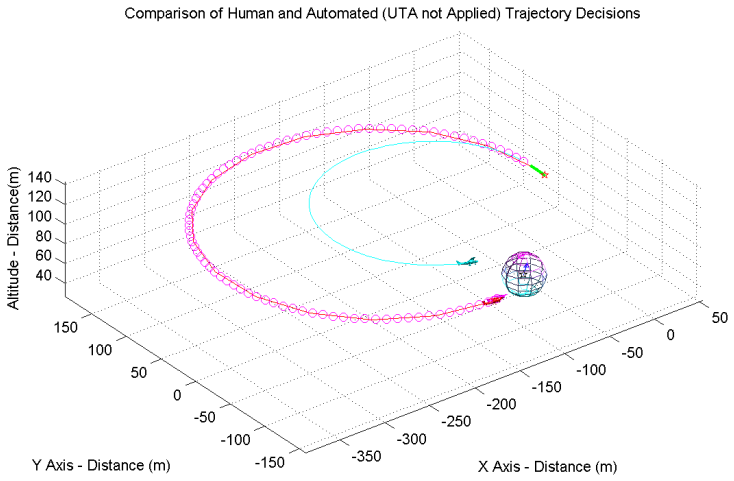


Figure 4.10: Comparison of HDM and automated trajectory decisions (LC-2) (UTA-4) for sample decision scenario (Figure 4.3)

clusion of HDM preferences (see Appendix C for comparison box plots all HDMs). HDM 2 generally executed flight manoeuvres where $\phi \in [20°, 40°]$ (Figure 4.12).

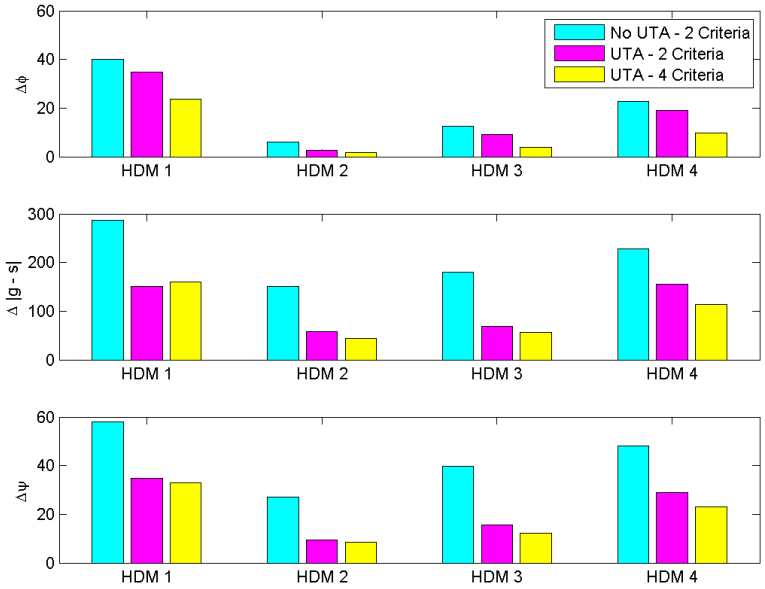


Figure 4.11: Average error comparison between human and automated trajectory decisions for all decision sets

The ADS with the inclusion of HDM preferences (UTA-4) executed primitives within a similar range to HDM 2. The LC-2 formulation does not explicitly take ϕ limitations into account, subsequently the ADS using an LC-2 optimisation had greater variance in the roll angle range of the primitives executed (Figure 4.12). It is expected that UTA-4 using HDM 2's decision data will not periodically execute manoeuvres with higher roll angle values unlike the ADS using an LC-2 optimisation. This is desired as the execution of flight manoeuvres with higher wing loading values has a greater possibility of leading to platform instability [106].

HDM 3 selected flight manoeuvres where a greater preference was placed on minimising the altitude of the platform s_z with respect to the goal altitude g_z (Figure 4.13). LC-2 only considers altitude minimisation as a component of $\text{crit}_{(|g-s|)}$, therefore during offline simulation, it was found that LC-2 optimisation had a greater variance in comparison to the HDM and UTA-4 trajectory solutions. It is expected that UTA-4 with the inclusion of HDM 3's decision data is more likely generate trajectories with lower $|g_z - s_z|$ values in comparison to the ADS using an LC-2 optimisation (Figure 4.13). This reflects on the candidate HDM's

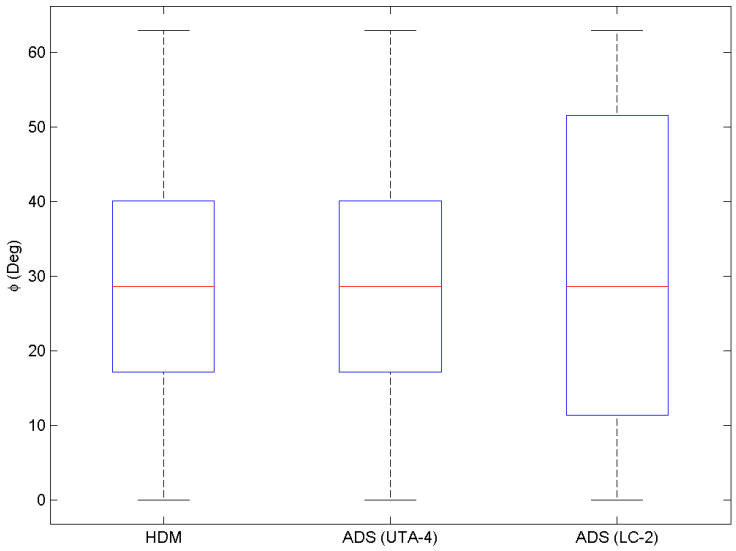


Figure 4.12: Box plots comparing UAS platform ϕ for offline trajectories selected by HDM 2, ADS LC-2 and ADS UTA-4 solutions

preference on maintaining a similar altitude to the goal which can be beneficial for certain missions e.g. airborne surveillance and video capture.

It was found that the ADS with the inclusion of human expert data to model preferences, generated decisions which were closer to the HDM decisions captured using the GUI implementation (Figure 4.11). Thus, the inclusion of preferences formulated using captured HDM decision data allows for the automated generation of trajectory decisions which are similar to the decisions generated by the candidate HDM for each given decision scenario. The following section demonstrates the inclusion of HDM preferences derived using UTA to generate feasible trajectories which mimic aspects of the candidate HDMs decision process in simulated 3D low altitude environments.

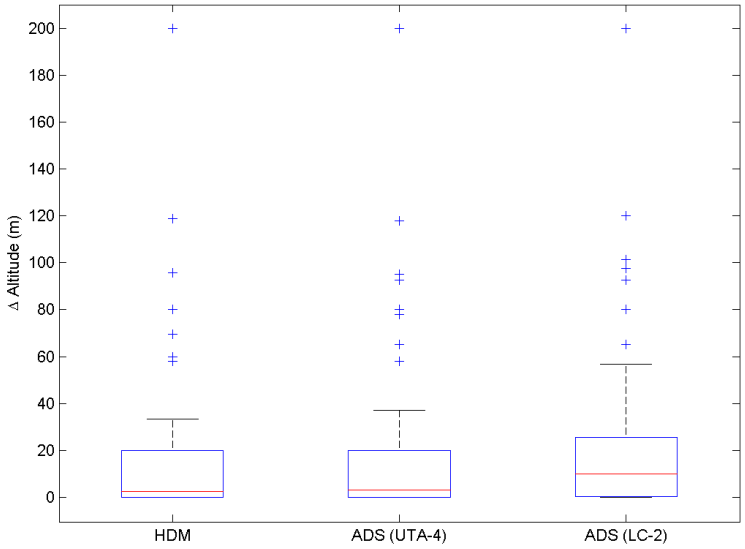


Figure 4.13: Box plots comparing UAS platform $|g_z - s_z|$ for offline trajectories selected by HDM 3, ADS LC-2 and ADS UTA-4 solutions

4.3 Results

This section presents the automated generation of feasible trajectories through the concatenation of primitives using MA theory (Section 3.1). The automated process encapsulate aspects of the HDM decision process through the inclusion of preferences formulated using UTA theory from HDM expert data captured.

4.3.1 Simulation setup

A simulated 3D terrain environment (figure 4.14) was setup in MATLAB to simulate mission scenarios where the UAS assignment includes safe and efficient navigation through a set of globally optimal waypoints. The simulation has been performed on a computer with an Intel Core 2 quad core processor operating at 2.8GHz to simulate how the inclusion of human expert data to the motion planning problem can lead to the generation of UAS flight trajectories which encapsulate aspects of the HDM decision process.

The ADS is tasked with generating an optimised, feasible and collision free trajectory through all mission level waypoints until the goal is reached (Figure 4.14). The waypoints can either be selected by the user, or provided by a mission planner. The advantage of using a mission planner is that global optimality is guaranteed as the planner will generate a set of waypoints which are globally optimal with respect to a predefined set of criteria. The mission planning solution by Wu [109] is used for the low altitude trajectory planning results in simulated environments with terrain present.

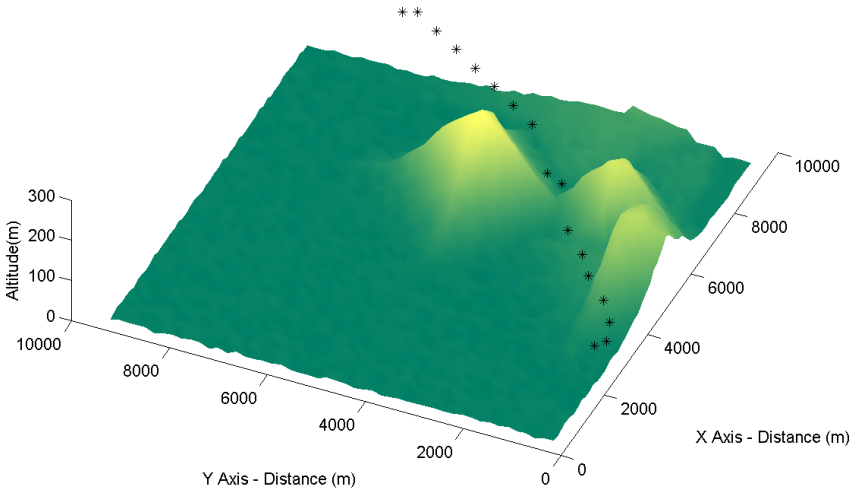


Figure 4.14: Simulated mission environment (terrain simulation 1)

The ADS generates a set of alternatives for each stage by selecting the number of primitives (m) and the samples per primitive (i) (Section 4.1.1). A large set of alternatives provides a greater number of final states which the platform can reach and a higher resolution of the region within the platforms performance bounds. Consequently, a large set of alternatives requires a greater computational effort and subsequently a longer time to plan. Table 4.1 lists the primitive types, (m) and (i) applied during automaton generation for the online simulations presented in this section.

Primitive Type	**Primitive No.** (m)	**Primitive Samples** (i)
Straight and Level	1	100
Coordinated Turn	12	100
Constant Climb	1	100
Helical Climb	12	100
Constant Descend	1	100
Helical Descend	12	100

Table 4.1: Primitive type, number and samples per primitive applied during online simulations

4.3.2 Preference selection during online planning

During online trajectory planning, the automated decision algorithm compares the current online decision scenario to the set of decision scenarios presented to the candidate HDM offline (Figure 4.4). A least squares formulation (4.11) is applied to map the preference data for the offline decision scenario which most closely matches the current online decision scenario.

The least squares formulations compares the following differences between the current online scenario and offline scenario set (Figure 4.4); distance to goal in x, y and z dimensions $(\Delta x, \Delta y, \Delta z)$ and platform roll angle $(\Delta\phi)$. The least squares formulation for n offline scenarios becomes:

For $i \in [1..n]$

$$LSQR_k = \min\left(\sqrt{(\Delta x)_i{}^2 + (\Delta y)_i{}^2 + (\Delta z)_i{}^2 + (\Delta\phi)_i{}^2}\right) \tag{4.11}$$

where $\Delta x, \Delta y, \Delta z, \Delta\phi \in [0..1]$

The ADS applies the preferences from the offline HDM decision with the lowest Least SQuares linear Regression ($LSQR_k$) value to the weighted sum formulation (4.8) to generate an optimised solution. The following section presents the results of the online simulations where the automated trajectory mimics aspects of HDM decision styles through the inclusion of preferences formulated using HDM decision data via UTA.

4.3.3 Simulation results

High altitude operations in civilian airspace are generally conducted in IFR under the guidance of air traffic control. Whilst automated trajectory planning can still provide benefits for UAS platforms operating at high altitudes, low altitude operations can be considered as more challenging, as terrain must be treated as a hazard during planning and operations.

Without the inclusion of collision avoidance methods, a safe output trajectory cannot be guaranteed, even with the use of optimal collision free waypoints. For the inclusion of collision avoidance during 3D trajectory planning using MA theory, the terrain map data is used to cull trim primitives which are below a specified terrain height, at the given grid location (Figure 4.15). This ensures that an optimised collision free trim primitive can be selected for each stage from the remaining collision free set of primitives.

The automated LC-2 solution is used as a reference and compared the solution generated by the ADS with the inclusion of the candidate HDM's decision patterns through UTA theory. The comparative trajectory applies the candidate HDM's decision style through the inclusion of HDM preferences formulated using UTA-4.

4.3.3.1 Terrain simulation 1

HDM 3's dataset was applied to UTA-4 and compared to the reference solution generated by LC-2 (Figure 4.16). Analysis of HDM 3's offline dataset showed that the HDM placed a greater preference on minimising $\text{crit}_{(|g_z - s_z|)}$ (Figure 4.13). Subsequently, during online trajectory planning in simulated environments, UTA-4 generated collision free trajectories which had lower $|g_z - s_z|$ on average than LC-2 (Figure 4.17). See Appendix D.1 for additional plots comparing UA trajectories generated for LC-2 and UTA-4 solutions for terrain simulation 1.

4.3.3.2 Terrain simulation 2

HDM 2's dataset was applied to UTA-4 and compared to the reference solution generated by LC-2 (Figure 4.18). HDM 2 preferred to minimise platform ϕ variance during the offline simulation set (Figure 4.12). LC-2 has a higher preference

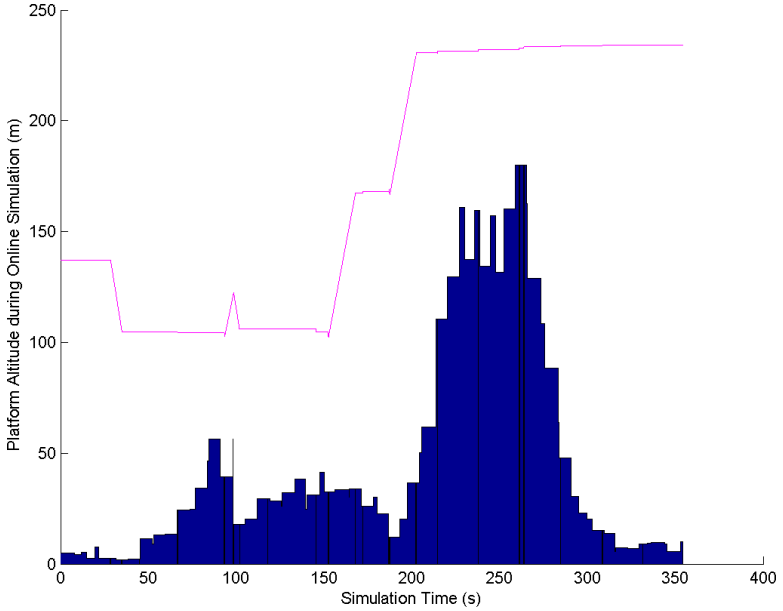


Figure 4.15: UAS platform altitude during simulation (UTA-4 with HDM 3 dataset) (terrain simulation 1)

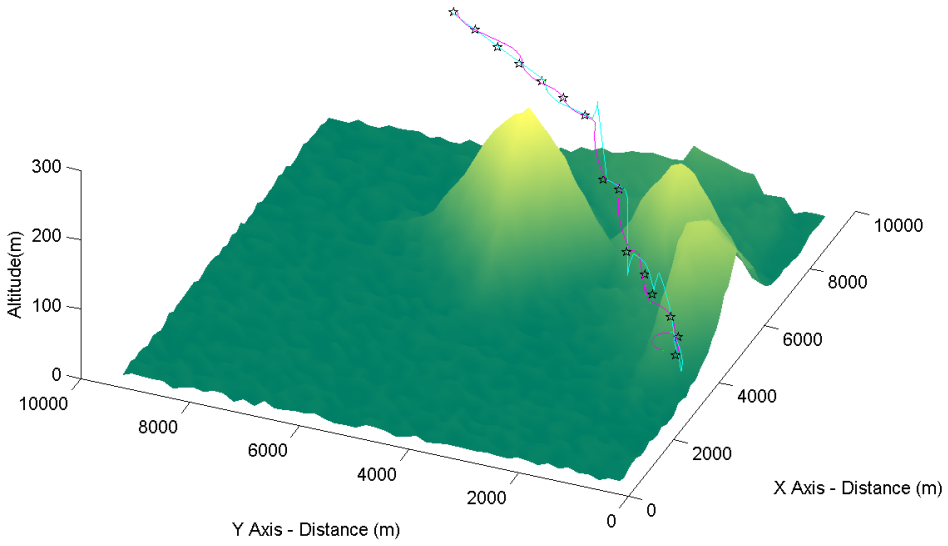


Figure 4.16: Comparing trajectories from LC-2 solution and UTA-4 with HDM 3 dataset (terrain simulation 1)

for $\text{crit}_{(|\Delta\psi|)}$ which leads to the selection of manoeuvres which exhibit a low $c_{(|\Delta\psi|)}$ (4.3). This can result in the selection of primitives on the edge of the platforms

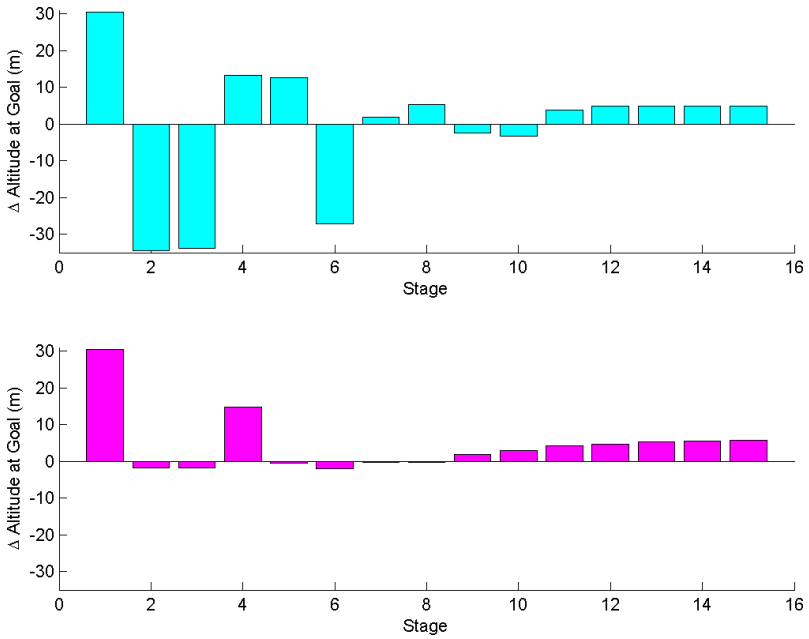


Figure 4.17: Comparing UAS platform ΔAltitude at goal for LC-2 solution and UTA-4 with HDM 3 dataset (terrain simulation 1)

wing loading performance bounds as LC-2 does not explicitly consider $\text{crit}_{(|\Delta\phi|)}$ during optimisation. This can be viewed in Figure 4.19 where LC-2 exhibits higher maximum ϕ values than UTA-4. See Appendix D.2 for additional plots comparing UA trajectories generated for LC-2 and UTA-4 solutions for terrain simulation 2.

4.4 Discussion

This chapter presented a new approach for the inclusion of human expert cognition into autonomous trajectory planning, for UAS operating in low altitude environments with terrain present. Expert decision data was gathered using a GUI, allowing for the quantification of the human decision making process. Preferences elicited from human decision data were applied using UTA to generate feasible 3D collision free trajectories which better represented candidate HDM priorities and mission objectives.

It has been demonstrated that mission requirements and HDM decision styles

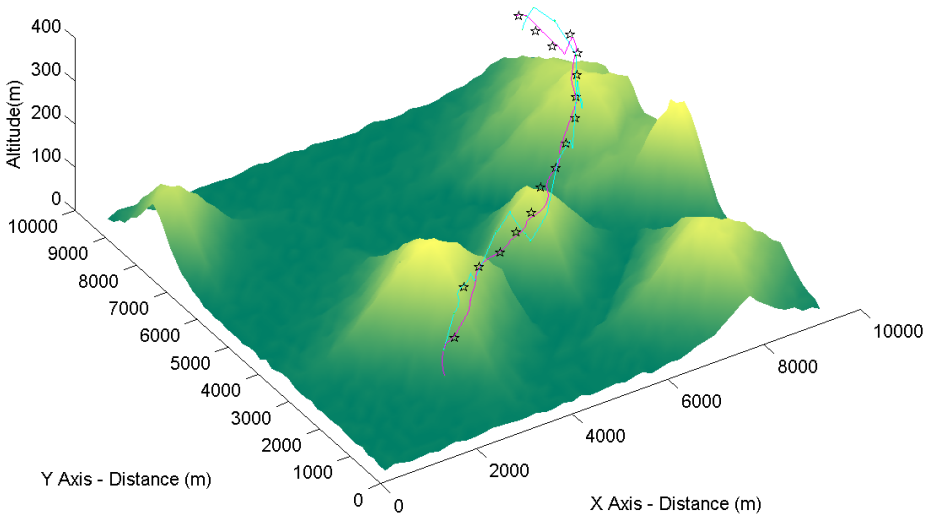


Figure 4.18: Comparing Trajectories from LC-2 solution and UTA-4 with HDM 2 Dataset (terrain simulation 2)

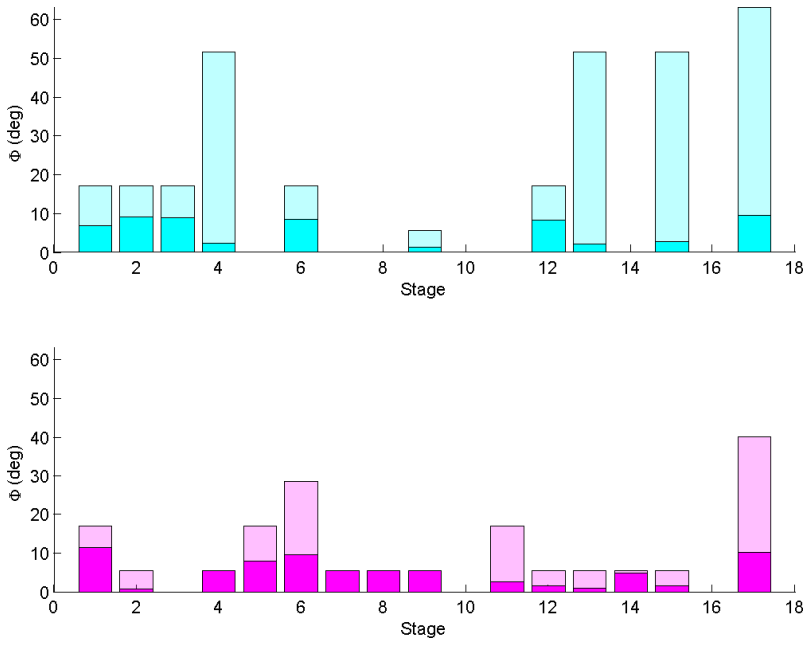


Figure 4.19: Comparing UAS Platform Mean and Maximum ϕ (per stage) for LC-2 solution and UTA-4 with HDM 2 Dataset (terrain simulation 2)

can be better represented in automated trajectory planning systems through the inclusion of HDM decision data through the UTA MCDA technique. Using automated decision algorithms which apply human expert decision strategies may result in increased confidence in UAS operations over populated regions and potentially bring civilian UAS closer to being operated autonomously in the NAS. During low altitude operations however, the environment may present several challenges not encountered in high altitude flight. Due to the potentially limited distances between objects, UAS may only have a limited decision window to generate and perform the appropriate manoeuvres for successful obstacle avoidance during online planning. Thus, real-time planning constraints may be imposed on the multi-objective trajectory planning process due to the existence of obstacles in the immediate path. The following chapter investigates the consideration of real-time planning constraints during low altitude operations in partially known low altitude environments.

5

Computationally Adaptive Real-Time Trajectory Planning

Conducting autonomous UAS operations in low altitude cluttered environments may present several challenges not encountered during high altitude flight. Terrain and urban structures become hazards to the safety of the UAS Thus, low altitude UAS operations in proximity of obstacles may place real-time constraints on the automated trajectory planner onboard.

The computational complexities of applying the DP optimisation strategy to MA based trajectory planning is discussed in Section 2.3. Hybrid architectures [49; 53; 92; 102] where DP is limited to optimisation over a single stage have been applied to decrease overall computation times. Furthermore, Kuwata [10] and Frazzoli [9] have demonstrated that it's possible to take real-time constraints into account during trajectory generation using geometric planning techniques.

Whilst Frazzoli [9] was able to generate trajectories in real-time using MA theory through the inclusion of a minimum trim execution time (τ), he stated that future work is still required for more efficient use of the available decision window. Frazzoli states *"The selection of the trajectory primitives is currently done manually: it would be desirable to obtain formal criteria defining the optimal choice of primitives, trading off the complexity of the resulting automaton with the achievable performance. A dynamic resizing of the automaton is also conceivable: in critical situations, when a decision has to be taken in a very short*

time, the automaton could be reduced to a few manoeuvres, whereas in a more secure situation the set of possible manoeuvres could be expanded."

Dynamically adjusting automaton size during planning could allow for more efficient use of finite planning window and onboard computational resources. For example, using dynamically adjustable automations can allow for real-time re-planning on two platforms with different computational capabilities. Albeit the platform with greater computational capability is expected to be able to compute a solution using a larger set of manoeuvres.

It is important to note that the onboard computational capabilities and flight performance of individual UA can vary. For a given platform with a predefined computational capability and V (assuming the trajectory planner is deterministic), it is expected that the computation times for each trajectory segment or stage is dependent on the automaton resolution (A). The following section investigates the relationship between A, platform computational capability and trajectory computation times to demonstrate the presence of real-time constraints during online trajectory planning in low altitude partially known environments.

5.1 Presence of real-time deadlines during MA trajectory planning

In the presence of real time deadlines, there is a finite length of time available (Finite Planning Window) for the UAS to complete the trajectory solution search before a predefined safety manoeuvre must be executed to ensure collision free flight (Figure 5.1). Convergence to a solution, if one exists, within this Finite Planning Window (FPW) is dependent on current system execution parameters such as automaton resolution and computational power available.

Whilst the computational capabilities of the system are dependent on the Central Processing Unit (CPU) power and the efficiency of the algorithm, if the algorithm developed is deterministic in nature, it is expected to generate a solution in the same or similar period of time for a predefined A resolution. The following section provides an overview of how the simulated 3D low altitude urban

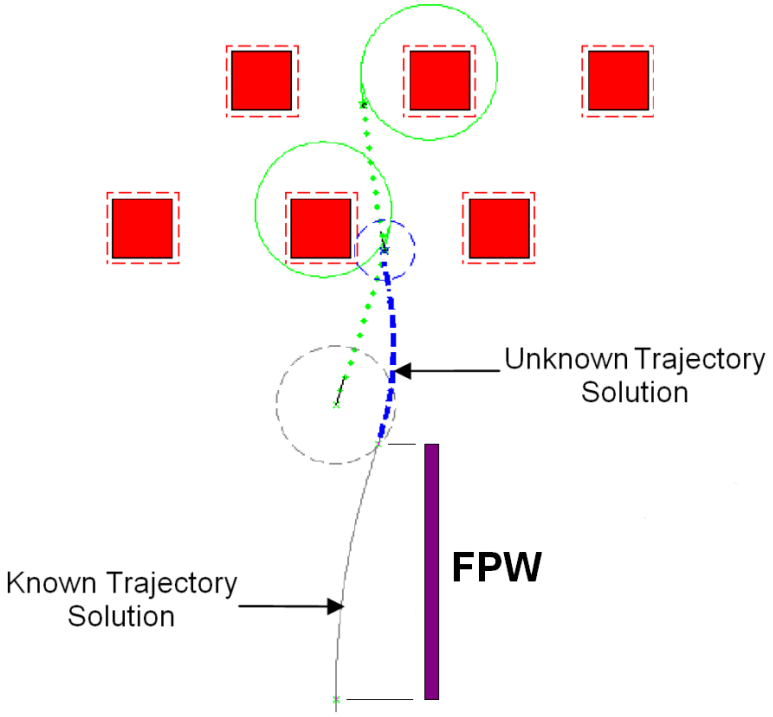


Figure 5.1: FPW during single stage MA based trajectory planning in cluttered environment

environment is setup to perform an investigation on the real-time constraints present during MA based trajectory planning.

5.1.1 Simulation setup

Simulated 3D environments were setup in MATLAB to represent low altitude urban terrain (Figure 5.2). The UAS platform is tasked with generating an optimised feasible collision free trajectory through a set of predefined waypoints, where each waypoint is represented by a spherical capture region and desired direction at goal (Figure 5.3).

The main system execution parameter is the automaton size or total number of alternatives (A) generated using MA theory. A is dependent on the platform ϕ, θ, and p resolution selected. ϕ and θ resolution indicates the sampling distance (in degrees) between trim manoeuvres (Table 5.1). Figure 5.4 presents a visual comparison between smaller ($1350A$) and larger ($41800A$) fixed wing automatons

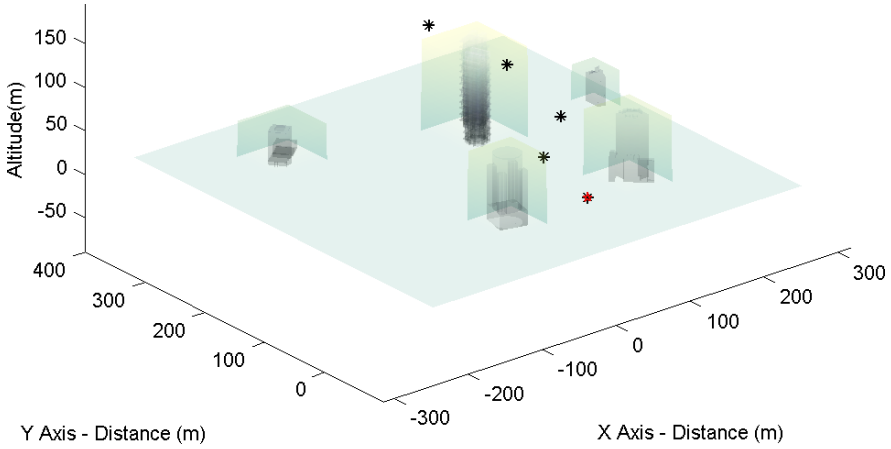


Figure 5.2: Simulated 3D environment representing low altitude urban terrain

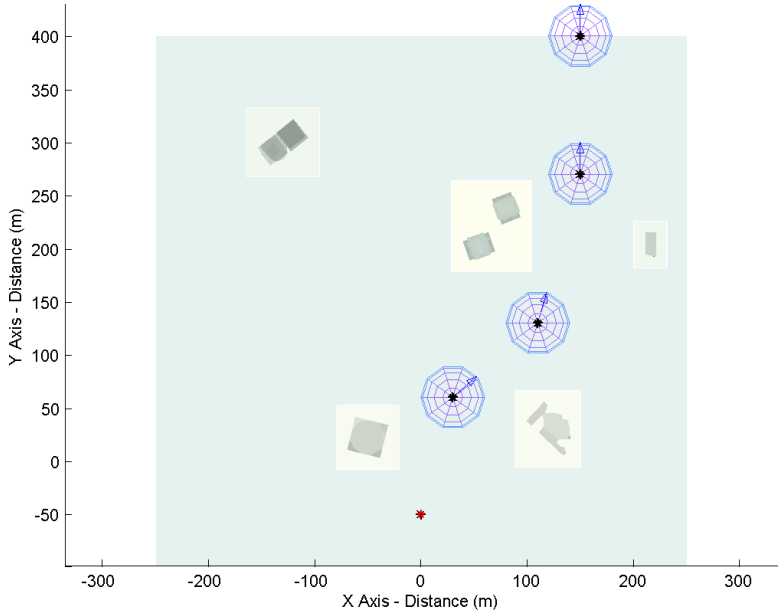


Figure 5.3: Waypoint capture represented as spherical regions

generated using the parameters listed in Table 5.1.

The simulations are conducted in partially known environments where the known environment abstraction is represented as a finite horizon scheme. During, planning in partially known environments, the planner has limited knowledge of the planning environment. This places a limit on the length of the primitives which can executed, as platform safety cannot be guaranteed if the selected ma-

A	ϕ **res.** (degrees)	θ **res.** (degrees)	p **res.**
1350	15.74	7.48	50
1950	10.49	7.48	50
5700	6.996	7.48	100
9100	10.49	2.493	100
13300	6.996	2.493	100
41800	6.996	1.49	200

Table 5.1: A applied during simulation

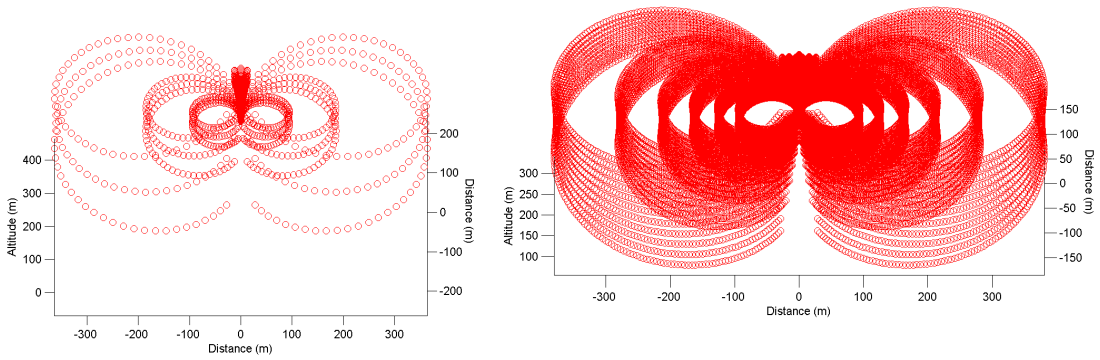


Figure 5.4: Comparison between smaller ($1350A$) and larger($41800A$) fixed wing automatons

noeuvre state is outside the environment abstraction available. It is expected that a shorter finite horizon window would require a platform with greater computational capabilities to compute a feasible solution in real-time.

During online planning, the FPW and finite horizon must be updated to reflect on the current platform state after the computation of every stage. The following section presents an overview of the FPW and finite horizon formulation applied to this research.

5.1.2 FPW during online planning in partially known environments

During single stage optimisation, the FPW has been defined as the maximum time available to the trajectory planner to compute the next known trajectory

segment to avoid entering a hold pattern (Figure 5.1). A cruise manoeuvre is executed at the start of the simulation with a predefined FPW length to provide the planner with an initial FPW to compute the first trajectory segment.

This research project applies a spherical region of radius R_{sphere} centered at the current platform position to represent the known part of the operational environment. The planner does not have access to environment abstraction outside of the sphere, therefore all trajectories which do not allow for the execution of a safe state manoeuvre within the known region are not considered during the decision making component (Figure 3.19). The initial FPW must be set to ensure that the platform can compute a feasible trajectory segment solution within the spherical region within the available decision window (5.1).

$$FPW_{initial} < \frac{R_{sphere}}{v_{initial}} \tag{5.1}$$

After the computation of the initial trajectory segment, the FPW is updated to reflect the time available to the planner to generate the next segment. However, during the computation of the next trajectory segment the finite horizon remains static. In order to have the most up to date environment abstraction, the known region is centred (C_{sphere}) at the start of the trajectory segment ($x0_k \{x, y, z\}$) currently being traversed. Therefore the FPW (5.2) and C_{sphere} (5.3) for the next trajectory segment (p_{k+1}) are defined as:

$$FPW_{k+1} = t_{(m)_k} + t_{(p)_k} \tag{5.2}$$

$$C_{(sphere)_{k+1}} = (x0 \{x, y, z\})_k \tag{5.3}$$

where $t_{(m)_k}$ and $t_{(p)_k}$ are the durations of the manoeuvre and trim primitives selected for the current stage k respectively.

5.1.3 Presence of real-time constraints during simulation

This section presents the planning time required to compute a feasible trajectory through a predefined set of waypoints (Figure 5.3) for a given set of system execution parameters. The simulations have been performed on System 1 (Core

A	**Segment Compute Time [mean (σ)]** (s)	**Min. FPW** (s)	**Time to Plan** (s)	**Flight Time** (s)
1350	0.567 (0.011)	1.160	3.97	20.35
1950	0.811 (0.010)	0.868	6.48	29.62
5700	2.203 (0.018)	1.030	15.42	20.19
9100	3.492 (0.016)	1.010	20.95	19.70
13300	5.214 (0.032)	1.030	36.5	20.19
41800	16.08 (0.076)	1.089	112.5	20.34

Table 5.2: Computation results for system 1 where $R_{sphere} = 250$m

A	**Segment Compute Time [mean (σ)]** (s)	**Min. FPW** (s)	**Time to Plan** (s)	**Flight Time** (s)
1350	1.174 (0.024)	1.160	8.22	20.35
1950	1.652 (0.026)	0.868	13.2	29.62
5700	4.491 (0.024)	1.030	31.4	20.19
9100	7.157 (0.068)	1.010	42.9	19.70
13300	10.41 (0.071)	1.030	72.8	20.19
41800	32.05 (0.109)	1.089	224.32	20.34

Table 5.3: Computation results for system 2 where $R_{sphere} = 250$m

2 duo at 3GHz) and System 2 (Core 2 duo simulated at 1.5GHz) to highlight the planners performance on platforms with different computational capabilities. The initial FPW was set to five seconds to provide the planner with a predefined period of time to find a valid solution for the next trajectory segment (if one existed).

In order to test the trajectory planners performance to compute a feasible trajectory in partially known environments, the simulations have been performed on system 1 and 2 respectively using an R_{sphere} of 250 metres. Table 5.2 and 5.3 present the planning time required to compute a feasible trajectory through a partially known environment on systems 1 and 2 respectively.

For the trajectory planner to successfully compute a feasible safe solution in real-time, the segment compute time must be less than the minimum FPW during the simulation (shortest trajectory segment length). For example, the

5700 A simulation can compute most trajectory segments within the available FPW, however it cannot compute a full solution in real time as the computation time required to generate segments two, three and six is greater than the FPW available (Figure 5.5).

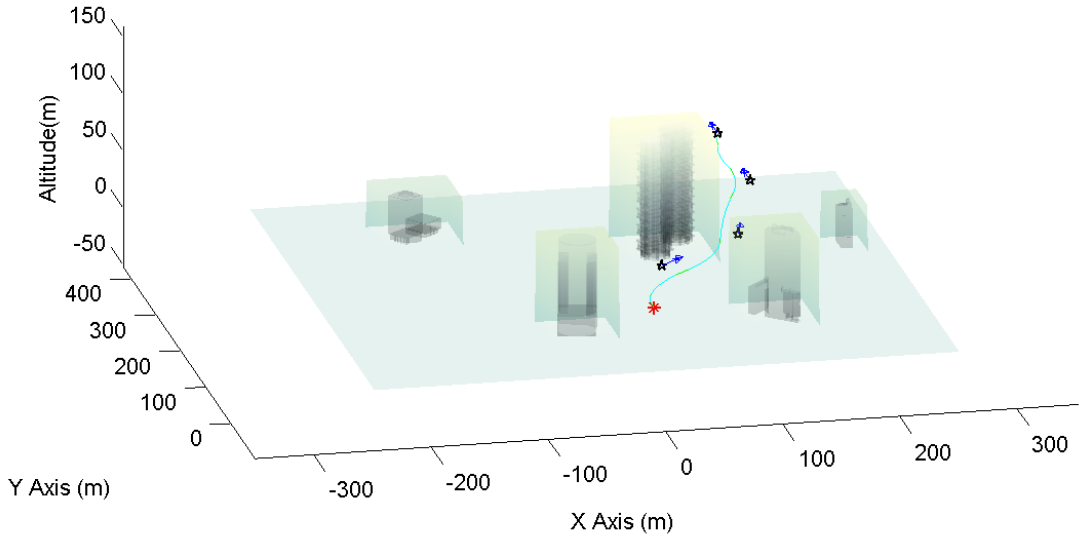


Figure 5.5: Simulated 3D trajectory generated using 5700A parameters

5.1.4 Summary of findings

The ability to generate an MA based trajectory solution in real-time was found to be dependent on the computational power available, the A resolution applied and p length for each segment.

For a fixed resolution of A, the MA algorithm computed the same trajectory solution on both computing platforms System 1 and System 2; this was expected as the algorithm is deterministic. It was also found that planning at higher A resolutions increased the time to compute for each segment (Figure 5.8). Therefore, due to the increased computational capabilities, System 1 (Table 5.2) was able to compute the solutions in real-time for a higher A resolution than possible with System 2 (Table 5.3).

The p length for each segment is dependent on the selection of the most optimal trim primitive for each segment. The optimal trim primitive selected

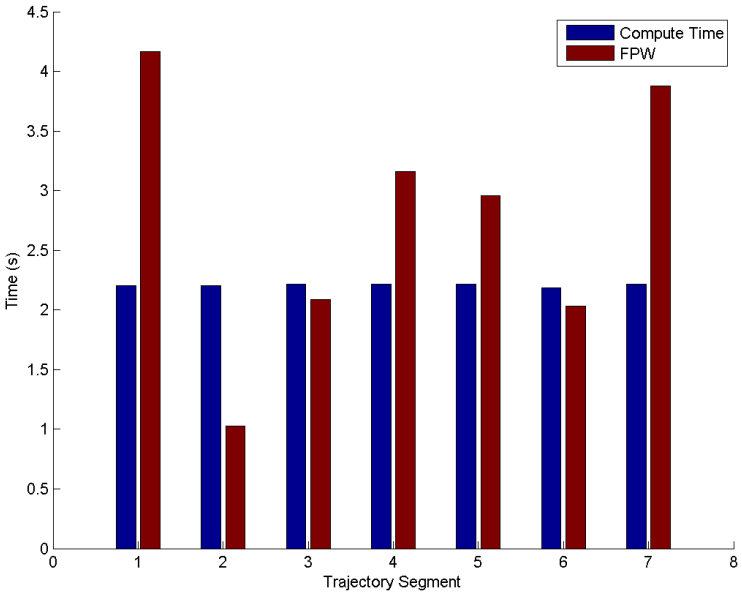


Figure 5.6: Comparison of segment computation time and FPW for 5700A simulation on system 1

may vary with the number of feasible A available and the optimisation method applied during manoeuvre selection. Furthermore, the p length of the current segment affects the FPW available for future stages. This makes it difficult to predict if the simulation using fixed parameters will compute a solution in real time as the p length is not expected to remain constant during planning.

Using fixed parameters, the only way to currently ensure that the MA based trajectory planner will compute a solution in real-time is perform planning using a low A resolution, even on platforms with higher computational capabilities. The use of a low A resolution however, potentially decreases the optimality and completeness of the segment solution. Additionally, the use of low A can potentially reduce the manoeuvrability of the aircraft or decrease planning precision as a lower number of primitives are available for selection.

The use of a higher resolution A set may provide a decision making component with a larger set of potentially more optimal A to select from, thus increasing planning precision and completeness. However, it must be noted that onboard computational capabilities can vary between UAS platforms due to the numer-

ous types of hardware available. Furthermore, smaller UAS platforms may be restricted in computational capability due to more stringent payload size, weight and power restrictions.

The author proposes a new approach where the automaton size is dynamically adjusted during planning for more efficient use of finite planning window and onboard computational resources. The Computationally Adaptive Trajectory Decision optimisation System (CATDS) is expected to increase planning precision and completeness in safe scenarios, whilst still ensuring that future trajectory segments can still be computed in real-time in time constrained situations. The following section details the development of the new approach which allows for the computation and optimisation of feasible 3D flight trajectories within real time planning deadlines.

5.2 Development of CATDS optimisation system

In the presence of real time deadlines, there is a finite length of time available, for the UAS to complete the trajectory solution search before a predefined safety manoeuvre (Section 3.5) must be executed to ensure safe autonomous operations. Convergence to a solution, if one exists, within this FPW is dependent on the A resolution and computational power available onboard.

Scenarios may exist where a feasible solution cannot be generated within the FPW if the A resolution is too high. Consequently, segment completeness and planning precision may be diminished if the A resolution is too low. This section presents a new approach which allows for the computation and optimisation of feasible 3D flight trajectories within real time planning deadlines, for UAS operating in partially known environments with obstacles present.

A novel Computationally Adaptive Trajectory Decision optimisation System (CATDS) has been developed and implemented in simulation to dynamically manage, calculate and schedule system execution parameters to ensure that the trajectory solution search can generate a feasible solution, if one exists, within a given FPW. The inclusion of the CATDS potentially increases overall mission

efficiency and may allow for the implementation of the system on different UAS platforms with varying onboard computational capabilities. This approach has been demonstrated in simulation using a fixed wing UAS operating in low altitude partially known environments with obstacles present.

The CATDS optimisation system is composed of offline and online components. The offline component benchmarks the computational performance of the system using sets of predefined execution parameters. The computational performance of the system can be estimated as the algorithm is deterministic in nature. It must be noted that the offline component requires re-execution if the computation capabilities of the system are updated. The online component dynamically selects the optimum A resolution with respect to the available computational power and FPW for the each trajectory segment. The following sections detail the development of the offline and online components of the CATDS.

5.2.1 CATDS offline component

The offline component of the CATDS calculates the average time to generate a primitive set (for a given ϕ, θ and p resolution), determine the A subset to ensure platform safety (Section 3.5) and apply MCDA techniques to select an optimised A for a single stage. This information provides the online component with a matrix of expected times ($t_{(benchmark)}$) to compute an optimised trim primitive for a range of A resolutions.

For this research, multiple p sets of varying A resolutions were executed offline to determine the computational performance of the platform. Table 5.4 gives an overview of the p sets applied during the offline phase.

The benchmark times to compute an offline solution for each unique p set and p resolution combination (Table 5.4) is presented in Figure 5.7. The time to compute an offline solution for a single stage was found to increase relatively linearly with the A resolution applied (Figure 5.8).

The benchmarking data is compiled offline, stored and then applied during the online component to ensure that the following stage can be computed within a specified FPW. The following section presents an overview of the online component of the CATDS system.

A	ϕ **res.** (degrees)	θ **res.** (degrees)	p	p **res.**
[600 900 1200 1500]	31.48	7.48	15	[40 60 80 100]
[1080 1620 2160 2700]	31.48	2.49	27	[40 60 80 100]
[1400 2100 2800 3500]	31.48	1.49	35	[40 60 80 100]
[1560 2340 3120 3900]	15.74	7.48	39	[40 60 80 100]
[2200 3300 4400 5500]	15.74	2.49	55	[40 60 80 100]
[2280 3420 4560 5700]	15.74	1.49	57	[40 60 80 100]
[2520 3780 5040 6300]	10.49	7.48	63	[40 60 80 100]
[3640 5460 7280 9100]	10.49	2.49	91	[40 60 80 100]
[3960 5940 7920 9900]	10.49	1.49	99	[40 60 80 100]
[5320 7980 10640 13300]	6.996	7.48	133	[40 60 80 100]
[5720 8580 11440 14300]	6.996	2.49	143	[40 60 80 100]
[8360 12540 16720 20900]	6.996	1.49	209	[40 60 80 100]

Table 5.4: A resolutions which form matrix of expected times

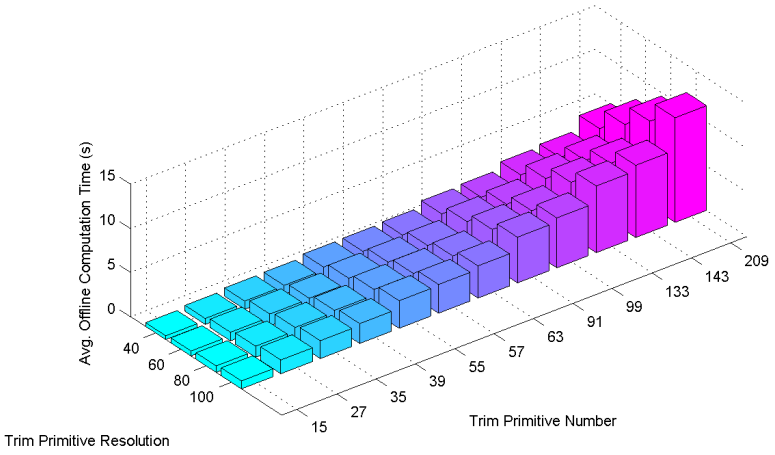


Figure 5.7: Time to compute offline solution with respect to p number and p resolution for system 1

5.2.2 CATDS online component

The online component of the CATDS takes into account the decision scenario and onboard computational capabilities of the UAS platform, to select the most appropriate execution parameters which ensure that the MA based trajectory planner can compute an optimised solution in real-time. In time constrained scenarios (shorter FPW) the CATDS decreases the automaton, enabling the faster solution computation. In scenarios where a longer FPW window is available, the

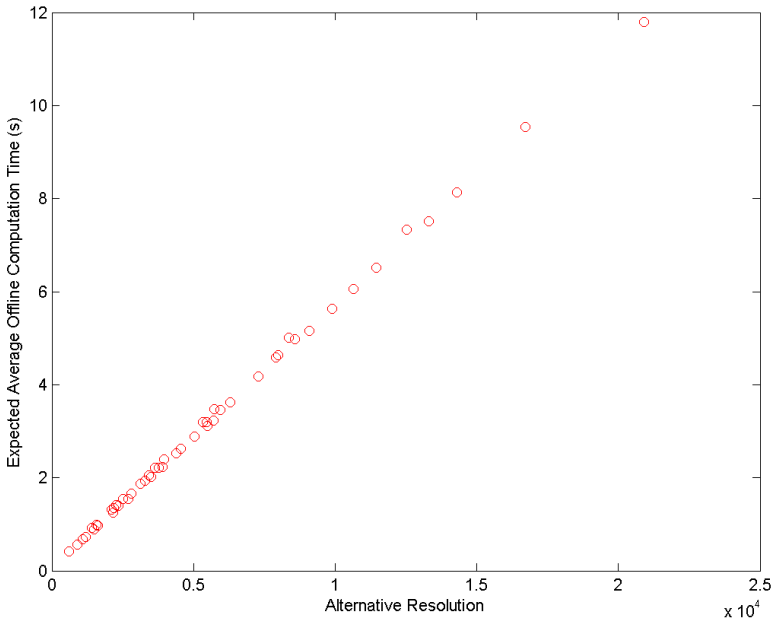


Figure 5.8: Time to compute offline solution with respect to A resolution for system 1

CATDS expands the p set to allow for the computation of a suitable trajectory solution for the current stage at a higher resolution.

In order to ensure that the MA based trajectory planner can compute a solution in real-time, several constraints are imposed by the CATDS. The minimum execution time for the following trajectory segment (p_{k+1}) must be long enough to ensure that the trajectory planner has sufficient time to compute the next trajectory segment. Thus, the minimum trajectory execution time $t_{p_{min}}$ cannot be less than the minimum time from the matrix of expected times computed by the offline benchmarking component ($t_{(benchmark)}$) (5.4).

$$t_{p_{min}} = \min\left(t_{(benchmark)}\right) \tag{5.4}$$

Without any additional constraints present, the CATDS essentially selects the highest A resolution which allows for the computation of the following trajectory segment within the available FPW (5.5).

$$t_p = \max\left(t_{(benchmark)} < FPW\right) \tag{5.5}$$

System	A Res. [Min Max]	Min. Segment Compute Time (s)	Min. FPW (s)	Time to Plan (s)	Flight Time (s)
1	[2280 10640]	1	1.066	18.672	21.183
2	[1400 5040]	1.218	1.268	15.842	20.231

Table 5.5: Computation results for real-time MA based planning with CATDS enabled

To validate the performance of the CATDS, online trajectory planning was conducted with the CATDS enabled in simulated 3D low altitude urban terrain (Figure 5.2) where $R_{sphere} = 250$m . The results are presented in Table 5.5 for systems 1 and 2.

The CATDS dynamically adjusts the segment A resolution (Figure 5.9) to ensure that each trajectory segment is computed within the available FPW (Table 5.5). Thus, enabling the CATDS allows for more efficient use of the available FPW during online planning (Figure 5.10). For systems with lower computational capabilities (e.g. system 2), the CATDS selects lower automaton A resolutions to ensure real-time planning constraints are still met.

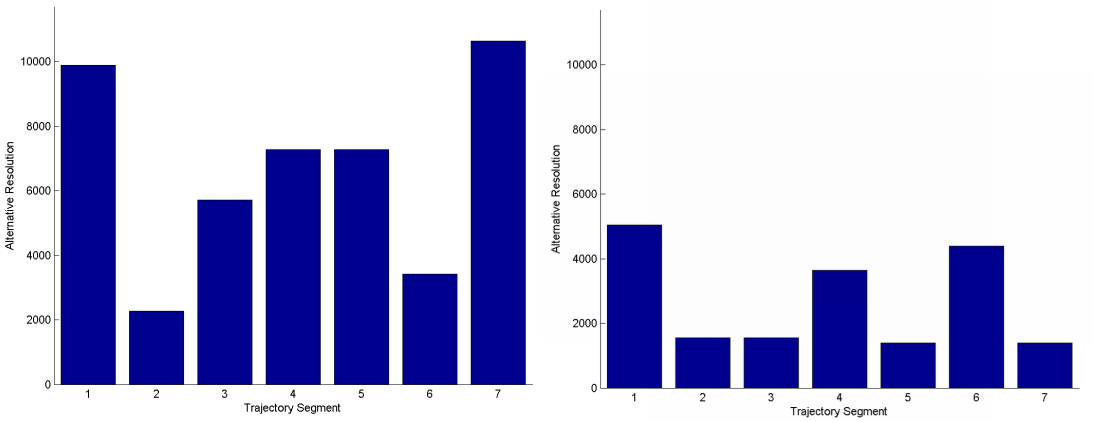


Figure 5.9: Segment A resolution with CATDS enabled where $R_{sphere} = 250$m on System 1 and 2 respectively

Franke [4] states that with increasing levels autonomy onboard UAS operators move away from direct control of the platform towards a management by exception control paradigm. Management by exception occurs when the UAS

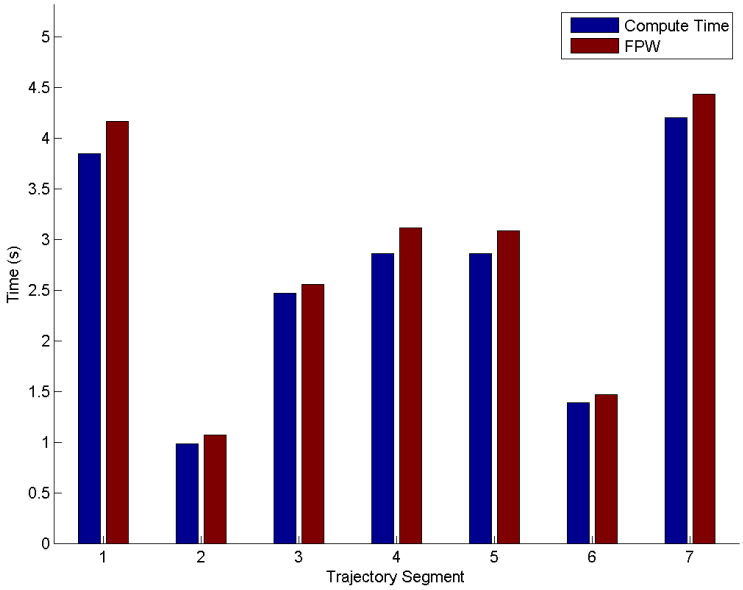


Figure 5.10: Comparison of trajectory compute time and FPW with CATDS enabled where $R_{sphere} = 250$ on System 1

performs planning and execution and informs the HDM of its current and future actions. The operator has the option to veto or override the current plans and revert to a lower control paradigm if required.

In order to apply a management by exception control paradigm to this research project requires that the HDM has sufficient time (FPW_{min}) to veto the current decision. Without consideration for future stages, HDM may have insufficient time to veto the current trajectory segment being traversed if $t_{p_{min}}$ is less than FPW_{min}. The following section presents the inclusion of the FPW_{min} constraint during real-time MA based trajectory planning.

5.2.3 Applying a minimum FPW

The inclusion of the FPW_{min} constraint within the CATDS can be achieved by either applying a minimum p length constraint or starting the computation for the following stage before the UAS has traversed the current trajectory segment. Applying a minimum p length constraint reduces the number of feasible alternatives available to the planner and may force the selection of sub-optimal

FPW_{min} (s)	A **Res. [Min Max]**	**Min. Segment Compute Time** (s)	**Min. FPW** (s)	**Time to Plan** (s)	**Flight Time** (s)
1	[600 5320]	0.53	1	14.22	19.75
2	[600 3420]	0.53	2	8.5	19.91

Table 5.6: Computation results with CATDS enabled and FPW_{min} present for System 1

trajectory segments. Thus, a FPW_{min} was applied by to this research project by starting the computation for next segment p_{k+1} before current trajectory segment had been completely traversed.

The CATDS uses a buffer (t_{BUFFER}) to ensure that enough time is available during computation of the current segment to generate the following stage with an FPW of FPW_{min} (5.6). This results in more conservative use of the available FPW during planning to ensure that the FPW_{min} constraint can be met. It must be noted, the trajectory planner must also account for the change in the location of the finite horizon centre (5.7).

$$\begin{aligned} t_{BUFFER} &= FPW_{min} - t_{p_{min}} \\ t_p &= \max\left(t_{(benchmark)} < (FPW - t_{BUFFER})\right) \end{aligned} \tag{5.6}$$

$$\begin{aligned} t_{sphere} &= FPW_{min} - t_{p_{k+1}} \\ C_{(sphere)_{k+1}} &= \left(p\left\{x, y, z, t_{sphere}\right\}\right)_k \end{aligned} \tag{5.7}$$

This research applies a FPW_{min} to represent the time available to a decision maker to veto the current trajectory segment being traversed before the next segment is computed and executed. Table 5.6 presents the computation results with the CATDS enabled and the inclusion of the FPW_{min} constraint of 1 and 2 seconds.

The inclusion of an FPW_{min} constraint whilst the CATDS system was enabled ensured that the trajectory planner took $t_{p_{k+1}}$ into account to provide the supervisory HDM with sufficient time to veto if they did not agree with the current trajectory segment selected. This was achieved by starting the computation

of the following stage in advance if $t_{p_{k+1}}$ was found to be less than FPW_{min}. It was also found that the inclusion of a longer FPW_{min} of 2 seconds (Figure 5.11) required a longer t_{BUFFER}, resulting in a more conservative approach to the allocation of segment computation time by the CATDS (Figure 5.12).

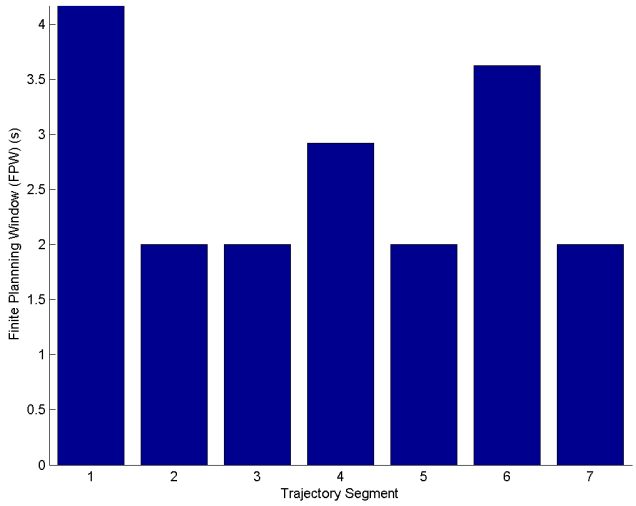


Figure 5.11: Segment FPW with CATDS enabled and FPW_{min} = 2s on system 1

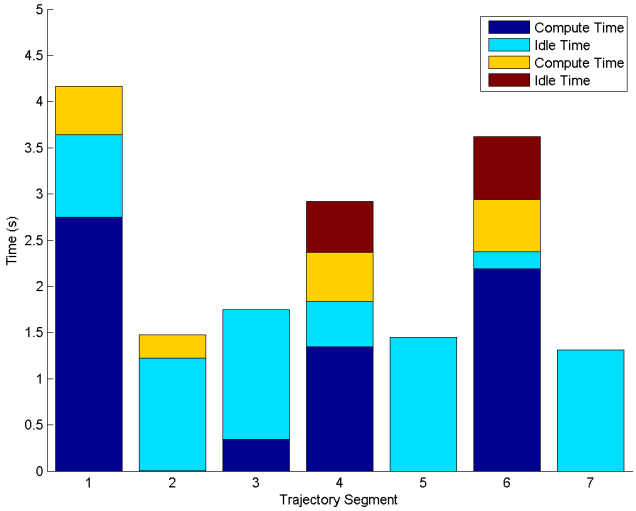


Figure 5.12: Segment computation and idle times with CATDS enabled and FPW_{min} = 2s on system 1

5.2.4 Summary of findings

This section presented a new approach for the real-time generation of trajectories for fixed wing UAS, operating in partially known low altitude environments. A novel CATDS was applied to demonstrate the generation of trajectories in real-time by taking into account onboard computational capabilities. The CATDS dynamically adjusted automaton resolution to ensure that a feasible solution could be found within the available FPW. Additionally, the inclusion of a minimum FPW within the CATDS provided a supervisory HDM a minimum period of time to veto onboard decisions and potentially allow the UA to operate at a higher level of autonomy using a management by exception paradigm [4].

With the inclusion of the CATDS system, it is now possible to apply HDM decision preferences to autonomous UA trajectories during operations in low altitude environments where real-time constraints are present. The inclusion of HDM preferences formulated in Section 4 is expected to generate feasible trajectories which represent aspects of HDM decision styles during real-time planning. Furthermore enabling the CATDS allows for more efficient use of the decision window whilst providing the HDM with sufficient time to veto decisions if required.

The following section compares the autonomous trajectories generated using UTA-4 and LC-2 decision algorithms to highlight how the inclusion of HDM decision data within UTA-4 allows for the generation of trajectories which better represent HDM and mission preferences during real-time trajectory planning.

5.3 Inclusion of HDM preferences during real-time planning

Mission waypoints representing the desired path from the current position to goal were input into the simulated 3D environment (Figure 5.13). The UA is tasked with formulating an optimised feasible collision free trajectory through the predefined waypoint set in real-time.

The following sections apply and compare LC-2 and UTA-4 decision algorithms to demonstrate how HDM decision priorities can be applied to autonomous

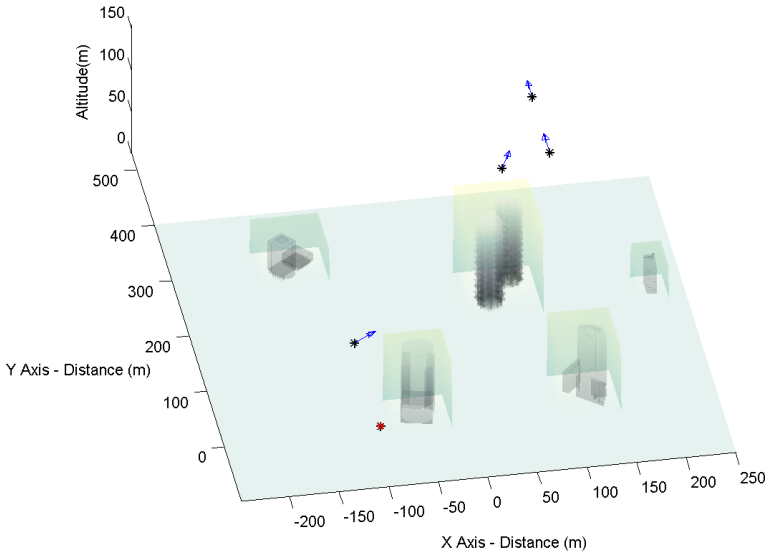


Figure 5.13: Waypoints representing desired path to goal

Decision Algorithm	A **Res. [Min Max]**	**Min. Segment Compute Time** (s)	**Min. FPW** (s)	**Time to Plan** (s)	**Flight Time** (s)
LC-2	[600 20900]	0.281	1	52.048	69.07
UTA-4 (HDM 2)	[600 20900]	0.265	1	57.562	111.3
UTA-4 (HDM 3)	[600 20900]	0.265	1	60.984	108.4

Table 5.7: Computation results with CATDS enabled and FPW_{min} = 1s on System 1

trajectory planning in the presence of real-time constraints through the application of UTA theory.

5.3.1 Results

Real-time planning simulations were performed by enabling the CATDS and applying a FPW_{min} of 1s and R_{sphere} of 250m on system 1. Table 5.7 presents the results for LC-2 and UTA-4 decision algorithms.

LC-2 generated a safe and feasible trajectory where the UA performed a helical spiral ascent [89] to avoid become trapped in local minima (Figure 5.14). For resulting trajectory segments where $t_{p_{k+1}} < 1$ second, the onboard planner began

computations before the current trajectory had been completely traversed, thus allowing for $FPW_{min} = 1$ second to be maintained (Figure 5.15).

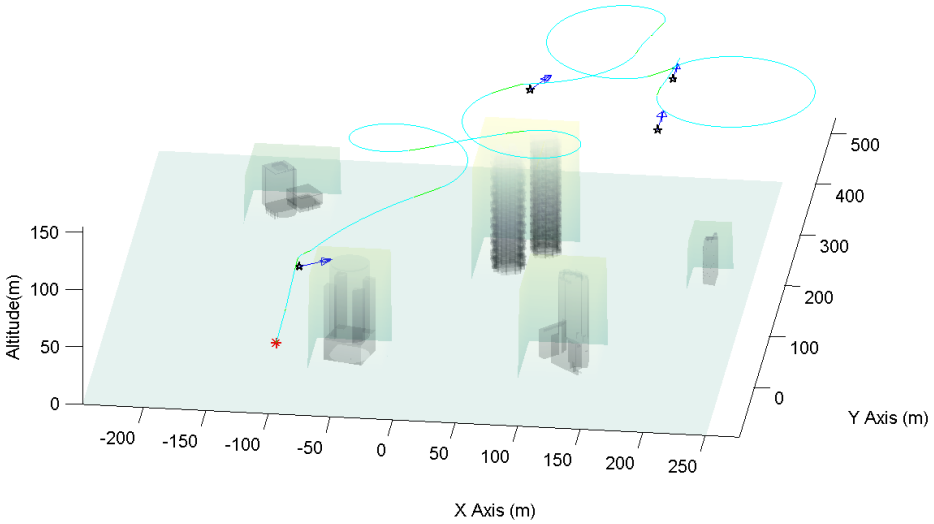


Figure 5.14: Trajectory generated by LC-2 solution with CATDS enabled

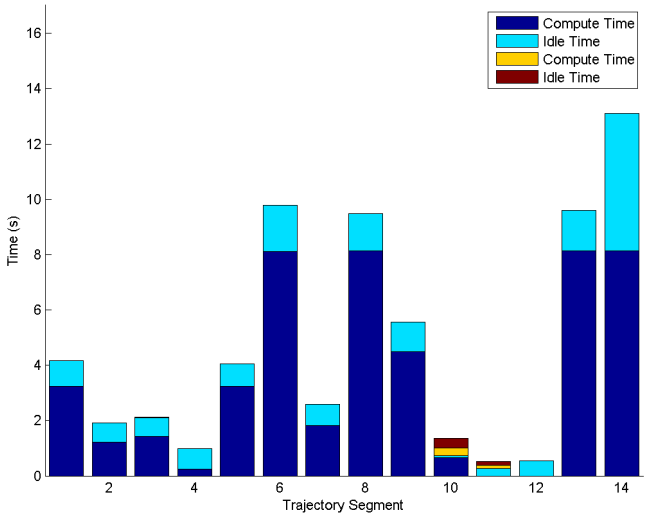


Figure 5.15: Segment computation and idle times with CATDS enabled and $FPW_{min} = 1$s on system 1 for LC-2 solution

LC-2 does not take into account HDM decision preferences, and therefore may not be representative of HDM and mission priorities. The following sections

further analyse the resulting trajectories generated by applying HDM decision preferences through UTA theory via UTA-4 decision algorithm in comparison to the LC-2 trajectory generated (Figure 5.14).

5.3.1.1 Inclusion of HDM 2 Data via UTA-4

Section 4.2.3 demonstrated that UTA-4 represented HDM priorities during primitive selection with greater accuracy than LC-2. HDM 2 was found to place a higher preference on $criterion_{(|\phi|)}$ (Figure 4.12). Whilst the platform cannot execute manoeuvres outside of its predefined ϕ bounds, consideration of $criterion_{(|\phi|)}$ can potentially reduce platform instability due to the execution of sharp turns during flight [106]. Figure 5.16 presents the 3D trajectory generated in real-time using HDM 2 decision data via UTA-4 by enabling the CATDS.

The inclusion of HDM 2 decision data within UTA-4 generated trajectories exhibited lower average ϕ than LC-2, even in low altitude urban environments (Figure 5.17). This demonstrates the inclusion of HDM 2's decision data allows the ADS to place a higher preference on the selection of primitives which better meet the candidate HDMs decision style.

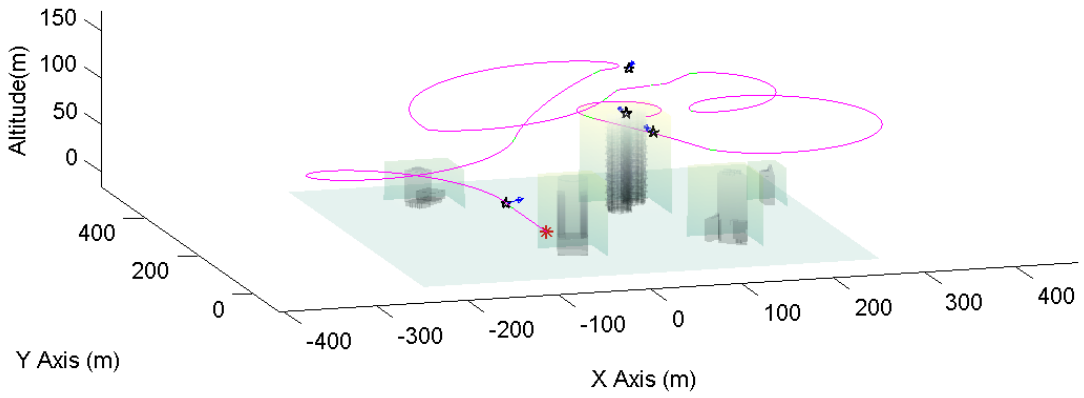


Figure 5.16: Trajectory generated by UTA-4 solution with CATDS enabled using HDM 2 dataset

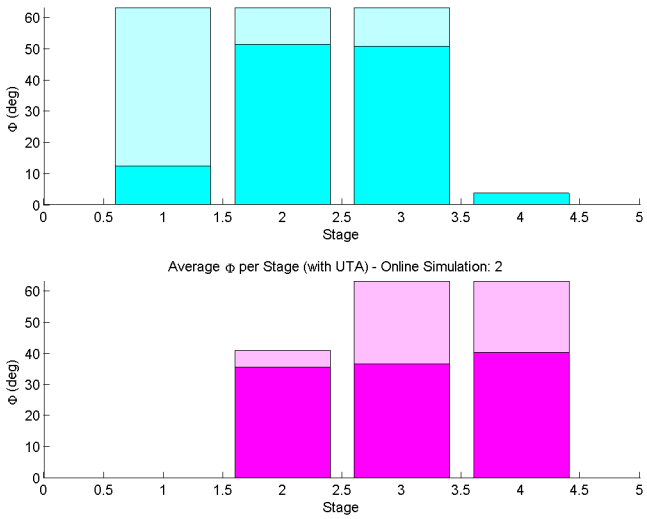


Figure 5.17: Comparing UAS Platform Mean and Maximum ϕ (per stage) for LC-2 solution and UTA-4 with HDM 2 Dataset

5.3.1.2 Inclusion of HDM 3 Data via UTA-4

It was found that HDM 3 placed a higher preference on $criterion_{(|g_z - s_z|)}$ (Figure 4.13) during the selection of primitives. This was also demonstrated through simulated online trajectory planning in long range low altitude environments (Section 4.3.3.1).

The inclusion of HDM 3 decision data within UTA-4 generated trajectories where the relative altitude difference between the UA and desired waypoint altitude once the waypoint had been reached, was lower than LC-2 (Figure 5.19). A HDM's preference for maintaining a similar altitude to the goal can be beneficial for certain missions such as surveillance or airborne sensing and capture. Figure 5.18 presents the 3D trajectory generated in real-time using HDM 3 decision data via UTA-4 by enabling the CATDS.

5.4 Discussion

This chapter presented a new approach for the real-time generation of trajectories for fixed wing UAS, operating autonomously in partially known low altitude

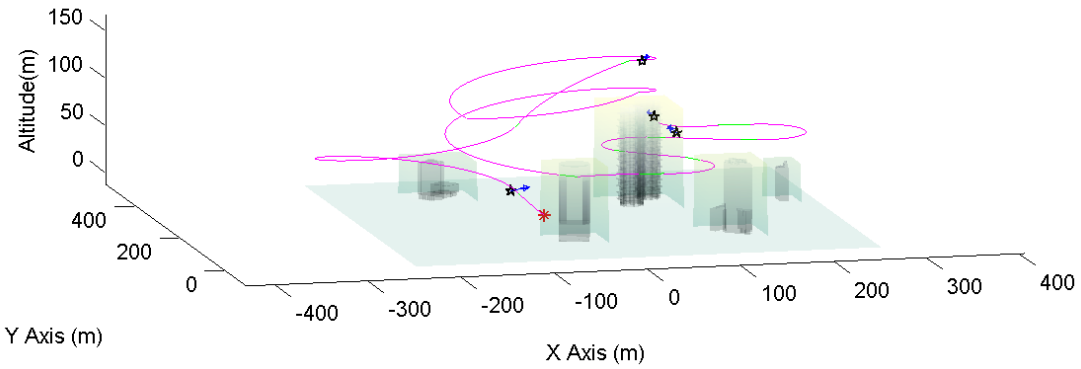


Figure 5.18: Trajectory generated by UTA-4 solution with CATDS enabled using HDM 3 dataset

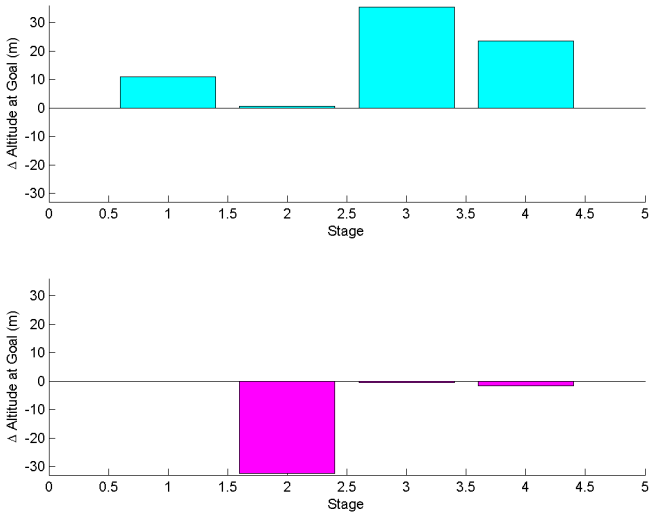


Figure 5.19: Comparing UAS Platform relative Δ altitude at goal for LC-2 solution and UTA-4 with HDM 3 Dataset

environments. A novel Computationally Adaptive Trajectory Decision optimisation System (CATDS) was applied for the generation of flight trajectories in the presence of real-time planning deadlines. The CATDS dynamically adjusted automaton resolution to ensure that a feasible solution could be found within the available FPW. This allowed for more efficient use of the available decision window and onboard computational resources.

The capability for a UA to operate autonomously in a safe manner was demon-

strated through the application of safe states to 3D partially known environments, where UA enters a hold manoeuvre if the HDM vetoes the current trajectory segments generated. This allows the supervisory HDM to operate at a higher level of autonomy using a management by exception paradigm [4]. Furthermore, the inclusion of a minimum FPW within the CATDS provides the HDM with a minimum period of time to veto UA ADS decisions during planning.

Enabling the CATDS allows for the inclusion of HDM decision preferences during autonomous UAS trajectory planning in low altitude partially known environments in the presence of real time constraints. HDM preferences formulated in Section 4 were applied via UTA-4 to generate trajectories which took candidate HDM priorities into account whilst meeting real-time planning constraints. This was demonstrated by comparing trajectories generated by UTA-4 and LC-2 in simulated partially known urban terrain.

The following chapter presents the conclusions of this research project.

6

Conclusions

This thesis presented the conceptualisation and implementation of new algorithms for embedding human expert cognition and real-time trajectory planning on autonomous UAS.

6.1 Thesis summary

In chapter 1, the problem of integrating UAS within the NAS was formulated, where a need for greater onboard autonomy to meet the ELOS requirement was established. Highly autonomous low altitude UAS operations in the presence of terrain and obstacles was highlighted as being a difficult challenge, but would provide UAS with the capability to perform a greater range of low altitude civilian missions in a safe manner. The challenges present during autonomous low altitude operations included; incorporation of complex platform dynamics, guarantee of platform safety, optimisation with respect to mission or HDM requirements and real-time planning constraints were also discussed.

Literature regarding intelligent control architectures and trajectory planning methods was presented in chapter 2, and discussed with respect to meeting autonomous low altitude operational challenges listed in the previous chapter. An overview of the candidate trajectory generation method, MA theory, was presented and its potential for the inclusion of UAS dynamic constraints for accurate platform tracking whilst taking into account real-time constraints was highlighted.

In chapter 3, MA theory was applied to generate feasible flight trajectories for fixed wing UA. Fixed wing trim states such as coordinated turns and helical flight modes were simulated and included to form an automaton of common fixed wing trim primitives. Fixed wing platform roll and pitch rate constraints were modelled and applied through the inclusion of manoeuvre primitives which linked trim primitives to form feasible flight trajectories. The application of safe state manoeuvres (finite receding horizon model) with consideration for platform attitude rates is extended to 3D partially known environments to guarantee platform safety during UAS operations. A review of decision making algorithms to optimise the collision free feasible trajectories based on MA theory is also presented.

The application of MCDA methodologies to better model HDM mission priorities was investigated in chapter 4. An overview of the MCDA process is presented in context of low altitude trajectory planning in partially known environments. Human expert decision data was captured via a GUI, allowing for the quantification of the candidate HDM's decision making process. Preferences were elicited from HDM decision data using UTA theory and applied to a multi-criteria aggregation technique to generate trajectory segments which encapsulated aspects of the candidate HDM's decision strategies. Trajectories computed using different HDM decision data were compared with a least cost solution with equal preferences for all criteria. This demonstrated that the unique decision styles of individual HDMs can be better represented during automated trajectory planning through the inclusion of HDM decision data via the UTA MCDA technique.

In chapter 5, the safe operation of UAS in low altitude environments with real-time planning constraints was investigated. Trajectory computation times using fixed automaton resolutions on two simulated platforms with different computational constraints was presented to show that real-time constraints can vary with computational capabilities and planning resolution. A flight management system (CATDS) was proposed and implemented to dynamically moderate automaton resolution to compute trajectory segments within real-time deadlines on platforms with different computational capabilities. The flight management system is an extension of the real-time planning research undertaken by Frazzoli [53]. CATDS was able to successfully compute solutions within real-time deadlines for different scenarios including; systems with different computational capability and

different sensor capabilities (represented as finite horizon lengths). The inclusion of CATDS enabled more efficient use of the available decision window and onboard computational resources. Furthermore, the inclusion of a minimum FPW within the CATDS provided a minimum period of time to veto UA ADS decisions. This provides the capability for a HDM to potentially revert to a supervisory role where the UAS operates at a higher level of autonomy (in a management by exception paradigm). Finally, enabling the CATDS allowed for the inclusion of HDM decision preferences during autonomous UAS trajectory planning in low altitude partially known environments in the presence of real time constraints.

Figure 6.1 illustrates the trajectory planning research presented in this thesis.

6.2 Contributions

The aim of this research was to investigate the research questions identified in Section 1.3.1. This thesis has highlighted contributions made in the research field of autonomous trajectory planning through the application of intelligent control and MCDA methodologies to autonomous UAS systems.

By answering the first question, Can trajectory planning be effectively automated for standalone autonomous operations on a UAS platform?, the research resulted in the:

1. Identification and categorisation of three potential areas of research which can potentially embed greater onboard capability within a trajectory planner as enablers for autonomous trajectory planning in the NAS. These potential challenges include; incorporation of complex platform dynamics for the generation of trackable trajectories, improving HDM trust for UAS operating at higher levels of autonomy and inclusion of finite planning deadlines for the generation of trajectories in low altitude partially known environments in real-time.

By answering the second question, Under what conditions can a flight management concept be developed to ensure that the supervisory HDM's mission criteria are successfully met during operations in low altitude environments with real time planning constraints present?, the research resulted in the:

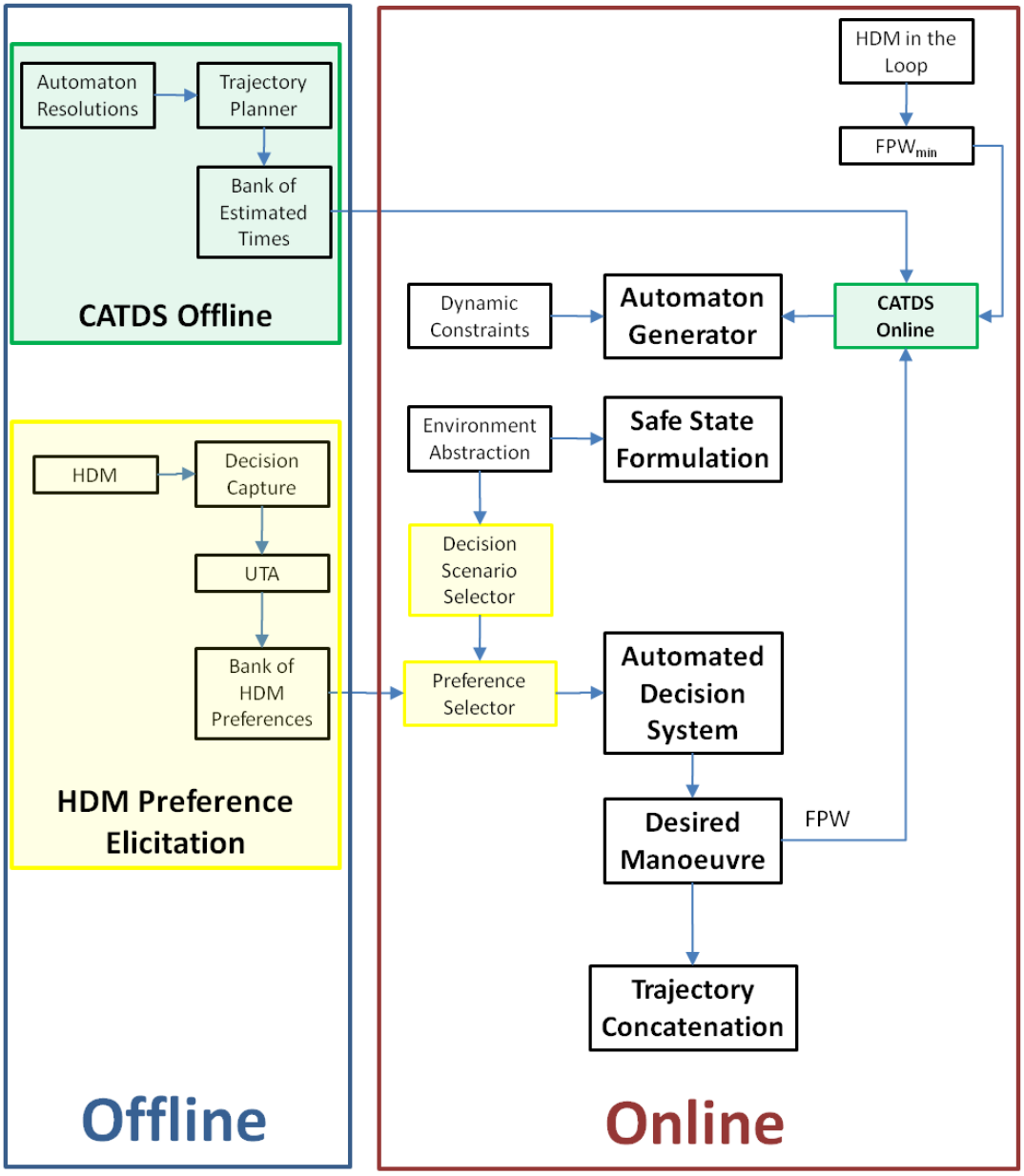


Figure 6.1: Illustration of trajectory planning research presented in this thesis

1. Extension of previous research to improve platform safety during low altitude UAS flight in the NAS. MA theory was applied to fixed wing platforms where attitude rate constraints were considered through the generation of manoeuvre primitives. This allowed for the inclusion of attitude rate con-

straints during the inclusion of implicit safety guarantees, through the application of safe state research in partially known 3D environments.

2. Development of a methodology for elicitation of human expert decision data representing HDM flying styles into mathematical value functions. A GUI was developed and implemented in simulation to interact with HDMs who were required to select the most optimal flight trajectory from a given set of alternatives.

3. Quantification of HDM decisions as mathematically modelled preferences through the application of UTA theory. This research allows for the quantification of HDM or pilot decisions and provides a deeper understanding of the decisions considered by a candidate HDM during UAS operations.

4. Development of a methodology to allow for real-time replanning in environments with variable planning deadlines by dynamically varying automaton size. This approach provides greater flexibility in providing real-time trajectory planning capabilities on platforms with different computational capabilities.

6.3 Future work

The concepts developed in this research project provides scope for future research in various areas. First, this research only considers the feed forward (trajectory generation) component without any external disturbances (e.g. wind). Inclusion of simulated wind models and incorporation of a trajectory tracking layer using a control scheme in future research could allow for more accurate modelling of the trajectory planning process.

Second, the optimisation strategy used, applies single stage optimisation in order to select the most optimal manoeuvre for each stage. Single stage optimisation has the advantage of faster computation and smaller memory footprint over multi-stage optimisation, however it also decreases global optimality. Extending this research to consider multiple stages in a computationally efficient manner would allow the generation of a more globally optimal trajectory.

Third, this research project considered four possible criteria during optimisation. Some candidate HDM's decision profiles were not as accurately modelled; this may lead to the possibility of additional criteria existing which have not been considered. Future research into the formulation of additional criteria may improve the modelling of HDM decisions into mathematical preference functions.

UAS have been employed in a diverse range of military applications to date and numerous UAS market forecasts portray a burgeoning future. The success of UAS operations within military fields has brought about the realisation of the potential of UAS utilisation in the civilian domain. To realise autonomous civilian UAS operations, seamless operation within the NAS will be required. This research project has investigated the development of a trajectory planning solution to provide UAS with the capability to operate with greater autonomy during low altitude operations in the NAS. Enabling autonomous low altitude UAS operations can allow for civilian missions to be undertaken in a safer manner, whilst reducing HDM workload and continuous communications link reliance.

Bibliography

[1] N. A. S. I. Group, "Airspace for everyone: An introduction to airspace reform," Australian Government, Tech. Rep., Oct 2003. ix, 4, 5

[2] P. Roberts, D. Fitzgerald, C. Wang, and R. Walker, "En-route human pilot automation considerations for future uav systems," Queensland University of Technology, Tech. Rep., 2005. ix, 7

[3] P. Chandler and M. Pachter, "Research issues in autonomous control of tactical uavs," in *Proceedings of the American Control Conference*, vol. 1, 1998, pp. 394–398. ix, 8

[4] J. Franke, V. Zaychik, T. Spura, and E. Alves, "Inverting the operator/vehicle ratio: Approaches to next generation uav command and control," June 2005. ix, 11, 40, 61, 112, 116, 122

[5] S. Tzafestas, *Methods and Applications of Intelligent Control*. Boston: Kluwer Academic, 1997. ix, 23

[6] R. P. Bonasso, D. Kortenkamp, D. P. Miller, and M. G. Slack, "Experiences with an architecture for intelligent, reactive agents," in *International Joint Conference on Artificial Intelligence*, 1995. ix, 22, 23, 24, 25

[7] J. Boskovic, R. Prasanth, and R. Mehra, "A multilayer control architecture for unmanned aerial vehicles," in *Proceedings of the American Control Conference*, U. o. Cincinnati, Ed., vol. 3, 2002, pp. 1825–1830. ix, 25, 26, 27

[8] M. Hwangbo, J. Kuffner, and T. Kanade, "Efficient two-phase 3d motion planning for small fixed-wing uavs," in *IEEE International Conference on Robotics and Automation*, 2007, pp. 1035–1041. ix, 30

[9] E. Frazzoli, M. Dahleh, and E. Feron, "Real-time motion planning for agile autonomous vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 1, 2002. ix, 31, 36, 37, 39, 41, 99

[10] Y. Kuwata, "Real-time trajectory design for unmanned aerial vehicles using receding horizon control," Master of Science, 2003. ix, 30, 32, 33, 36, 37, 39, 45, 50, 99

[11] S. Scherer, S. Singh, L. Chamberlain, and S. Saripalli, "Flying fast and low among obstacles," in *IEEE International Conference on Robotics and Automation*, 2007. ix, 34, 35, 37

[12] R. Beard, D. Kingston, M. Quigley, D. Snyder, R. Christiansen, W. Johnson, T. McLain, and M. Goodrich, "Autonomous vehicle technologies for small fixed-wing uavs," *Journal of Aerospace Computing, Information and Communication*, vol. 2, 2005. x, 58

[13] J. Topsakal, "Robust motion planning in the presence of uncertainties using a maneuver automaton," Masters Thesis, 2005. x, 10, 38, 58, 59

[14] Y. Kuwata, "Real-time trajectory design for unmanned aerial vehicles using receding horizon control," Masters Thesis, 2001. x, 59, 60

[15] R. Parasuraman, T. Sheridan, and C. Wickens, "A model for types and levels of human interaction with automation," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 30, no. 3, 2000. xiv, 6

[16] "Unmanned aircraft systems roadmap," Secretary of Defense, Tech. Rep., 2005. 1

[17] "Department of defense dictionary of military and associated terms," 2001. 1

[18] K. Dalamagkidis, V. K. P., and L. A. Piegl, *On Integrating Unmanned Aircraft Systems into the National Airspace System.* Springer Science, 2009. 1, 2, 5

[19] "A feasibility study using intelligent software agent technology for weapon-target pairing, utilizing unmanned aerial vehicles," U.S. Army Aviation and Missile Command Acquisition Cente, Tech. Rep., Dec 30 2008. 1

[20] P. Schaefer, R. Colgren, R. Abbott, H. Park, A. Fijany, F. Fisher, M. James, S. Chien, R. Mackey, M. Zak, T. Johnson, and S. Bush, "Technologies for reliable autonomous control (trac) of uavs," in *Digital Avionics Systems Conference*, vol. 1, 2000. 1, 3, 25

[21] "Unmanned aircraft systems roadmap 2007 - 2035," Secretary of Defense, Tech. Rep., Dec 10 2007. 1

[22] T. Cox, "Civil uav capability assessment (draft version)," NASA, Tech. Rep., 2004. 1

[23] S. J. Zaloga, D. Rockwell, and P. Finnegan, "World unmanned aerial vehicle system: Market profile and forecast 2011 edition," Teal Group, Tech. Rep., 2011. 1, 2

[24] A. Kochan, "Automation in the sky," *Industrial Robot: An International Journal*, vol. 32, no. 6, pp. 468 – 471, 2005. 2

[25] K. Wong, C. Bil, G. Gordon, P. Gibbens, and ., "Study of the unmanned aerial vehicle (uav) market in australia," August 1997. 2

[26] S. Wegener, "Uav autonomous operations for airborne science missions," American Institute of Aeronautics and Astronautics, Tech. Rep., 2004. 2, 10, 13

[27] A. Ollero and I. Maza, *Multiple Heterogeneous Unmanned Aerial Vehicles.* Berlin Heidelberg: Springer, 2007. 2, 13

[28] C. A. A. of Australia, "Advisory circular - uav operations, design specification, maintenance and training of human resources," Tech. Rep., July 2002. 2

[29] EUROCONTROL, "Specifications for the use of military unmanned aerial vehicles as operational air traffic outside segregated airspace," Tech. Rep., 25 April 2006. 2

[30] M. T. DeGarmo, "Issues concerning integration of unmanned aerial vehicles in civil airspace," MITRE, Center for Advanced Aviation System Development, Tech. Rep., 2004. 3, 22

[31] U. S. of Defense, "Unmanned aerial vehicles roadmap," Secretary of Defense, Tech. Rep., December 2002. 3

[32] A. S. A. T. M. Group, "Cross industry business case and cost-benefit analysis: Ads-b avionics fitment, introduction of tso 145/146 navigators and extended surveillance coverage," ASTRA, Tech. Rep., June 2005. 3

[33] C. A. A. o. Australia, "Discussion paper: Carriage and use of ads-b avionics," CASA, Tech. Rep. DP 0310AS, December 2004. 4

[34] ——, "Ads-b: Automatic dependant surveillance broadcast," CASA, Tech. Rep., March 2006. 4

[35] G. K. Crosby, D. K. Kraus, W. S. Ely, T. P. Cashin, K. W. McPherson, K. W. Bean, J. M. Stewart, and B. D. Elrod, "A ground - based regional augmentation system (gras) the australian proposal," in *Institute of Navigation GPS*, 2000. 4

[36] T. Murphy and R. Hartman, "The use of gbas ground facilities in a regional network," in *IEEE Position Location and Navigation Symposium,*, 2000, pp. 514–521. 4

[37] "Airspace integration plan for unmanned aviation," Office of the Secretary of Defense, Tech. Rep., Dec 2004. 5

[38] R. Baumeister, R. Estkowski, G. T. Spence, and R. A. Clothier, "Test architecture for prototyping automated dynamic airspace control," 26-29 October 2009. 5

[39] B. T. Clough, "Metrics, schmetrics! how the heck do you determine a uavs autonomy anyway?" in *Performance Metrics for Intelligent Systems Workshop*, 2002. 5, 6

[40] H.-M. Huang, K. Pavek, B. Novak, J. Albus, and E. Messina, "A framework for autonomy levels for unmanned systems (alfus)," in *AUVSI unmanned Systems North America*. 6

[41] M. L. Cummings and P. J. Mitchell, "Predicting operator capacity in remote supervision of multiple unmanned vehicles," *IEEE Systems, Man, and Cybernetics, Part A Systems and Human*, vol. 38, no. 2, pp. 451–460, 2008. 6

[42] P. Narayan, P. Wu, D. Campbell, and R. Walker, "An intelligent control architecture for unmanned aerial systems (uas) in the national airspace system (nas)," in *2nd International Unmanned Air Vehicle Systems Conference*, 2007. 7, 27

[43] S. M. LaValle, *Planning Algorithms*. New York: Cambridge University Press, 2006. 9, 27, 30, 31, 40, 51, 66, 68, 72

[44] R. Stengel, *Flight Dynamics*. Princeton University Press, 2004. 10

[45] B. L. Stevens and F. L. Lewis, *Aircraft control and simulation*. John Wiley and Sons, 2003. 10

[46] J. Figueira, S. Greco, and M. Ehrgott, *Multi Criteria Decision Analysis: State of the Art Surveys*. Boston: Springer, 2005. 11, 71, 77, 78

[47] P. Meyer, "Progressive methods in multiple criteria decision analysis," Ph.D. dissertation, 2007. 11

[48] E. Frazzoli, M. Dahleh, and E. Feron, "Maneuver-based motion planning for nonlinear systems with symmetries," *IEEE Transactions on Robotics and Automation*, vol. 21, no. 6, pp. 1077–1091, 2005. 15, 20, 30, 31, 37, 38, 39, 42, 57, 65, 66

[49] T. Schouwenaars, B. Mettler, E. Feron, and J. How, "Robust motion planning using a maneuver automaton with built-in uncertainties," in *AIAA Aerospace Sciences and Exhibit*, 2003. 15, 18, 31, 41, 65, 66, 69, 99

[50] E. Jacquet-Lagreze and J. Siskos, "Assessing a set of additive utility functions for multicriteria decision-making, the uta method," *European Journal of Operational Research*, vol. 10, pp. 151–164, 1982. 16, 17, 76, 77, 79, 149

[51] P. C. Fishburn, *Utility Theory for Decision Making*. New York: John Wiley and Sons, Inc., 1970. 17, 71

[52] R. Keeney and H. Raiffa, *Decision with multiple objectives: Preferences and value tradeoffs*. Cambridge University Press, 1976. 17, 71

[53] E. Frazzoli, M. Dahleh, and E. Feron, "A hybrid control architecture for aggressive maneuvering of autonomous helicopters," in *Proceedings of the 38th IEEE Conference on Decision and Control*, vol. 3, 1999, pp. 2471–2476. 17, 31, 39, 42, 65, 66, 67, 69, 72, 74, 99, 124

[54] M. Freed, R. P. Bonasso, M. Dalal, W. Fitzgerald, C. Frost, and R. Harris, "An architecture for intelligent management of aerial observation missions," *AIAA*, vol. 6938, 2005. 24, 25

[55] N. Muscettola, P. P. Nayak, B. Pell, and B. C. Williams, "Remote agent: to boldly go where no ai system has gone before," *Artificial Intelligence*, vol. 103, no. 1-2, pp. 5–47, 1998. 25

[56] L. Wills, S. Kannan, S. Sander, M. Guler, B. Heck, J. Prasad, D. Schrage, and G. Vachtsevanos, "An open platform for reconfigurable control," *IEEE Control Systems Magazine*, vol. 21, no. 3, pp. 49–64, 2001. 25

[57] G. Rabideau, "Mission operations with autonomy: A preliminary report for earth observing-1," in *4th International Workshop on Planning and Scheduling for Space*, 2004. 25

[58] P. Doherty, P. Haslum, F. Heintz, T. Merz, P. Nyblom, T. Persson, and B. Wingman, "A distributed architecture for autonomous unmanned aerial vehicle experimentation," in *7th International Symposium on Distributed Autonomous Systems*, 2004. 25

[59] R. Arkin, "Motor schema based navigation for a mobile robot: An approach to programming by behavior," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 4, 1987, pp. 264–271. 25

[60] M. Gupta and N. Sinha, *Intelligent Control Systems - Theory and Applications*. New York: IEEE Press, 1996. 26

[61] L. Cork, R. Walker, and S. Dunn, "Fault detection, identification and accommodation techniques for unmanned airborne vehicles," in *Australian International Aerospace Congress*, 2005. 26

[62] S.-M. Li, J. Boskovic, S. Seereeram, R. Prasanth, J. Amin, R. Mehra, R. Beard, and T. McLain, "Autonomous hierarchical control of multiple unmanned combat air vehicles (ucavs)," in *Proceedings of the American Control Conference*, vol. 1, 2002, pp. 274–279. 27

[63] J.-C. Latombe, *Robot Motion Planning*. Massachusetts: Kluwer Academic Publishers, 1991. 27

[64] I. K. Nikolos, K. P. Valavanis, N. C. Tsourveloudis, and A. N. Kostaras, "Evolutionary algorithm based offline/online path planner for uav navigation," *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics*, vol. 33, no. 6, pp. 898–912, 2003. 28, 29

[65] K. B. Judd and T. W. McLain, "Spline based path planning for unmanned air vehicles," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, vol. AIAA-2001-4238, 2001. 28

[66] B. Cao, G. Dodds, and G. Irwin, "Constrained time-efficient and smooth cubic spline trajectory generation for industrial robots," *IEEE Control Theory and Applications*, vol. 144, no. 5, pp. 467–475, 1997. 28

[67] H. Ozaki and C.-J. Lin, "Optimal b-spline joint trajectory generation for collision-free movements of a manipulator under dynamic constraints," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 4, 1996, pp. 3592–3597. 28

[68] L. Singh and J. Fuller, "Trajectory generation for a uav in urban terrain, using nonlinear mpc," in *Proceedings of the American Control Conference*, vol. 3, 2001, pp. 2301–2308. 28, 33

[69] D. Rathbun, S. Kragelund, A. Pongpunwattana, and B. Capozzi, "An evolution based path planning algorithm for autonomous motion of a uav through uncertain environments," in *Proceedings of the 21st Digital Avionics Systems Conference*, vol. 2. IEEE, 2002, pp. 8D2–1–12. 28, 33

[70] E. Koyuncu, N. K. Ure, and G. Inalhan, "Integration of path/maneuver planning in complex environments for agile maneuvering ucavs," *Journal of Intelligent Robot Systems (2010) :*, no. 57, p. 143170, 2010. 29, 33

[71] L. Dubins, "On curves of minimal length with a constraint on average curvature with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, pp. 497 – 516, 1957. 29, 30, 36, 45, 50, 51, 65

[72] N. D. Richards, M. Sharma, and D. G. Ward, "A hybrid a*/automaton approach to on-line path planning with obstacle avoidance," in *AIAA 1st Intelligent Systems Technical Conference*, vol. 1, 2004, pp. 141–157. 30, 41

[73] K. Savla, E. Frazzoli, and F. Bullo, "Traveling salesperson problems for the dubins vehicle," *IEEE Trans. on Automatic Control*, 2007. 30

[74] E. Anderson, R. Beard, and T. McLain, "Real-time dynamic trajectory smoothing for unmanned air vehicles," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 3, pp. 471–477, 2005. 30, 39, 50

[75] H. Chitsaz and S. LaValle, "Time-optimal paths for a dubins airplane," 12-14 Dec 2007. 30

[76] E. Frazzoli, M. Dahleh, and E. Feron, "Robust hybrid control for autonomous vehicle motion planning," in *Proceedings of the 39th IEEE Conference on Decision and Control*, vol. 1, 2000, pp. 821–826. 31

[77] ——, "Real-time motion planning for agile autonomous vehicles," in *Proceedings of the American Control Conference*, vol. 1. IEEE, 2001, pp. 43–49. 31

[78] D. Hsu, L. Kavraki, J. Latombe, R. Motwani, and S. Sorkin, "On finding narrow passages with probabilistic roadmap planners," in *Workshop on Algorithmic Foundations of Robotics*, 1998. 31

[79] T. Schouwenaars, J. How, and E. Feron, "Receding horizon path planning with implicit safety guarantees," in *Proceedings of the American Control Conference*, vol. 6. IEEE, 2004, pp. 5576–5581. 31, 32, 36, 60, 61, 68

[80] T. Schouwenaars, B. Mettler, E. Feron, and J. How, "Hybrid model for trajectory planning of agile autonomous vehicles," *Journal of Aerospace Computing, Information and Communication*, vol. 1, 2004. 31, 32, 36

[81] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959. 32

[82] P. Doherty, G. Granlund, K. Kuchcinski, E. Sandewall, K. Nordberg, E. Skarman, and J. Wiklund, "The witas unmanned aerial vehicle project," in *14th European Conference on Artificial Intelligence*, W. Horn, Ed. IOS Press, 2000. 33

[83] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, 1985, pp. 500–505, tY - CONF. 34

[84] M. G. Park, J. H. Jeon, and M. C. Lee, "Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing," in *Proceedings of the IEEE International Symposium on Industrial Electronics*, vol. 3, 2001, pp. 1530–1535, tY - CONF. 34

[85] D. Shim, H. Chung, and S. Sastry, "Conflict-free navigation in unknown urban environments," *IEEE Robotics and Automation Magazine*, 2006. 34, 35, 37, 61

[86] S. Griffiths, J. Saunders, A. Curtis, B. Barber, T. McLain, and R. Beard, "Maximizing miniature aerial vehicles: Obstacle and terrain avoidance for mavs," *IEEE Robotics and Automation Magazine*, 2006. 34, 35, 37

[87] T. Koo, F. Hoffmann, H. Shim, B. Sinopoli, S. Sastry, D. Georges, O. Sename, L. Dugard, and C. Canudas de Wit, "Hybrid control of an autonomous helicopter," in *Proceedings from Workshop Motion Control Conference*, D. Georges, O. Sename, L. Dugard, and C. Canudas de Wit, Eds. Elsevier Sci IFAC, 1999, pp. 265–270. 35

[88] T. Schouwenaars, M. Valenti, E. Feron, and J. How, "Implementation and flight test results of milp-based uav guidance," in *IEEE Aerospace Conference*, 2005, pp. 1–13. 36, 37

[89] D. Anisi, J. Robinson, and P. gren, "On-line trajectory planning for aerial vehicles: A safe approach with guaranteed task completion," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*. AIAA, 2006. 36, 49, 61, 68, 117

[90] E. Gagnon, C. Rabbath, and M. Lauzon, "Heading and position receding horizon control for trajectory generation," in *Proceedings of the American Control Conference*, 2005, pp. 134–139. 39

[91] C. L. Bottasso, D. Leonello, and B. Savini, "Path planning for autonomous vehicles by trajectory smoothing using motion primitives," *IEEE Transactions on Control Systems Technology*, 2007. 39, 41, 51, 58

[92] L. Singh, J. Plump, M. McConley, and B. Appleby, "Software enabled control: Autonomous agile guidance and control for a uav in partially unknown urban terrain," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2003. 41, 65, 69, 99

[93] C. Dever, M. Bernard, E. Feron, and J. Popovic, "Nonlinear trajectory generation for autonomous vehicles via parameterized maneuver classes," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 29, no. 2, pp. 289–302, 2006. 41

[94] D. Mellinger, N. Michael, and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," 2010. 41

[95] "Airworthiness standards: Normal, utility, acrobatic, and commuter category airplanes," 1996. 46

[96] N. A. V. Piercy, *Aerodynamics*. English University, 1947. 46, 47, 48, 49, 50

[97] F. A. Administration, "Aeronautical information manual," 2010. 48

[98] R. Nelson, *Flight Stability and Automatic Control*. McGraw Hill, 1998. 51, 53, 54

[99] H. Kim, D. Shim, and S. Sastry, "Nonlinear model predictive tracking control for rotorcraft-based unmanned aerial vehicles," in *Proceedings of the American Control Conference*, vol. 5, 2002, pp. 3576–3581. 58

[100] S. Park, J. Deyst, and J. How, "A new nonlinear guidance logic for trajectory tracking," in *Proceedings of the AIAA Guidance Navigation and Control Conference*. American Institute of Aeronautics and Astronautics, 2004, pp. 2004–4900. 58

[101] R. Bellman, "On the theory of dynamic programming." *National Academy of Sciences*, vol. 38, p. 716, 1952. 65

[102] M. McConley, M. Piedmonte, B. Appleby, E. Frazzoli, E. Feron, and M. Dahleh, "Hybrid control for aggressive maneuvering of autonomous aerial vehicles," in *Proceedings of the 19th Digital Avionics Systems Conferences*, vol. 1, 2000. 65, 69, 99

[103] G. Gigerenzer, P. Todd, and A. R. Group, "Simple heuristics that make us smart," Tech. Rep., 1999. 70, 74

[104] B. Roy, "Classement et choix en prsence de points de vue multiples (la mthode electre)," *la Revue d'Informatique et de Recherche Oprationelle (RIRO)*, vol. 8, pp. 57–75, 1968. 71

[105] E. Triantaphyllou, B. Shu, S. N. Sanchez, and T. Ray, "Multi-criteria decision making: An operations research approach," *Encyclopedia of Electrical and Electronics Engineering*, vol. 15, pp. 175–186, 1998. 71

[106] F. A. Administration, *Performance Maneuvers*, 2nd ed. Aviation Supplies and Academics, Inc., 2004, ch. 9. 75, 89, 119

[107] C. Bana e Costa and J. Vansnick, "A theoretical framework for measuring attractiveness by a categorical based evaluation technique (macbeth)," in *XIth Int. Conf. on MultiCriteria Decision Making*, 1994, p. 1524. 76

[108] G. Bous, P. Fortemps, F. Glineur, and M. Pirlot, "Acuta: A novel method for eliciting additive value functions on the basis of holistic preference statements," *European Journal of Operational Research*, vol. 206, no. 2, pp. 435 – 444, 2010. 77

[109] P. P. Wu, D. A. Campbell, and T. Merz, "On-board multi-objective mission planning for unmanned aerial vehicles," in *IEEE Aerospace Conference*, 2009. 92

[110] I. Ashdown, H. Blackford, N. Colford, and F. Else, "Common hmi for uxvs: Design philosophy and design concept," BAE Systems, Tech. Rep., 2010. 144

[111] F. De Crescenzio, G. Miranda, F. Persiani, and T. Bombardi, "A first implementation of an advanced 3d interface to control and supervise uav (uninhabited aerial vehicles) missions," *Presence: Teleoperators and Virtual Environments*, vol. 18, no. 3, 2009. 144

[112] B. J. A. van Marwijk, C. Borst, M. Mulder, M. Mulder, and M. M. van Paassen, "Supporting 4d trajectory revisions on the flight deck: Design of a humanmachine interface," *The International Journal of Aviation Psychology*, vol. 21, no. 1, pp. 35–61, 2011. 144

Appendices

Appendix A

Human Expert Data Capture

This appendix provides additional information regarding the capture of decision data from Human Decision Makers (HDMs).

A.1 HDM Information

HDM decision data for this research was collected from four HDMs. The author did complete the HDM decision capture experiment, but the results are not included as part of the overall results. Table A.1 provides further information regarding HDM occupation and aircraft operational experience.

HDM	**Company**	**Aircraft operational experience**
1	ARCAA	UA trajectory planning experience (ARCAA)
2	ARCAA	Aerodynamics, Aircraft GPS navigational experience (QUT)
3	ARCAA	UA operations experience (Boeing - scan eagle)
4	ARCAA	Private Pilots Licence (PPL), UA operations experience (ARCAA)

Table A.1: HDM aircraft operational experience information

A.2 Graphical User Interface

The Graphical User Interface (GUI) was developed using MATrix LABoratory's (MATLABs) GUI Design Environment (GUIDE). The GUI presents the HDM

with an automaton and cycles through decision scenarios in order to collect HDM decision data (Figure A.1).

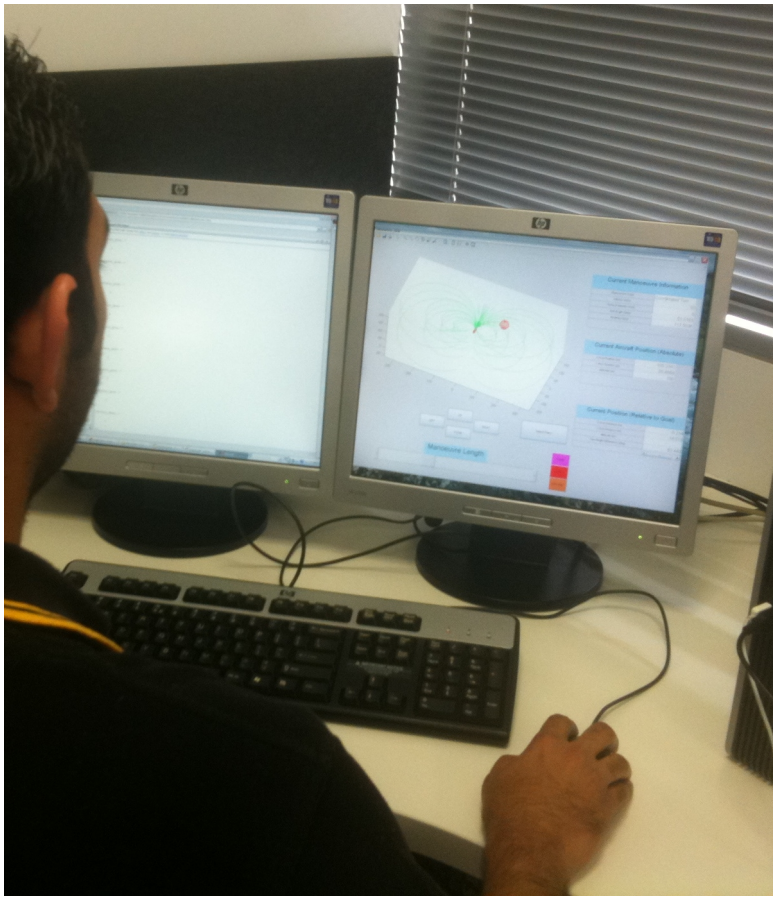


Figure A.1: Data Capture

Whilst research has been conducted into the development of Human Machine Interfaces (HMIs) and Heads Up Displays (HUDs) to improve the supervision and control of UAS [110; 111; 112], no relevant research regarding data capture of HDM decisions for trajectory planning was found. This GUI did not explicitly apply UA HMI development methodologies, rather it was iteratively improved based on HDM feedback during data capture.

The main aim of the GUI is to allow the candidate HDM the ability to efficiently cycle through alternatives so that they can quickly select their desired solution. Desired alternative selection is accomplished by selecting the preferred trim primitive and adjusting the manoeuvre length (if necessary).

A.3 Decision Scenarios

Each Human Decision Maker (HDM) is sequentially presented with decision scenarios and tasked with selecting what they consider to be the most suitable trim primitive to execute for each particular scenario .

A decision scenario is defined as the relative difference between the goal and UA positions ($(x_g - x_p)$, $(y_g - y_p)$, $(z_g - z_p)$) and the relative orientation of the UA with respect to the desired direction at the goal ($(\psi_g - \psi_p)$). In addition, the automaton generated will be unique to the platform roll angle (ϕ_p) due to the inclusion attitude rate constraints; this results in a unique set of A for the HDM to consider. Thus, each unique decision scenario can be represented as ($(x_g - x_p)$, $(y_g - y_p)$, $(z_g - z_p)$, $((\psi_g - \psi_p))$, ϕ_p).

120 decision scenarios were completed by each HDM. Scenarios were separated into five sets to allow HDMs to complete all decision scenarios over several sessions. Table A.2 lists the decision scenarios completed by each HDM during the data capture process.

Scenario Set	**Scenario Number**	$(x_g - x_p)$ (m)	$(y_g - y_p)$ (m)	$(z_g - z_p)$ (m)	$(\psi_g - \psi_p)$ (Bearing°)	ϕ_p (Degrees)
1	1	0	-100	0	270	-30
1	2	100	100	0	180	-30
1	3	-100	-100	0	0	-30
1	4	-100	0	0	315	-30
1	5	100	-100	0	135	-30
1	6	-100	100	0	90	-30
1	7	0	100	0	45	-30
1	8	100	0	0	225	-30
1	9	-100	0	20	135	-30
1	10	100	100	20	90	-30
1	11	0	-100	20	45	-30
1	12	-100	-100	20	180	-30
1	13	100	-100	20	270	-30
1	14	-100	100	20	225	-30
1	15	0	100	20	0	-30
1	16	100	0	20	315	-30
1	17	0	-100	-20	270	-30

1	18	-100	100	-20	180	-30
1	19	100	-100	-20	0	-30
1	20	100	100	-20	135	-30
1	21	-100	-100	-20	45	-30
1	22	0	100	-20	90	-30
1	23	-100	0	-20	225	-30
1	24	100	0	-20	315	-30
2	1	0	-100	0	270	30
2	2	-100	0	0	315	30
2	3	100	-100	0	135	30
2	4	100	100	0	180	30
2	5	0	100	0	45	30
2	6	-100	100	0	90	30
2	7	100	0	0	225	30
2	8	-100	-100	0	0	30
2	9	100	0	20	315	30
2	10	-100	-100	20	180	30
2	11	0	100	20	0	30
2	12	-100	100	20	225	30
2	13	100	-100	20	270	30
2	14	-100	0	20	135	30
2	15	100	100	20	90	30
2	16	0	-100	20	45	30
2	17	-100	100	-20	180	30
2	18	100	-100	-20	0	30
2	19	100	0	-20	315	30
2	20	-100	0	-20	225	30
2	21	-100	-100	-20	45	30
2	22	100	100	-20	135	30
2	23	0	100	-20	90	30
2	24	0	-100	-20	270	30
3	1	0	-100	0	270	0
3	2	100	100	0	180	0
3	3	-100	-100	0	0	0
3	4	-100	0	0	315	0
3	5	100	-100	0	135	0
3	6	-100	100	0	90	0
3	7	0	100	0	45	0

3	8	100	0	0	225	0
3	9	-100	0	20	135	0
3	10	100	100	20	90	0
3	11	0	-100	20	45	0
3	12	-100	-100	20	180	0
3	13	100	-100	20	270	0
3	14	-100	100	20	225	0
3	15	0	100	20	0	0
3	16	100	0	20	315	0
3	17	0	-100	-20	270	0
3	18	-100	100	-20	180	0
3	19	100	-100	-20	0	0
3	20	100	100	-20	135	0
3	21	-100	-100	-20	45	0
3	22	0	100	-20	90	0
3	23	-100	0	-20	225	0
3	24	100	0	-20	315	0
4	1	400	-400	0	135	0
4	2	0	400	0	45	0
4	3	0	-400	0	270	0
4	4	400	0	0	225	0
4	5	-400	0	0	315	0
4	6	-400	-400	0	0	0
4	7	-400	400	0	90	0
4	8	400	400	0	180	0
4	9	0	-400	80	45	0
4	10	-400	400	80	225	0
4	11	400	0	80	315	0
4	12	400	400	80	90	0
4	13	-400	0	80	135	0
4	14	-400	-400	80	180	0
4	15	0	400	80	0	0
4	16	400	-400	80	270	0
4	17	400	0	-80	315	0
4	18	400	400	-80	135	0
4	19	-400	0	-80	225	0
4	20	-400	400	-80	180	0
4	21	400	-400	-80	0	0

4	22	0	400	-80	90	0
4	23	0	-400	-80	270	0
4	24	-400	-400	-80	45	0
5	1	1000	1000	0	180	0
5	2	1000	-1000	0	135	0
5	3	-1000	-1000	0	0	0
5	4	1000	0	0	225	0
5	5	0	1000	0	45	0
5	6	-1000	0	0	315	0
5	7	-1000	1000	0	90	0
5	8	0	-1000	0	270	0
5	9	-1000	0	200	135	0
5	10	-1000	1000	200	225	0
5	11	1000	1000	200	90	0
5	12	-1000	-1000	200	180	0
5	13	0	1000	200	0	0
5	14	1000	0	200	315	0
5	15	1000	-1000	200	270	0
5	16	0	-1000	200	45	0
5	17	-1000	-1000	-200	45	0
5	18	1000	1000	-200	135	0
5	19	-1000	1000	-200	180	0
5	20	-1000	0	-200	225	0
5	21	1000	-1000	-200	0	0
5	22	1000	0	-200	315	0
5	23	0	1000	-200	90	0
5	24	0	-1000	-200	270	0

Table A.2: Computation results with CATDS enabled and FPW_{min} = 1s on System 1

Appendix B

Preference elicitation from HDM decision data

Preference elicitation via UTA [50] is performed through pairwise comparisons of alternatives, where one alternative is given a preference or indifference over the other. As the number of alternatives increases, the number of pairwise comparisons (and subsequently, memory and processing time) required increases exponentially. However, the use of larger automaton or alternative set A provides the HDM with more alternatives from which they can select the alternative which more closely matches their view of the desired solution.

Thus, global preferences were generated using a smaller subset of alternatives (A_R) pseudorandomly selected from (A). A_R allows for faster convergence due to less pairwise comparisons required whilst still providing the HDM with the capability to select the desired alternative from the higher resolution A. Additionally, multiple pseudorandom A_R are generated and compared to ensure that the global preferences are formed using an A_R which accurately represents A. The following section provides an overview of the preference elicitation process applied to this research.

B.1 Alternative subset formulation

The A presented to the HDM contains 516 alternatives from which the human expert selects their desired decision for each decision scenario.

A_R is applied during preference elicitation in order to decrease computation time and reduce the possibility of UTA not finding a valid set of preferences to represent the HDM decisions for each decision scenario. A_R is formed by pseudorandomly selecting 20 alternatives from the complete set of A (including the alternative selected by the HDM). Pseudorandom selection of alternatives was performed using a randomised number generator with a fixed state. A fixed states allows for the selection of the same A_R sets for each HDM.

To overcome the possibility that the downsampled subset does not accurately represent the full set of A, multiple pseudorandom subsets are generated. The following section provides an overview of the selection of global preferences from preferences generated using multiple A_R.

B.2 Selecting global preferences from multiple subsets

To ensure that the global preferences are formed using an A_R which accurately represents A, preferences are generated for multiple A_R and compared to select a global preference set which best represents the given HDM decision. 100 pseudorandom A_R sets are generated and corresponding preferences are formulated using UTA theory.

The preferences formulated from the A_R sets are applied to the Automated Decision System (ADS) (4.8) to generate a bank of trajectory solutions. All trajectory solutions generated from the A_R sets are compared to the HDM decision against the following parameters; roll angle (Δ_ϕ), euclidean position between goal and current state ($\Delta_{|g-s|}$) and platform yaw angle (Δ_{psi}). A least squares method is applied to minimise the sum of the squared residuals and determine the UTA preference which has the best fit. The least squares equation is presented in (B.1) where the A_R resulting in an automated trajectory solution with the best fit is represented as $A_{R(\text{Best Fit})}$.

For $A_R(i)$ where i $\in$ [1..100]

$$\begin{aligned} LSQR(i) &= \min\left(\sqrt{(\Delta_{|g-s|})_i{}^2 + (\Delta_{\phi})_i{}^2 + (\Delta_{psi})_i{}^2}\right) \\ A_R(\text{Best Fit}) &= A_R(\min LSQR) \end{aligned} \tag{B.1}$$

As the number of pseudorandom A_R sets are increased, it is more likely that an A_R subset results in the formulation of preferences which represented HDM decision styles. This convergence can be seen in Figures B.1, B.2, B.3, B.4 where UTA-4 algorithm converges to a trajectory solution which matches closer to the HDM decision. LC-2 remains constant as the least cost formulation doesn't apply preference information.

B.2.1 HDM 1

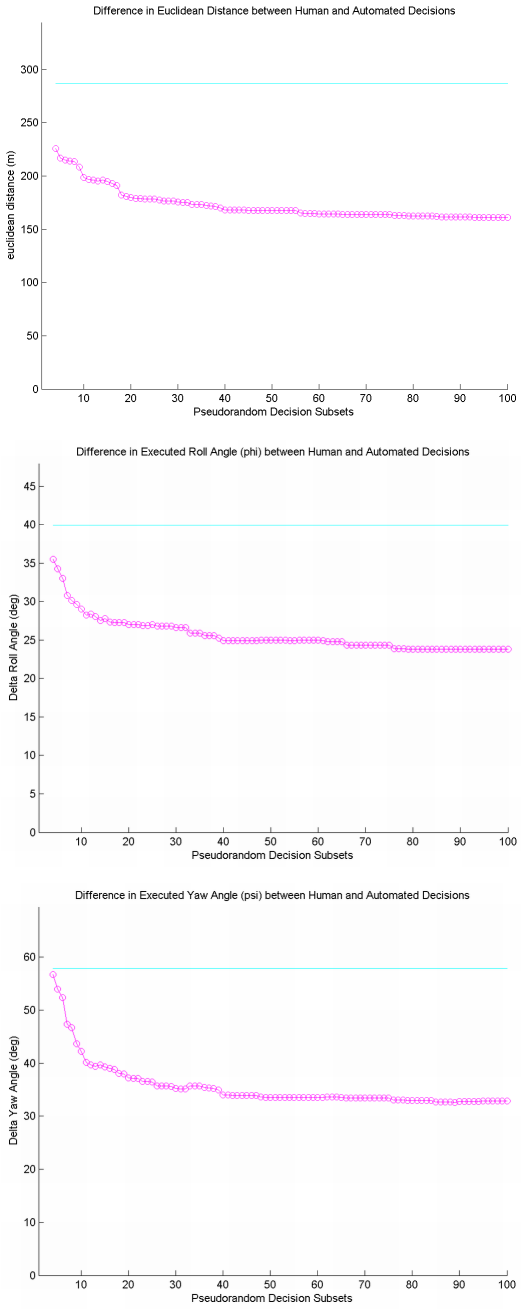


Figure B.1: Comparison of $(\Delta_{|g-s|})$, (Δ_{ϕ}) and (Δ_{psi}) for HDM 1 with respect to the number of A_R sets included in (B.1) for LC-2 and UTA-4

B.2.2 HDM 2

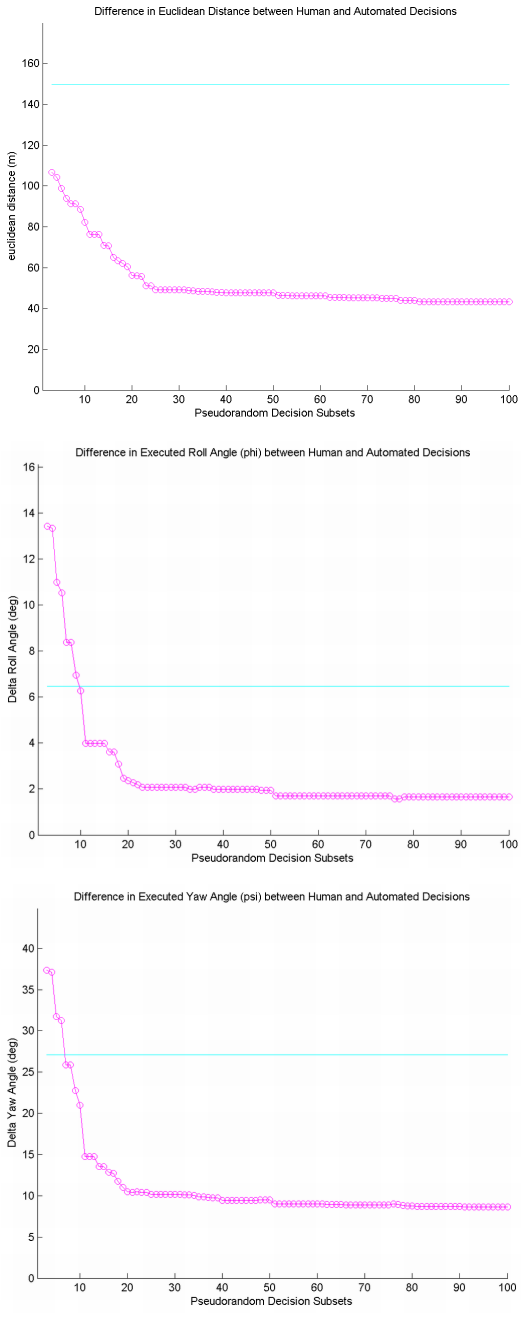


Figure B.2: Comparison of $(\Delta_{|g-s|})$, (Δ_{ϕ}) and (Δ_{psi}) for HDM 2 with respect to the number of A_R sets included in (B.1) for LC-2 and UTA-4

B.2.3 HDM 3

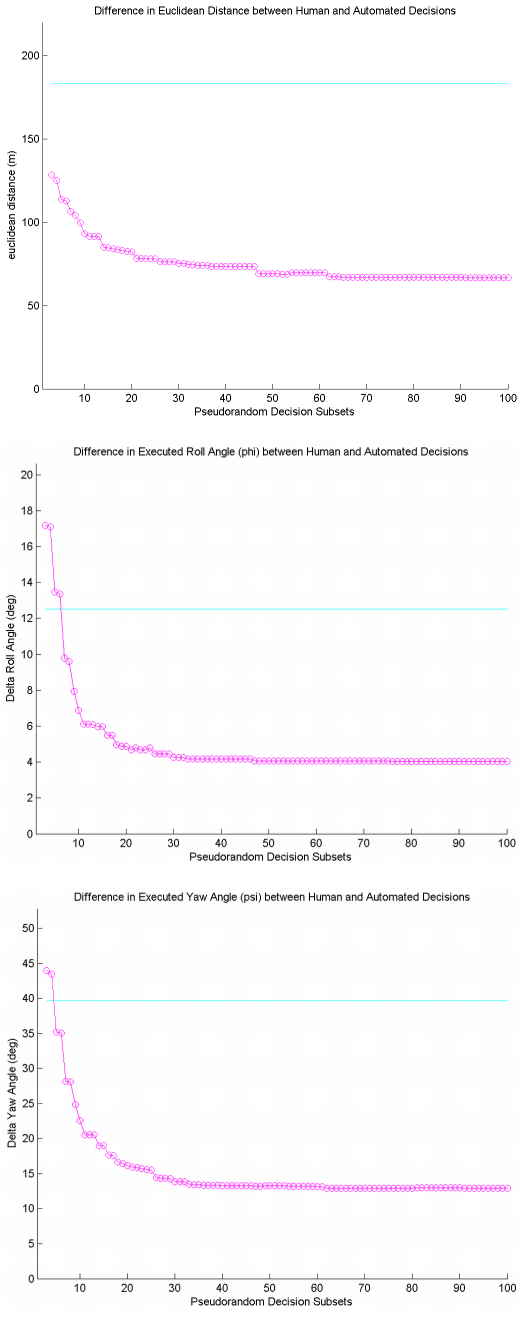


Figure B.3: Comparison of $(\Delta_{|g-s|})$, (Δ_{ϕ}) and (Δ_{psi}) for HDM 3 with respect to the number of A_R sets included in (B.1) for LC-2 and UTA-4

B.2.4 HDM 4

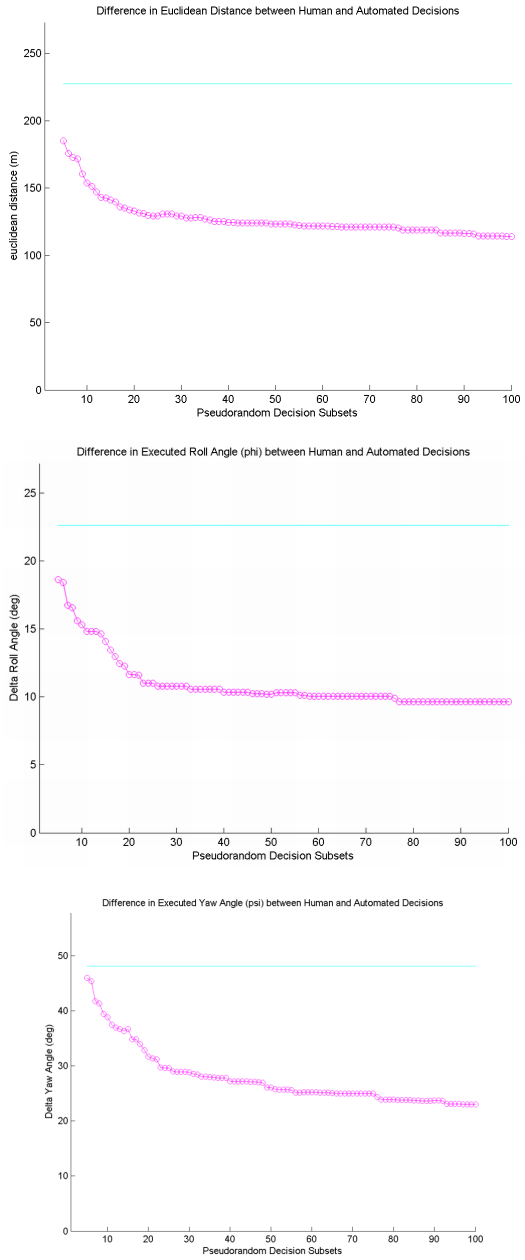


Figure B.4: Comparison of $(\Delta_{|g-s|})$, (Δ_{ϕ}) and (Δ_{psi}) for HDM 4 with respect to the number of A_R sets included in (B.1) for LC-2 and UTA-4

Appendix C

Offline HDM and ADS Decision Analysis

This appendix section presents the individual HDM's offline decision set plots. The box plots (Figures C.1, C.2, C.3, C.4) compare the costs placed on the trajectories selected by the HDM, UTA-4 (using corresponding HDM decision data) and LC-2 algorithms for all decision scenarios presented during data capture (Table A.2).

C.1 HDM 1

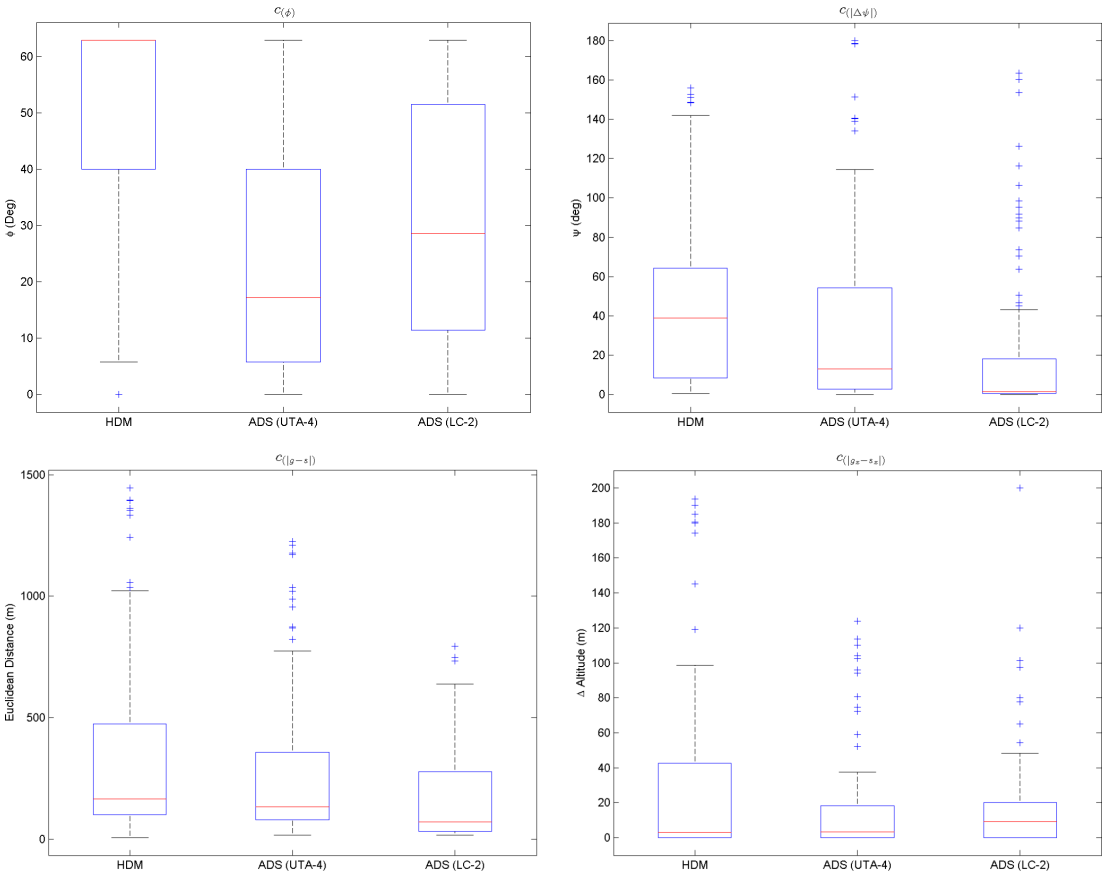


Figure C.1: Box plots comparing HDM 1 and corresponding automated decisions

C.2 HDM 2

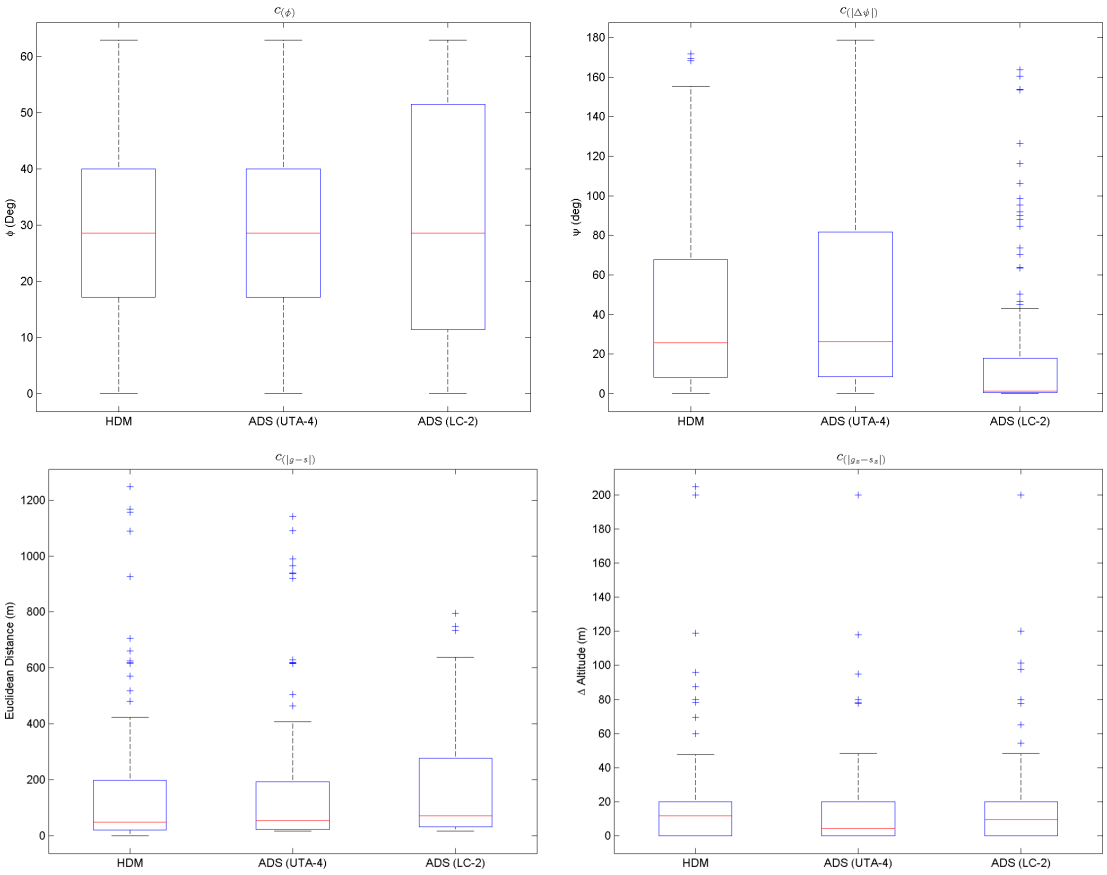


Figure C.2: Box plots comparing HDM 2 and corresponding automated decisions

C.3 HDM 3

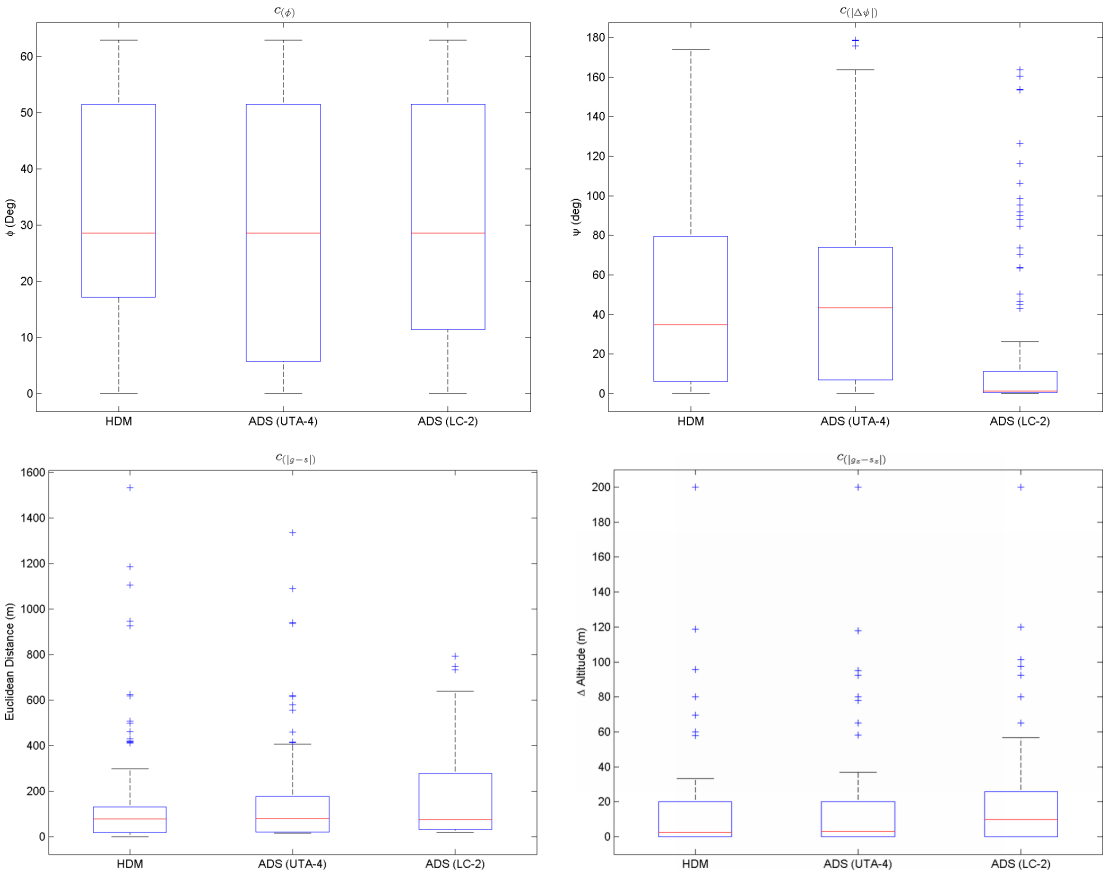


Figure C.3: Box plots comparing HDM 3 and corresponding automated decisions

C.4 HDM 4

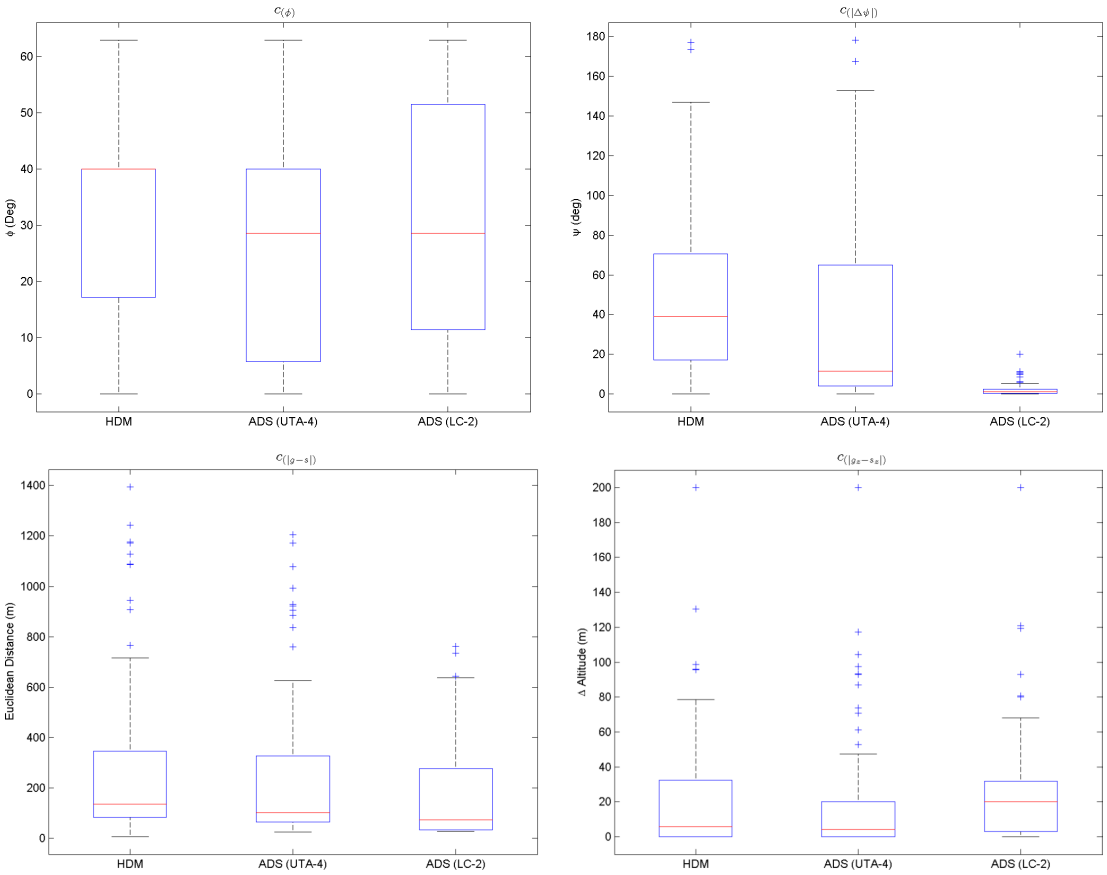


Figure C.4: Box plots comparing HDM 4 and corresponding automated decisions

Appendix D

Chapter 4 Simulation Results

This section presents additional plots comparing UA trajectories generated for LC-2 and UTA-4 solutions for simulations presented in Section 4.3.3.

D.1 Terrain simulation 1

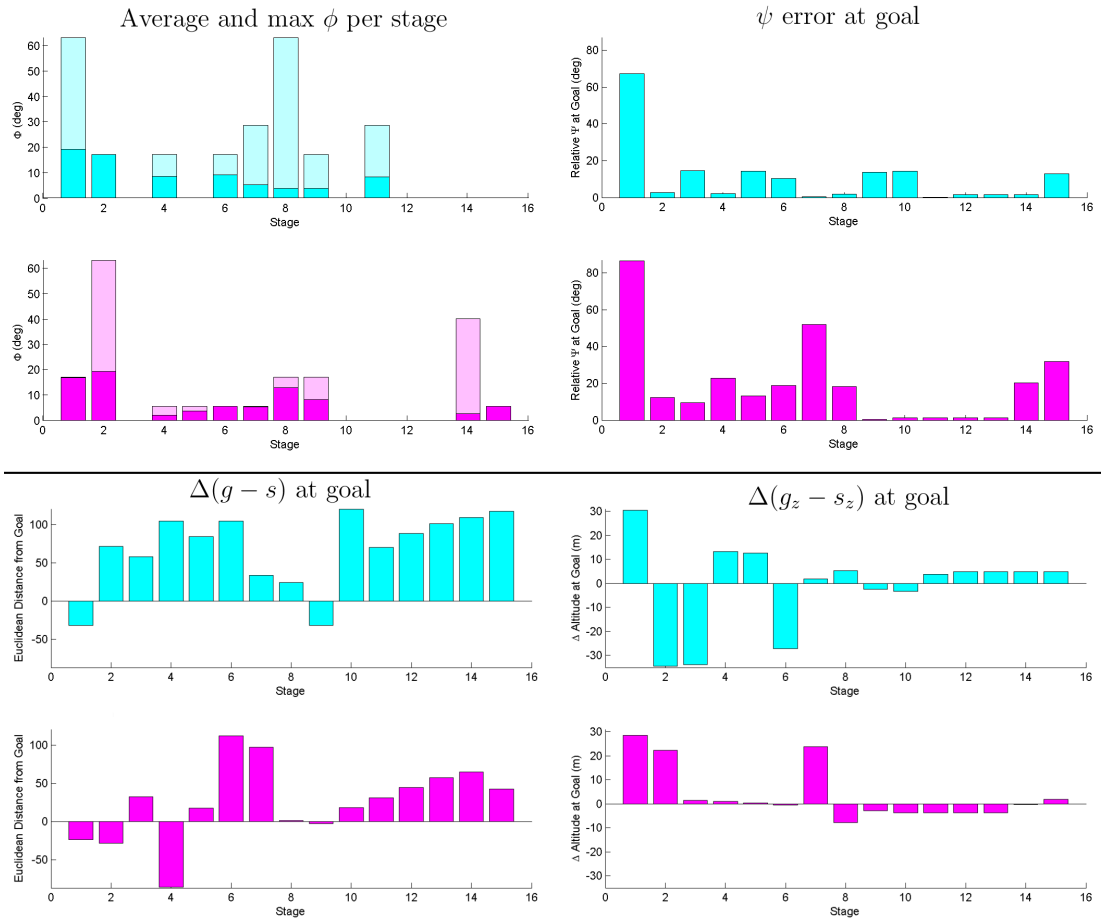


Figure D.1: Comparing UA trajectories generated for LC-2 solution and UTA-4 with HDM 2 dataset

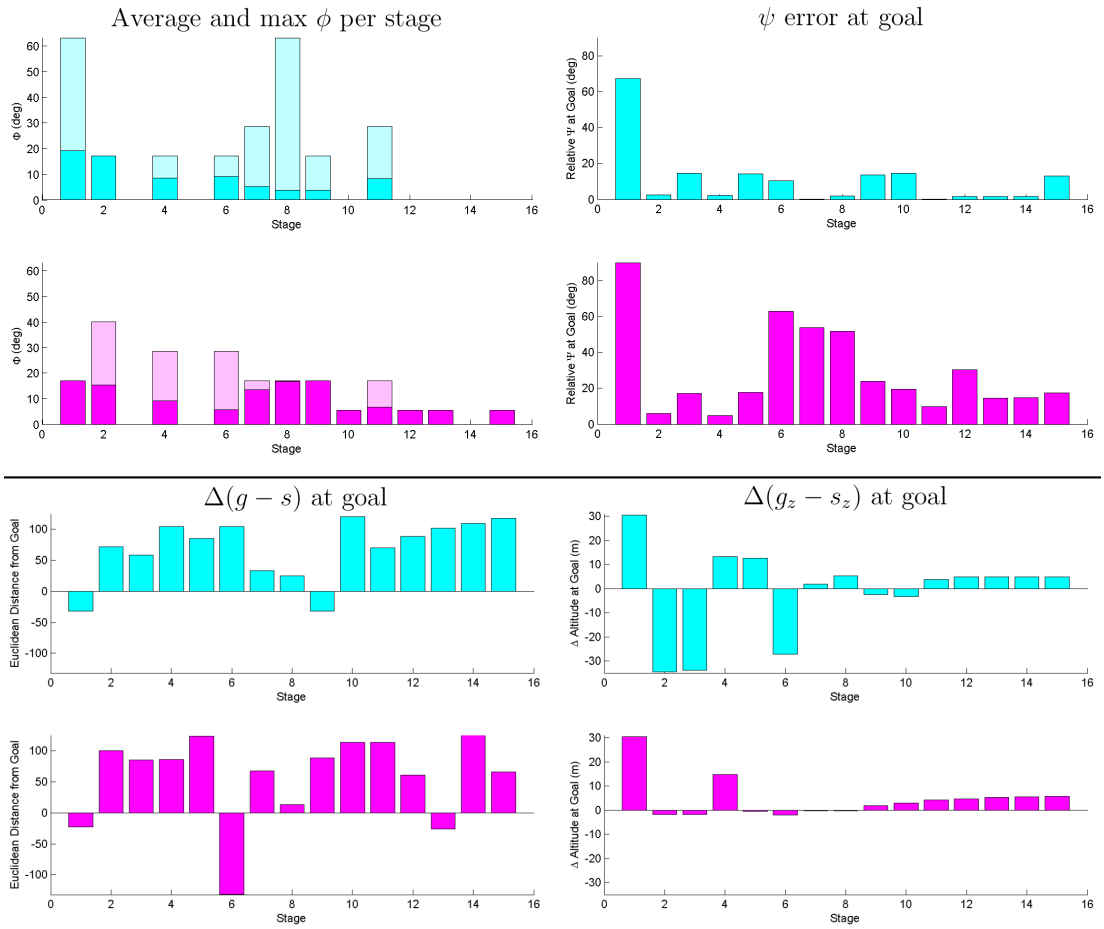


Figure D.2: Comparing UA trajectories generated for LC-2 solution and UTA-4 with HDM 3 dataset

D.2 Terrain simulation 2

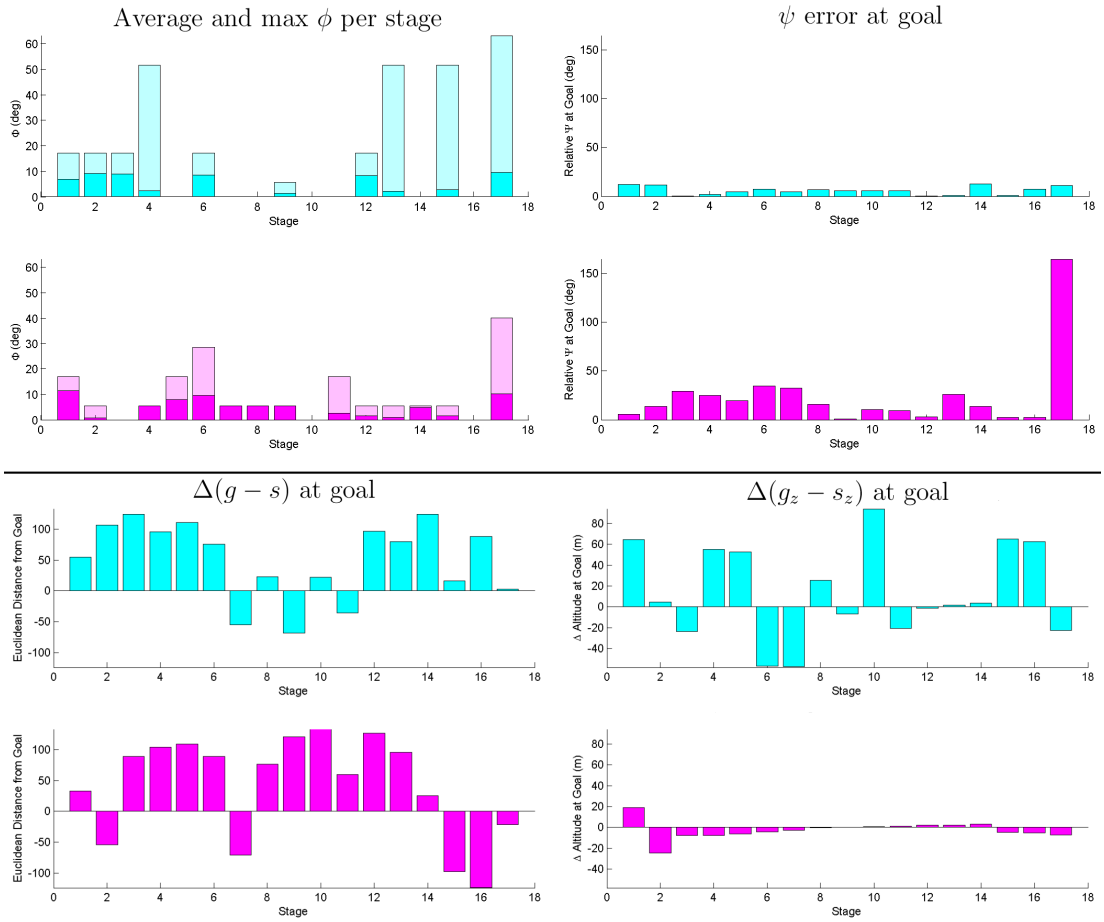


Figure D.3: Comparing UA trajectories generated for LC-2 solution and UTA-4 with HDM 2 dataset

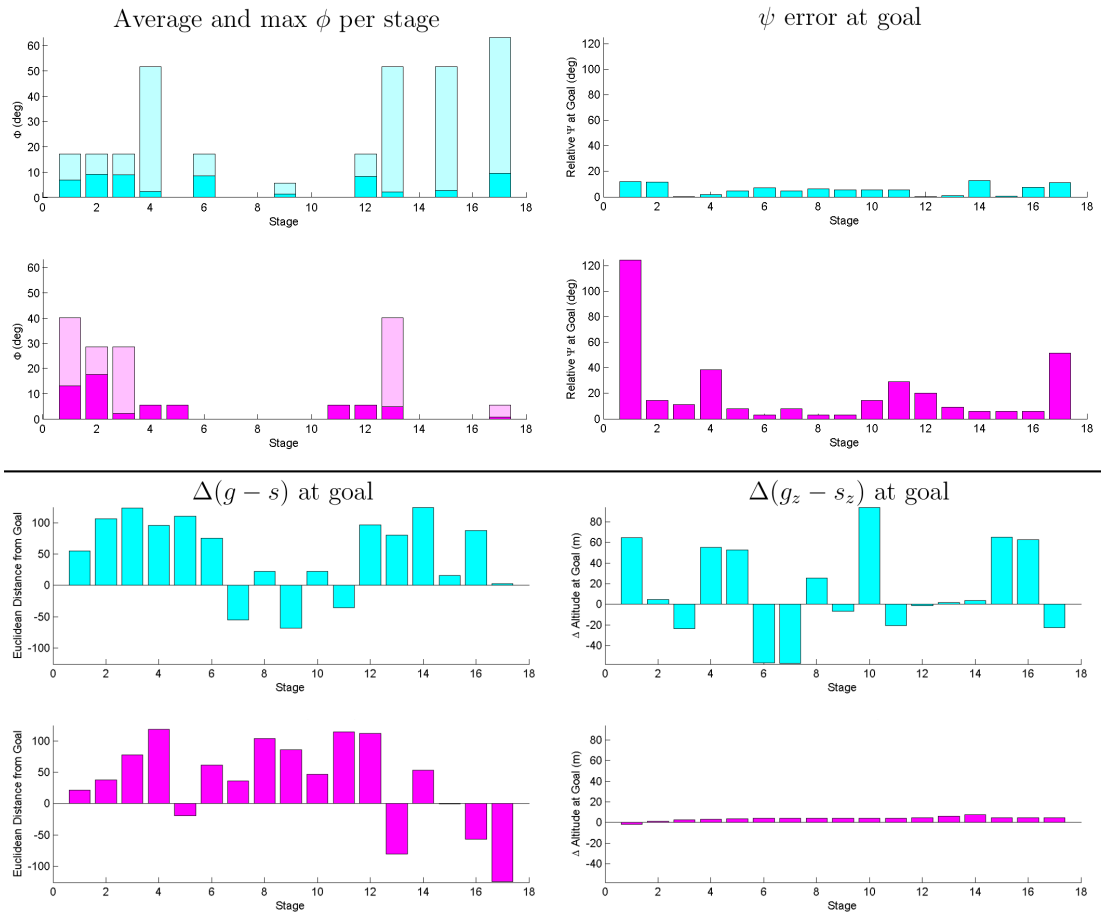


Figure D.4: Comparing UA trajectories generated for LC-2 solution and UTA-4 with HDM 3 dataset

Appendix E

Publications

During the course of this project, a number of papers have been published based on the work presented in this thesis. They are listed here for reference.

- **Narayan, Pritesh P.**, Meyer, Patrick, and Campbell, Duncan (2011) "Embedding Human Expert Cognition into Autonomous UAS Trajectory Planning," Submitted for Review. *IEEE Transactions on Systems, Man and Cybernetics Part B: Cybernetics*

- **Narayan, Pritesh P.**, Campbell, Duncan A., and Walker, Rodney A. (2009) "Computationally adaptive multi-objective trajectory optimisation for UAS with variable planning deadlines," In *IEEE Aerospace Conference* 2009, 7-14 March 2009, Big Sky, Montana.

- **Narayan, Pritesh P.**, Wu, Paul P., and Campbell, Duncan A. (2008) "Unmanning UAVs Addressing Challenges in On-Board Planning and Decision Making," In Legras, Francois (Ed.) *First International Conference on Humans Operating Unmanned Systems (HUMOUS)*, 3 - 4 September 2008, Telecom Bretagne, Brest, France.

- **Narayan, Pritesh P.**, Campbell, Duncan A., and Walker, Rodney A. (2008) "Multi-Objective UAS Flight Management in Time Constrained Low Altitude Local Environments," In 46th *AIAA Aerospace Sciences Meeting and Exhibit*, 7 - 10 January 2008, Reno, Nevada, USA.

- **Narayan, Pritesh P.**, Wu, Paul P.Y., Campbell, Duncan A., and Walker, Rodney A. (2007) "An Intelligent Control Architecture for Unmanned Aerial Systems (UAS) in the National Airspace System (NAS)," In *2nd International Unmanned Air Vehicle Systems Conference*, 20th to 21st March, 2007, Grand Hyatt, Melbourne, Australia.

- Wu, Paul P.Y., **Narayan, Pritesh P.**, Campbell, Duncan A., Lees, Michael, and Walker, Rodney A. (2006) "A High Performance Fuzzy Logic Architecture for UAV Decision Making," In *IASTED International Conference on Computational Intelligence*, Nov 20-22, San Francisco.

Papers one to five (where the author is the first author) are included on the following pages.

Embedding Human Expert Cognition into Autonomous UAS Trajectory Planning

Pritesh Narayan, Patrick Meyer and Duncan Campbell, *Member, IEEE*

***Abstract*—This paper presents a new approach for the inclusion of human expert cognition into autonomous trajectory planning, for Unmanned Aerial Systems (UAS) operating in environments with terrain present. Sets of candidate flight manoeuvres (primitives) are generated through the application of manoeuvre automaton theory and aircraft dynamic models. Smooth trajectories are formed via the concatenation of predefined trim and manoeuvre primitives. During typical UAS operations, multiple objectives may exist, therefore the use of Multi-Criteria Decision Aid (MCDA) techniques can potentially allow for convergence to trajectory solutions which better reflect overall mission requirements. In that context, Multi-attribute Value Theory has been applied to optimize trajectories with respect to multiple objectives. A Graphical User Interface (GUI) was developed to allow for knowledge capture from a human expert (pilot or mission commander) through simulated decision scenarios. The gathered expert decision data is converted into value functions and corresponding criteria weightings using UTility Additive (UTA) theory. This allows for the quantification of the human decision making process during manned operations and allows the trajectory optimizer to generate similar decisions during autonomous online trajectory planning. This approach has been demonstrated in this paper through simulation using a fixed wing UAS operating in low altitude environments with terrain present.**

***Index Terms*—Unmanned Aircraft, Unmanned Aerial System, Trajectory, Autonomous, Multi-Criteria Decision Aid, Optimisation.**

I. Introduction

UNMANNED Aerial Systems (UAS) have been employed in a diverse range of military applications to date. With respect to civilian applications, geographically sparse countries, such as Australia, have considerable potential for utilization of UAS in asset management, search and rescue, remote sensing operations and atmospheric observation [1]. However, seamless operation of UAS platforms within the National Airspace System (NAS) is required to ultimately realize this potential [2], [3].

Operation of UAS in the NAS creates a new set of challenges that are not applicable to many military applications. From a regulatory perspective, UAS need to: (i) demonstrate an Equivalent Level Of Safety (ELOS) to that of a human piloted aircraft, (ii) operate in compliance with existing aviation regulations and (iii) appear transparent to other airspace users [4].

The majority of UAS operations still require human operators to perform mission management and piloting tasks through real time communications links with the unmanned platform. This can result in high operator workload and places greater reliance on the communications link. One method to decrease operator workload is through increased levels of onboard autonomy through the inclusion of intelligent control architectures [5], [6], [7].

Intelligent control architectures [6], [8] are hierarchical methodologies which allow for the automation of aspects of UAS operations which would otherwise require a human in the loop. This research component focuses on the automation of the trajectory planning aspect of intelligent control systems. Trajectory planning is the generation of feasible collision free flight tracks in an optimal manner. In the presence of communications failures, the inclusion of automated trajectory planning processes can allow for the UAS to safely continue autonomous operations even at lower altitudes where terrain must be treated as a hazard.

Automating the trajectory planning process is however, non-trivial and some challenges include: incorporation of complex platform dynamics, trajectory optimization to meet mission objectives, and the guarantee that the generated solution is collision free. Additionally, during typical manned and unmanned operations, multiple mission objectives may exist. These objectives can include platform safety (collision avoidance and consideration of platform constraints); successful completion of the mission; minimizing fuel, time, and/or distance; or minimizing deviation from the current path. The application of Multi-Criteria Decision Aid (MCDA) techniques [9] can potentially allow for convergence to a solution which better reflects overall mission requirements.

Even with greater levels of autonomy present onboard, due to the potential risks of platform failure, UAS operations are expected to be continuously monitored by Human Decision Makers (HDMs) at the ground station. Franke [10] states that with increasing levels autonomy onboard, UAV operators move away from direct control of the platform towards a management by exception control paradigm. Management by exception occurs when the UAS performs planning and execution and informs the HDM of its current and future actions. The operator has the option to veto or override the current plans and revert to a lower control paradigm if required. Operating at higher autonomy levels requires the HDM to have a sense of trust with the automation, where he/she feels that the UAS onboard systems are making correct decisions.

P. Narayan is a PhD Candidate, Queensland University of Technology, Australia.

P. Meyer is an Associate Professor, Institut Télécom, Télécom Bretagne, UMR CNRS 3192 Lab-STICC, France.

D. Campbell is an Associate Professor, Queensland University of Technology, Australia.

Manuscript received Aug 1, 2011; revised ....., 2011.

It is important to note, that during the decision making process, the HDM will apply his/her own values, priorities and preferences for a given decision problem [11]. Different human operators may possess varying viewpoints on whether a given solution is acceptable or to be vetoed. Supervising HDM maybe reluctant to allow a UAS which they are supervising to continue operations autonomously if they do not agree with the decisions being made by automated systems onboard.

The analysis of expert decision data gathered from a set of human operators may provide a deeper understanding of objectives considered and the preferences they apply during the decision making process. Incorporating this information into the multi-objective optimization process can potentially allow automated trajectory optimizers to better encapsulate mission criteria considered by supervising HDMs and subsequently increase the acceptance of the autonomous trajectory solution [10].

This paper presents a new method for the encapsulation of the preferences of a human pilot, through multi-criteria trajectory planning for autonomous UAS operating in partially known low altitude environments. An outline of automated trajectory generation approaches and related work is given in Section II. Section II also outlines the candidate trajectory generation process, where the solution is generated through the concatenation of primitives through the application of Manoeuvre Automaton (MA) theory. Section III provides an overview of the Multi-Criteria Decision Aid (MCDA) process in context with the research problem. Section IV then presents the application of the MCDA process to the current research problem to formulate preferences from HDM decision data. Simulation results presented in section V, demonstrate how the inclusion of the human expert decision data can allow for the generation of feasible trajectories which mimic aspects of the candidate decision maker. Finally, conclusions are presented in Section VI.

II. Autonomous Trajectory Planning Overview

Due to the potential risk of platform failure, UAS operations are continuously monitored by the human operator/s via a ground station. In the event of a communications link loss, the platform can either continue operations autonomously (if the capability is present onboard), or perform a forced landing. In populated regions of the NAS, performing a forced landing may not be desirable (especially with the use of larger platforms).

The implementation of an automated trajectory planning system onboard UAS platforms has the benefit of overcoming potential ground station link issues. This allows for continued autonomous UAS operations in cluttered environments, even in the presence of communications link failures. However, automating the trajectory planning process is non-trivial and some challenges include: incorporation of complex platform dynamics and trajectory optimization to meet given mission requirements.

The inclusion of vehicle dynamics during the trajectory planning process, allows for the generation of flight trajectories which take platform constraints into account. Vehicle dynamics are used to calculate the performance envelope which the aircraft must remain within to ensure that the platform does not operate outside performance bounds.

A. Flight trajectory representation

Flight trajectories are generally represented through the use of either spline based or geometric approximations. Polynomial or spline based techniques [12], [13] place control points in a particular order to generate the desired trajectory. Geometric based techniques require the concatenation of aircraft flight manoeuvres to form a smooth flight track [14], [15], [16], [17].

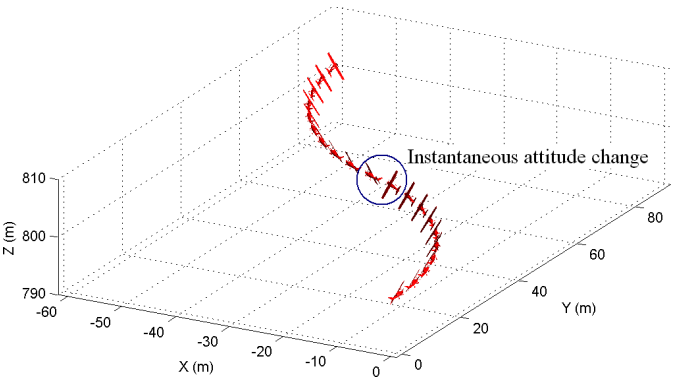


Fig. 1. Visual representation of trim concatenation without inclusion of attitude rate constraints

The actuator control power available on fixed wing platforms is finite; this leads to a transient period where the vehicle does not remain in a state of equilibrium while the platform transitions between different states of trim. While the platform remains in a state outside equilibrium (trim conditions), attitude rates will be non-zero. During periods when the platform is not in a state of equilibrium, the trajectory planner must account for platform attitude rates as a component of the overall aircraft performance envelope. This requires the continuous tracking of the platform attitude during the trajectory planning process. A candidate method which allows for the inclusion of attitude rates as a component of overall performance bounds is Manoeuvre Automaton (MA) theory [18], [19]. Figure 1 presents a visual example of the concatenation of two coordinated turn manoeuvres without the consideration for actuator control power.

B. Manoeuvre automaton theory

MA theory, proposed by Frazzoli et al. [18], [19] can be used in the generation of feasible flight trajectories through the sequential concatenation of predefined motion primitives (Figure 2). MA employs two types of primitives: trims and manoeuvres. Trim primitives represent the vehicle during a state of equilibrium whilst manoeuvre primitives characterize the vehicle operating outside a state of equilibrium. Primitives are generated using a dynamic model of the vehicle, thus platform stability can be implicitly guaranteed through generation of primitives which ensure that the vehicle remains within predefined performance bounds.

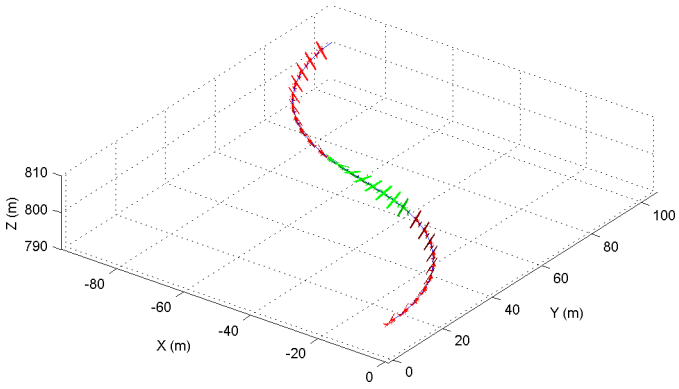


Fig. 2. Visual representation of trim concatenation with inclusion of attitude rate constraints

For this paper, MA theory is used to describe a time-invariant non-linear, dynamical system , described as a set of Ordinary Differential Equations (ODE) as [18], [19]:

$$\dot{x} := \frac{d}{dt}x(t) = f(x(t), u(t)) \tag{1}$$

where u is the control input (execution time, manoeuvre type) = $(\tau, primitive)$ and x is the state vector ([position], [attitude], [attitude rate]) = $([x, y, z], [\phi, \theta, \psi], [(\dot{\phi}), (\dot{\theta})])$.

1) Trim primitive representation: Trim primitives represent the UAS platform operating in a state of equilibrium. Using MA theory, trim primitives can be generated by placing the body fixed roll $(\dot{\phi})$ and pitch $(\dot{\theta})$ rates to zero and maintaining a constant velocity (V), roll (ϕ) and pitch (θ) angle for the duration (τ_q) of the primitive execution.

Trim primitives were generated in simulation within the MATrix LABoratory (MATLAB) programming environment using a six Degree of Freedom (DOF) flight dynamics model based on the Aerosonde Unmanned Aircraft (UA) dataset (available in the Aerosim blockset). Six predefined trim primitives have been implemented in simulation including: cruise, coordinated turn, climb, descent, helical climb and helical descent.

The initial platform state $x(t_i) = x_i$ reaches a final state $x(t_f) = x_f$ due to the execution of a given trim primitive (q); this can be represented as:

$$\begin{aligned} x_f &= x_i + \tau_q \dot{x_q} \\ t_f &= t_i + \tau_q \end{aligned} \tag{2}$$

where $\{V, \phi, \theta\}$ are constants and $\{\dot{\phi}, \dot{\theta}\} = \{0, 0\}$.

It is of importance to note, that for a platform to enter a state of equilibrium (execution of a trim primitive), the initial platform attitude must equal the attitude requirements of the trim primitive to be executed; $\{\phi, \theta\}_i = \{\phi, \theta\}_q$. If the initial platform attitude does not equal the attitude required to execute the given trim primitive ($\{\phi, \theta\}_i \neq \{\phi, \theta\}_q$), a manoeuvre primitive must be inserted to ensure that body fixed attitude rate constraints are included within performance bounds.

2) Manoeuvre primitive representation: During the execution of a manoeuvre primitive, the UAS does not have to remain in a state of equilibrium. For a fixed wing platform, the body fixed attitude rate constraint becomes $\{\dot{\phi}, \dot{\theta}\} = \{\dot{\phi}_{max}, \dot{\theta}_{max}\}$. In this paper, manoeuvre primitives (p_m) are employed to connect two trim primitives, if required, in the formation of feasible trajectories. This allows for the consideration of attitude rates as an additional platform constraint during periods where the UAS is not in a state of equilibrium, where $\{\phi, \theta\}_i \neq \{\phi, \theta\}_q$.

If $\{\phi, \theta\}_i \neq \{\phi, \theta\}_q$, the UAS platform dynamic model is propagated until the platform reaches the desired state configuration.

While

$$\begin{aligned} \{\phi, \theta\}_k &\neq \{\phi, \theta\}_q \\ x_{k+1} &= x_k + \dot{x_p}\Delta T \\ t_{k+1} &= t_k + \Delta t \end{aligned} \tag{3}$$

where $\{\dot{\phi}, \dot{\theta}\} = \{\dot{\phi}_{max}, \dot{\theta}_{max}\}$.

C. Generating feasible trajectories through concatenation

A smooth, nominal, feasible and collision free trajectory is required for safe guidance of the UA from its current state to the desired goal state. The final trajectory is formed through sequential concatenation of selected trim primitives (and corresponding manoeuvre primitives, if required) where each trim primitive selected for execution can be considered as a stage. Concatenation without optimisation, may lead to the generation of trajectories which do not accurately represent mission objectives, thus a decision strategy is required to generate trajectories which best meet one or more mission criteria.

Dynamic programming (DP) [20] has been previously employed in related research [21], [22], [18] for the optimization of feasible trajectories generated through the application of MA theory. DP is a sequential optimization method which finds the least cost (optimal) solution from a set of alternative solutions. To guarantee that the optimal solution is found, the DP algorithm must consider all possible alternatives across all stages.

In comparison to the application of DP to trajectory planning with respect to a generic graph search implementation, the current UA position can be treated as the current node. Each possible state the platform can reach through the execution of currently stored trim primitives must be treated as neighboring nodes. Expanding each neighboring node would cause the algorithm to grow exponentially in computational complexity for each additional stage considered in the overall optimization process [23].

To decrease the computational complexity and resulting time to plan, Frazzoli [18], [19] applies a hybrid architecture to the motion planning problem for rotary aircraft. The hybrid architecture involves integration of DP (optimised over single stage) with other other optimisation algorithms such

as Rapidly Exploring Random Trees (RRT) [18] and Model Predictive Control (MPC) [21].

The research presented in this paper uses the DP search algorithm but limits the search to single stage optimization. This converts the DP algorithm to a greedy search implementation, which essentially chooses the trim primitive, trim execution time and manoeuvre execution time with the least cost for the each stage in a sequential manner. The UAS position after execution of the optimal trim primitive is taken as the next node for expansion, and continues until the goal is reached.

Executing a DP search algorithm iteratively over a single stage without explicit consideration for future stages ensures that the computational complexity and resulting time to plan remains comparatively small. However, not considering all stages during the optimization process means that global trajectory solution optimality and completeness cannot be guaranteed.

Global path solution optimality and completeness can be guaranteed through the application of an intelligent control architecture with a mission/path planning layer which uses a deterministic search algorithm to generate an optimal set of waypoints from the current position to the goal [23]. In addition, during operations in dynamic and partially known environments, a greedy motion planning implementation can suffice as it may not be possible to find a global trajectory solution due to limited environment representation.

D. Summary of findings

This section presented the generation of feasible collision free trajectories for fixed wing platforms operating autonomously using MA theory. Planning in 3D enviroments was possible through the formulation of common aircraft flight modes. Attitude rate constraints were included through the inclusion of manoeuvre primitives to allow for increased trackeability. Single stage DP optimisation was selected for the generation of trajectories in a computationally efficient manner.

Operating at higher autonomy levels requires the HDM to have a sense of trust with the automation, where he/she feels that the UAS onboard systems are making correct decisions. During the decision making process, the HDM will apply his/her own values, priorities and preferences for a given decision problem [11]. Furthermore, different human operators may possess varying viewpoints on whether a given solution is acceptable or should be vetoed. The inclusion of multiple criteria through Multi-Criteria Decision Aid (MCDA) strategies during the trajectory selection process may allow for better representation of the HDM's preferences and mission objectives. MCDA is a field of research for the development of multi-criteria decision tools to assist HDMs and can also be applied to autonomous scenarios [24]. The following section investigates application of MCDA methodologies to represent HDM and mission requirements more accurately during the generation of feasible trajectories based on MA theory.

III. MCDA STRATEGY

Many problems can be solved through the application of decision analysis and decision aid techniques. The decision aid process generally provides a HDM with the most appropriate solution from a given set of alternatives. Each alternative will have one or more characteristics (criteria) which represent different dimensions in which an HDM can view the desirability of a given alternative by.

During the course of flight operations, the pilot/UAS operator may have to consider multiple criteria in order to achieve mission success. Examples of mission criteria generally include: achieving the mission goal/s; safety of the vehicle, the environment and the public at all times; mission efficiency (minimising time, fuel and/or cost); and/or limiting operations to below a specified altitude ceiling. Mission objectives and their priorities can dynamically change at any point during UAS operations (usually at the discretion of the operator).

Decision making during autonomous trajectory planning requires the selection of the most optimal feasible collision free trajectory with respect to one or more criteria. Gigerenzer et al. [25] have shown that HDMs do consider multiple criteria during real-life decision making processes. Therefore, the use of MCDA methodologies during autonomous trajectory planning may allow for convergence to a solution which better reflects overall mission requirements. The following section presents an overview of MCDA techniques.

A. MCDA overview

MCDA is a category of decision aid methods in which decisions are formulated through the comparison of alternatives with respect to multiple criteria. Many MCDA techniques [24] have been published to date which can be used to determine the most suitable alternative, or to sort or rank a set of alternatives. MCDA techniques can roughly be divided into two categories: on the one hand Multiple Attribute Value Theory (MAVT) [26], [27], which aims at aggregating the multiple points of view into a unique synthesis criterion, and, on the other hand outranking methods [28] which aim at comparing the decision alternatives pairwisely and accept incomparability.

MCDA allows for the encapsulation of the HDM's decision style through the inclusion of preference information and a relevant set of criteria. Preference information can take various forms, among which for example the relative importance of each criterion to the HDM. The capture of these human preferences is called preference elicitation and depends on the HDM's individual decision experiences and training that he may have received. The following section presents a generic overview of the MCDA process and its use in the context of automated flight operations.

B. MCDA process

The MCDA process requires the implementation of algorithms which attempt to mimic aspects of the HDM's decision making style and take into account his/her preferences. Classically, an MCDA process can be divided into the following four steps [29]:

1) *Determining the relevant criteria and alternatives;*
2) *Evaluating the alternatives on all the criteria;*

3) *Eliciting the HDM's preferences related to the current decision problem;*
4) *Combining the evaluations and the preferential information to solve the decision problem and produce a decision recommendation.*

In the sequel we detail each of these steps in our context.

1) Determining relevant criteria and alternatives: The DP algorithm is applied to this research for optimal trajectory selection, but the search is limited to single stage only. Optimisation using DP search over one stage involves selection of the optimal manoeuvre and corresponding jump time from a predefined set of manoeuvres for each stage in an iterative manner. Each discrete jump time, for a given manoeuvre, can be represented as a unique decision alternative, as it will result in a different final state if executed. Let A be the set of such alternatives.

For this research problem, the decision alternatives generated are sets of trim primitives (p) which can be safely executed by the platform (Figure 6). The trim primitives represent a set of sampled states (i) which the UAS can reach after successful execution of a relevant manoeuvre. Thus, the total number of alternatives for (m) trim primitives is:

$$|A| = \sum_{n=1}^{m} p_n i_n \tag{4}$$

The criteria represent different dimensions with which an alternative can be viewed by. In literature, it was found that Frazzoli et. al [18] applied two such criteria: minimizing euclidean distance of current (s) and goal (g) states (criterion $\text{crit}_{(|g-s|)}$); and minimizing platform yaw (ψ_s) and goal yaw (ψ_g) angles (criterion $\text{crit}_{(|\Delta\psi|)}$) during optimal manoeuvre selection.

If $\text{crit}_{(|g-s|)}$ and $\text{crit}_{(|\Delta\psi|)}$ do not completely encapsulate the HDM's decision strategies, then the inclusion of additional criteria allows the onboard trajectory planner to take into account certain aspects of the mission which cannot be considered using only the current two criteria. For example, executing very sharp turns (high bank angles) can lead to platform instability [30].

Platform safety can be implicitly considered through the inclusion of $\text{crit}_{(|\phi|)}$ which focuses on the minimization of high platform roll angles (ϕ_s). It is important to note that through the implementation of MA theory, platform safety can be increased without penalizing platform manoeuvrability.

The second additional criterion $(\text{crit}_{(|g_z-s_z|)})$ considers the minimization of the altitude of the goal (g_z) and current state (s_z). For decision scenarios where the goal is not at the same altitude as the platform, this criterion captures how focused a HDM is on reaching the required altitude.

2) Evaluating alternatives on all the criteria: In order to perform decisions on the set of alternatives (e.g. generating the most optimal decision or ranking/sorting), an evaluation scale needs to be attached to each of the criteria. Each alternative is then evaluated by placing a cost to go (from current state to the alternate state) on all attached criteria.

Whilst Frazzoli has not explicitly defined the criteria applied in literature, $\text{crit}_{(|g-s|)}$ can be expressed in 3D planning space as the euclidean distance between the goal location (g) and the current location (s). A lower cost $(c_{(|g-s|)})$ is placed on p which drive the UAS platform closer to the goal (5).

$\text{crit}_{(|\Delta\psi|)}$ allows for greater control of the heading of the platform. For this research (ψ_g) represents the direction to next goal. The cost $(c_{(|\Delta\psi|)})$ can be calculated by taking the absolute difference between the desired (ψ_d) and absolute platform headings (ψ_a). Alternatives with a resulting (ψ_a) closer to (ψ_d) will have a lower cost placed on them (6).

$$c_{(|g-s|,i)} = |g-s| \in [\min|g-s|, max|g-s|] \tag{5}$$

$$c_{(|\Delta\psi|,i)} = |\psi_d - \psi_a| \in [0, \pi] \tag{6}$$

The evaluation of $\text{crit}_{(|\phi|)}$ has been performed by placing a greater cost $(c_{(\phi,i)})$ on trim primitives which are executed with higher roll angles (7). Finally, a greater cost $(c_{(|g_z-s_z|)})$ is placed on trim primitives which do decrease the relative vertical distance between the platform state (s_z) and goal state (g_z) for $\text{crit}_{(|g_z-s_z|)}$ (8).

$$c_{(\phi,i)} = \phi \in [0, \phi_{max}] \tag{7}$$

$$c_{(|g_z-s_z|,i)} = |g_z - s_z| \in [\min|g_z - s_z|, max|g_z - s_z|] \tag{8}$$

Each candidate HDM may have their own perception of the relative importance of each criteria and thus the desirability of the alternatives presented. If an automated onboard trajectory planner applies multiple criteria without accounting for the relative importance placed on each criteria by the candidate HDM, the trajectory solution maybe quite different from what the UAS operator expects. The following section provides an overview of methods present in literature which formulate preferences through the analysis of HDM decision data.

3) Eliciting the HDM's preferences related to the current decision problem: Roughly speaking, this elicitation can be performed either by questioning the HDM directly on the values of the various preferential parameters, or by extracting this information via a disaggregation technique from an order on some alternatives which the HDM is able to express.

To capture such expert knowledge in a direct way, one can use the MACBETH technique (Measuring Attractiveness by a Categorical Based Evaluation Technique) [31]. MACBETH's goal is to build a cardinal scale measuring the attractiveness of options through a learning process involving an interactive software. The HDM is asked to perform qualitative pairwise comparisons regarding his preferences between various evaluation levels and express himself on a scale reaching from *very weak* to *extreme*.

A well-known disaggregation approach is UTA (UTilité Additive) [32]. Here the HDM is tasked first with ranking a few well-known alternatives. Linear Programming (LP) techniques are then used to perform an ordinal linear regression in order to determine a preference model which is consistent with the HDM's overall preferences. Both MACBETH and UTA approaches generate value functions and weighting vectors which correspond to the HDM's preferences. These can then

be used in MAVT based decision algorithms. UTA has been selected as the candidate method for the conversion of HDM decision strategies to preference parameters as it allows for more intuitive capture if the alternatives are presented to the HDM visually through a Graphical User Interface (GUI) (Figure 3).

Let $A = \{x, y, z, \ldots\}$ be the set of alternatives and $J = \{g_1, \ldots, g_n\}$ be the set of n criteria. Each criterion can be seen as a real-valued function on the set A. Let $g(x)$ be the vector of evaluations of alternative x of A on the criteria of J. The criteria aggregation model in UTA is assumed to be an additive value function of the following form:

$$u(g(x)) = \sum_{i=1}^{n} w_i u_i(g_i(x)) \quad \forall x \in A \tag{9}$$

where u_i $(i = 1, \ldots, n)$ are real-valued functions called marginal value functions which are normalized between 0 and 1, w_i is the weight of criterion i, and u is the overall value function. A higher value of u_i is associated with a better alternative on criterion i.

In UTA, the ranking given by the HDM on a subset of alternatives is transformed into a set of linear constraints on u, which are added to the UTA disaggregation LP (see [32] for further details). The objective of this LP is to minimize the gap between the initial ranking given by the HDM and the one produced by the aggregation model. The output of the UTA LP is a set of value functions and associated weights which represent the HDM's preferences, based on the input ranking that he/she provided.

4) Determining a ranking of the alternatives: In order to determine which alternative is the most attractive for the HDM, a ranking of all the alternatives is computed. This allows the HDM direct access not only to the "best" solution, but to the remaining solutions and corresponding rankings.

The aggregation technique used here is based on MAVT and requires the value functions and the weights obtained by the UTA technique. Consequently, the aggregation formula (9) is applied on the set of feasible alternatives. Thus, each of the alternatives gets an overall value (u), which allows to rank them from the most to the least attractive one.

C. Summary of findings

This section presented a brief overview of MCDA and outlines the MDCA process to generate feasible trajectories which applied aspects of candidate HDMs decision styles. Alternatives were defined as unique feasible sampled states which could be reached by the UAS platform. Criteria represented different dimensions with which a HDM could view the desirability of each alternative by.

The UTA disaggregation technique was selected to formulate preference information to represent HDM preferences and priorities for each criteria. An additive, MAVT decision strategy would then be applied to incorporate HDM preferences during the aggregation of value functions representing mission criteria. The following section details the application of the proposed MCDA process to the current research problem to generate trajectory solutions which more accurately represent HDM and mission objectives.

IV. Application of the proposed MCDA process to the current research problem

This section applies the MCDA process to the current research problem to formulate preferences which represent HDM's mission priorities. The following section details an overview of the HDM data capture process.

A. Expert knowledge capture and decision modeling strategies

One way of viewing the trajectory planning problem using single stage optimisation is that the candidate HDM is presented with unique decision scenarios, where they must select the most appriopriate trajectory segment in an iterative manner until the mission is completed. During trajectory selection, the HDM's preferences may vary depending on the decision scenario presented to them, for example, the HDM may have a different set of preferences in mind when the UA is closer to the goal as opposed to decision scenarios where the UA position is farther from the goal.

A decision scenario can be defined as the relative difference between the goal and UA positions ($|x_g - x_p|$, $|y_g - y_p|$, $|z_g - z_p|$) and the relative orientation of the UA with respect to the desired direction at the goal ($\psi_g - \psi_p$). In addition, the automaton generated will be unique to the platform roll angle (ϕ_p) due to the inclusion attitude rate constraints; this results in a unique set of A for the HDM to consider. Thus, each unique decision scenario can be represented as ($|x_g - x_p|$, $|y_g - y_p|$, $|z_g - z_p|$, ($\psi_g - \psi_p$), ϕ_p). Figure 4 shows an example scenario presented to the candidate HDM.

The capture of HDM decision data for each unique decision scenario only provides a discrete snapshot of the candidate HDM's decision preferences for that particular scenario. In order to perfectly model a HDM's decision style would require data capture over an extremely large (approaching infinity) set of unique decision scenarios; this is not feasible. Thus, a sampled set of unique scenarios (which represent a discrete approximated subset of unique decision scenarios) are presented to the HDM via the GUI during data capture.

In order to elicit human expert decision preferences, a GUI was developed to generate a set of simulated decision scenarios, and to capture the corresponding candidate HDM's decision patterns (Figure 3). The HDM uses the GUI to intuitively select what they consider to be the most suitable decision from a set of alternatives (discrete sample points along each trim primitive) for each unique decision scenario. The trim primitives include straight and level flight, climb, descend, coordinated turn, helical turn and helical descent manoeuvres.

120 unique decision scenarios are completed by each HDM to form a bank of HDM decisions (Figure 5). The HDM decisions are then used to form preferences, for inclusion into a MAVT based ADS, that generates trajectories which incorporate aspects of HDM decision strategies. The following section provides an overview of the formulation of preferences through the application of UTA theory to the current research problem.

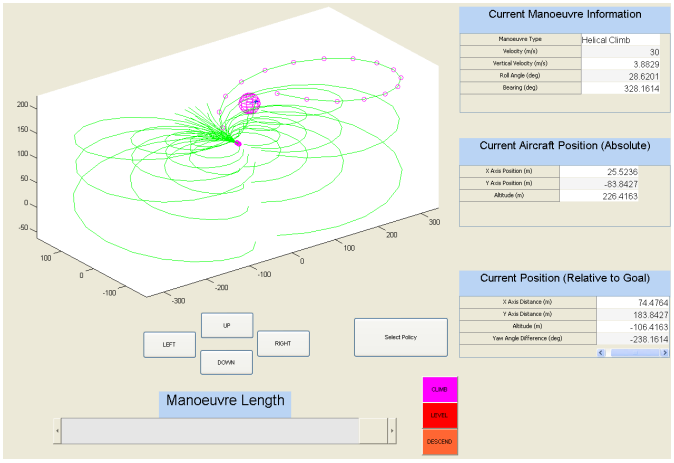


Fig. 3. Graphical User Interface developed for HDM data capture

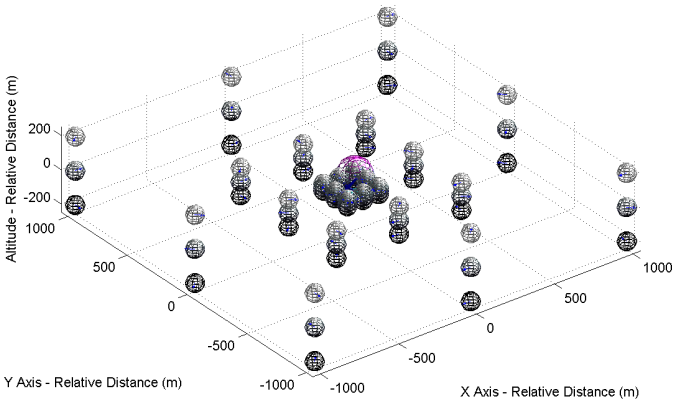


Fig. 5. Decision sets completed by HDM

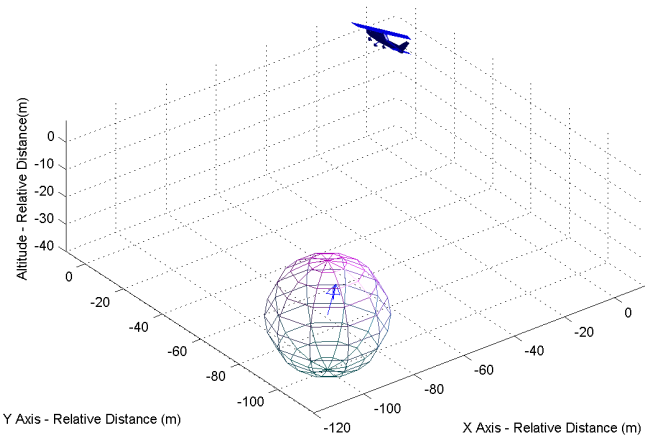


Fig. 4. Example decision scenario presented to HDM

B. Preference formulation using UTA

UTA is applied to all decision sets completed by the HDM to form a selectable bank of preference data (Figure 5). The following sections present three experiments on three different problem formulations. A least cost formulation (LC-2) represents the inclusion of $\text{crit}_{(|g-s|,i)}$ and $\text{crit}_{(\Delta\psi,i)}$ with equal preference weighting as the reference solution. UTA-2 represents the inclusion of $\text{crit}_{(|g-s|)}$ and $\text{crit}_{(|\Delta\psi|)}$ where UTA is applied to generate value functions and weighting values using the candidate HDM's decision data. UTA-4 describes the inclusion of all four criteria presented in Section III-B1 where value functions and weighting values are again generated from candidate HDM decision data using UTA.

1) LC-2: An ADS applying the LC-2 decision algorithm generates trajectories where $\text{crit}_{(|g-s|)}$ and $\text{crit}_{(|\Delta\psi|)}$ are given equal preference. The cost functions $c_{(|g-s|)}$ and $c_{(|\Delta\psi|)}$ can be equivalently represented as value functions $\mu_{(|g-s|,i)}$ and $\mu_{(\Delta\psi,i)}$ respectively (10)(11).

$$\mu_{(|g-s|,i)} = 1 - \left(\frac{c_{(|g-s|,i)}}{\max(c_{(|g-s|,1..n)}} \right) \tag{10}$$

$$\mu_{(\Delta\psi,i)} = 1 - \left(\frac{(c_{\Delta\psi,i})}{\pi} \right) \tag{11}$$

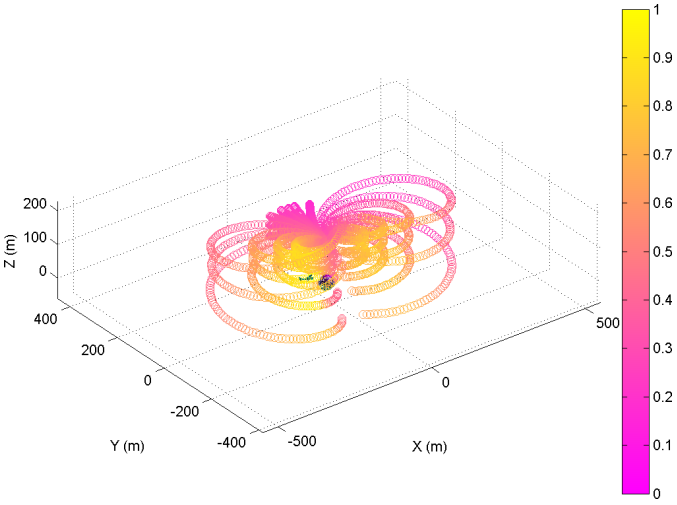


Fig. 6. Normalized aggregated decision values for all criteria (LC-2)

LC-2 may not accurately represent mission requirements as the candidate HDM may have their own perceptions on which criteria's are relevant to the current mission scenario and the preference given to each relevant criteria.

2) UTA-2: UTA theory is applied to the HDM decision sets to generate value functions and weighting values for $\text{crit}_{(|g-s|)}$ and $\text{crit}_{(|\Delta\psi|)}$ which provide a mathematical representation of the HDM's decision style for each given scenario. Figure 7 shows the value functions generated using UTA theory for the sample decision scenario when $\text{crit}_{(|g-s|)}$ and $\text{crit}_{(|\Delta\psi|)}$ are applied (Figure 4). Note that the weighting value is embedded within each value function (the maximum value of the value function corresponds to the weight coefficient of Formula 9).

The aggregate decision values generated by the ADS, with the application of UTA-2 (Figure 8) for the set of A shows that the desirable alternatives are concentrated into a singular region. In comparison, the aggregate decision values generated by LC-2 (Figure 6), UTA-2 shows a region focused near the goal state where alternatives have the highest utility decision

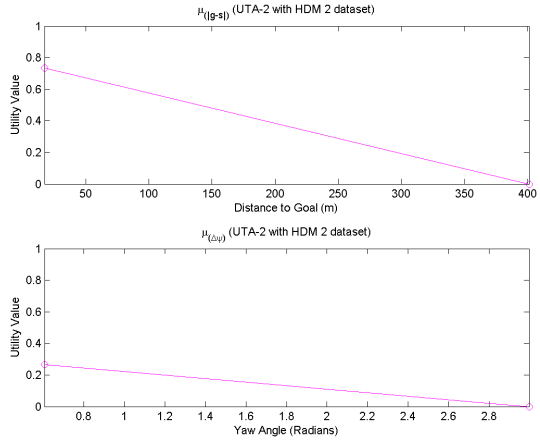


Fig. 7. UTA value functions (UTA-2 with HDM 2 dataset) representing HDM preferences for sample decision scenario (Figure 4)

values. This is due to the value functions generated using UTA place a higher preference for $\text{crit}_{(|g-s|)}$ during optimisation (Figure 7).

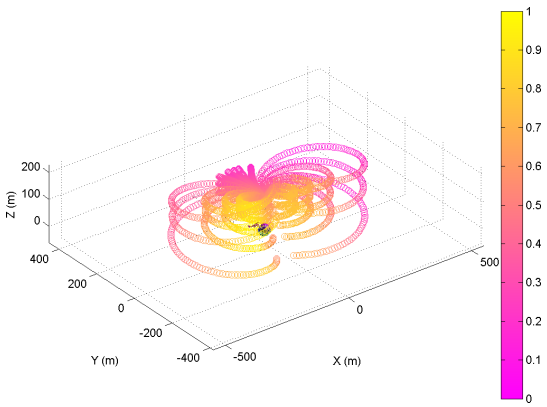


Fig. 8. Normalised aggregated decision values for all criteria (UTA-2 with HDM 2 dataset))

3) UTA-4: In order to investigate if the inclusion of additional criteria can allow UTA to represent HDM decisions with further accuracy, UTA-4 applies two additional criteria ($\text{crit}_{(|\phi|)}$ and $\text{crit}_{(|g_z - s_z|)}$) during preference formulation using UTA theory. Figure 9 shows the value functions generated using UTA theory for the sample decision scenario (Figure 4) when the two previous and two additional criteria are applied.

The aggregate decision values generated by the ADS (UTA-4 with HDM 2 dataset) (Figure 10) show several regions which are near optimal. We can see on Figure 9 that $\text{crit}_{(|\Delta\psi|)}$ has the greatest effect, thus all alternatives which have a low $c_{(|\Delta\psi|)}$ appear as near optimal solutions.

The following section compares UTA-4 and UTA-2 against LC-2 (reference least cost solution) to investigate the HDM decision modelling accuracy of UTA theory using HDM datasets captured using.

C. Accuracy of UTA

The average results for all decision sets were compared to the trajectories selected by the HDM to determine how

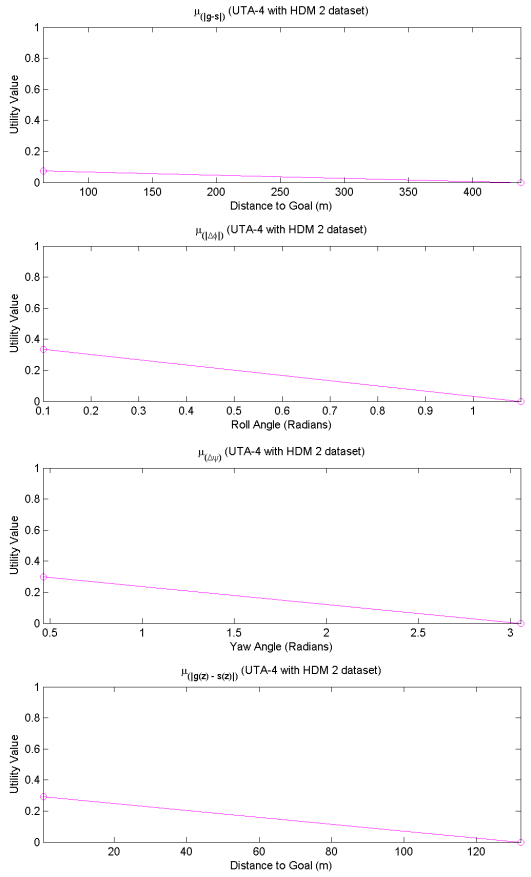


Fig. 9. UTA value functions (UTA-4 with HDM 2 dataset) representing HDM preferences for sample decision scenario (Figure 4

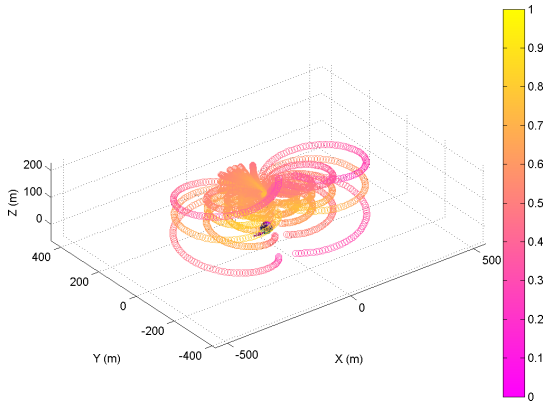


Fig. 10. Normalised aggregated decision values for all criteria (UTA-4 with HDM 2 dataset)

accurately the decisions were modelled by calculating the difference between the human and the ADS solutions for; roll angles (Δ_ϕ), euclidean position between goal and current state ($\Delta_{|g-s|}$) and platform yaw angles (Δ_{psi}).

The application of UTA-2 generated decisions which had a lower average Δ_ϕ, $\Delta_{|g-s|}$ and Δ_{psi} in comparison to the automated generation of decisions using LC-2 (Figure 11). This implies that $\text{crit}_{(|g-s|)}$ and $\text{crit}_{(|\Delta\psi|)}$ are relevant and considered by the HDM during the decision making process.

Additionally, the inclusion of additional criteria (UTA-4) generated decisions which were even closer to the HDM decisions captured than with just the inclusion of two criteria (UTA-2) (Figure 11).

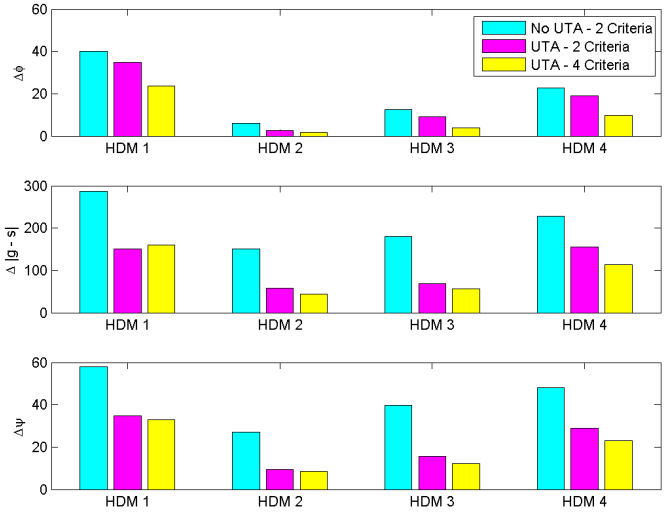


Fig. 11. Average error comparison between human and automated trajectory decisions for all decision sets

Further analysis of the individual HDM's offline decision set shows how the UAS platform is expected to perform during autonomous operations with the inclusion of HDM preferences. HDM 2 generally executed flight manoeuvres where $\phi \in [20°, 40°]$ (Figure 12). The ADS with the inclusion of HDM preferences (UTA-4) executed primitives within a similar range to HDM 2. The LC-2 formulation does not explicitly take ϕ limitations into account, subsequently the ADS using an LC-2 optimisation had greater variance in the roll angle range of the primitives executed (Figure 12). It is expected that UTA-4 using HDM 2's decision data will not periodically execute manoeuvres with higher roll angle values unlike the ADS using an LC-2 optimisation. This is desired as the execution of flight manoeuvres with higher wing loading values has a greater possibility of platform instability [30].

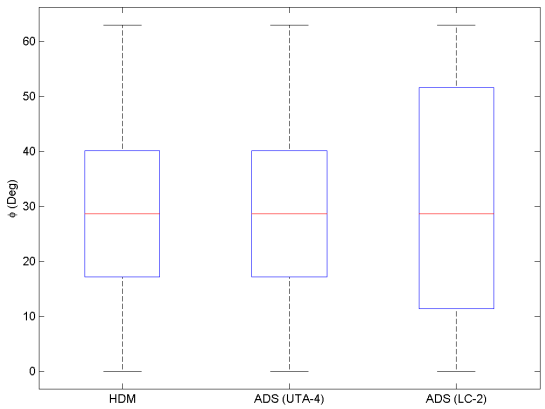


Fig. 12. Box plots comparing UAS platform ϕ for offline trajectories selected by HDM 2, ADS LC-2 and ADS UTA-4 solutions

HDM 3 selected flight manoeuvres where a greater preference was placed on minimizing the altitude of the platform s_z with respect to the goal altitude g_z (Figure 13). LC-2 only considers altitude minimisation as a component of $\text{crit}_{(|g-s|)}$, therefore during offline simulation, it was found that LC-2 optimisation had a greater variance in comparison to the HDM and UTA-4 trajectory solutions. It is expected that UTA-4 with the inclusion of HDM 3's decision data is more likely generate trajectories with lower $|g_z - s_z|$ values in comparison to the ADS using an LC-2 optimisation (Figure 13). This reflects on the candidate HDM's preference on maintaining a similar altitude to the goal which can be beneficial for certain missions e.g. airborne surveillance and video capture.

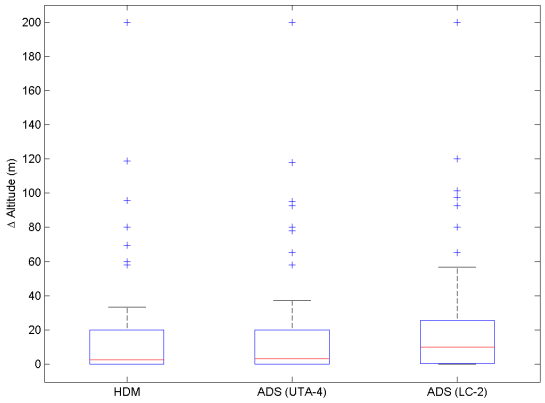


Fig. 13. Box plots comparing UAS platform $|g_z - s_z|$ for offline trajectories selected by HDM 3, ADS LC-2 and ADS UTA-4 solutions

It was found that the ADS with the inclusion of human expert data to model preferences, generated decisions which were closer to the HDM decisions captured using the GUI implementation (Figure 11). Thus, the inclusion of preferences formulated using captured HDM decision data allows for the automated generation of trajectory decisions which are similar to the decisions generated by the candidate HDM for each given decision scenario. The following section demonstrates the inclusion of HDM preferences derived using UTA to generate feasible trajectories which mimic aspects of the candidate HDMs decision process in 3D low altitude simulated environments.

V. Results

This section presents the automated generation of feasible trajectories through the concatenation of primitives using MA theory (Section II-B). The automated process mimics aspects of the HDM decision process through the inclusion of preferences formulated using UTA theory from HDM expert data captured.

A. Simulation setup

A simulated 3D terrain environment (figure 14) was setup in MATLAB to simulate mission scenarios where the UAS assignment includes safe and efficient navigation through a set of globally optimal waypoints. The simulation has been performed on a computer with an Intel Core 2 quad core processor operating at 2.8GHz to simulate how the inclusion of

Primitive Type	**Primitive No.** (m)	**Primitive Samples** (i)
Straight and Level	1	100
Coordinated Turn	12	100
Constant Climb	1	100
Helical Climb	12	100
Constant Descend	1	100
Helical Descend	12	100

TABLE I
PRIMITIVE TYPE, NUMBER AND SAMPLES PER PRIMITIVE APPLIED DURING ONLINE SIMULATIONS

human expert data to the motion planning problem can lead to the generation of UAS flight trajectories which mimic aspects of the HDM decision process.

The ADS is tasked with generating an optimized, feasible and collision free trajectory through all mission level waypoints until the goal is reached (Figure 14). The waypoints can either be selected by the user, or provided by a mission planner. The advantage of using a mission planner is that global optimality is guaranteed as the planner will generate a set of waypoints which are globally optimal with respect to a predefined set of criteria. We use a mission planning solution by Wu [33] for the low altitude trajectory planning results in simulated environments with terrain present.

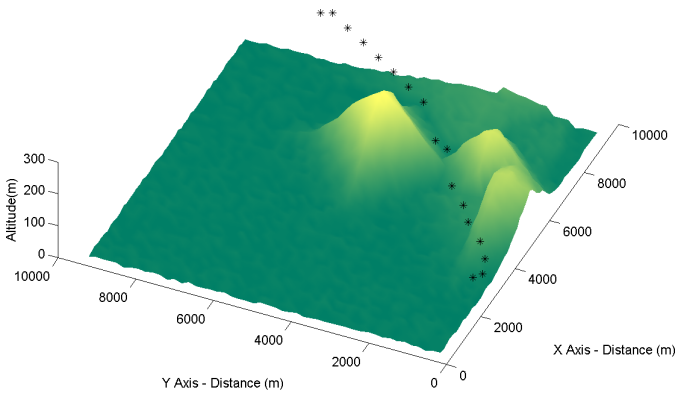


Fig. 14. Simulated mission environment (terrain simulation 1)

The ADS generates a set of alternatives for each stage by selecting the the number of primitives (m) and the samples per primitive (i) (Section III-B1). A large set of alternatives provides a greater number of final states which the platform can reach and a higher resolution of the region within the platforms performance bounds. Consequently, a large set of alternatives requires a greater computational effort and subsequently a longer time to plan. Table I lists the primitive types, (m) and (i) applied for this section.

B. Preference selection during online planning

During online trajectory planning, the automated decision algorithm compares the current online decision scenario to the set of decision scenarios presented to the candidate HDM offline (Figure 5). A least squares formulation (12) is applied to map the preference data for the offline decision scenario which most closely matches the current online decision scenario.

The least squares formulations compares the following differences between the current online scenario and offline scenario set (Figure 5); distance to goal in x, y and z dimensions $(\Delta x, \Delta y, \Delta z)$ and platform roll angle $(\Delta\phi)$. The least squares formulation for n offline scenarios becomes:

For $i \in [1..n]$

$$LSQR_k = \min\left(\sqrt{(\Delta x)_i{}^2 + (\Delta y)_i{}^2 + (\Delta z)_i{}^2 + (\Delta\phi)_i{}^2}\right) \tag{12}$$

where $\Delta x, \Delta y, \Delta z, \Delta\phi \in [0..1]$

The ADS applies the preferences from the offline HDM decision with the lowest least squares formulation value $(LSQR_k)$ to the weighted sum formulation (9) to generate an optimized solution. The following section presents the results of the online simulations where the automated trajectory mimics aspects of HDM decision styles through the inclusion of preferences formulated using HDM decision data via UTA theory.

C. Simulation results

High altitude operations in civilian airspace are generally conducted in IFR under the guidance of air traffic control. Whilst automated trajectory planning can still provide benefits for UAS platforms operating at high altitudes, low altitude operations can be considered as more challenging, as terrain must be treated as a hazard during planning and operations (Figure 16) .

Without the inclusion of collision avoidance methods, a safe output trajectory cannot be guaranteed, even with the application computed set of optimal collision free waypoints. For the inclusion of collision avoidance during 3D trajectory planning using MA theory, the terrain map data is used to cull trim primitives which are below a specified terrain height, at the given grid location (Figure 15). This ensures that an optimized collision free trim primitive can be selected for each stage from the remaining collision free set of primitives.

The automated LC-2 solution is used as a reference and compared the solution generated by the ADS with the inclusion of the candidate HDM's decision patterns through UTA theory. The comparative trajectory applies the candidate HDM's decision style through the inclusion of HDM preferences formulated using UTA-4.

1) Terrain Simulation 1: HDM 3's dataset was applied to UTA-4 and compared to the reference solution generated by LC-2 (Figure 16). Analysis of HDM 3's offline dataset showed that the HDM placed a greater preference on minimizing $\text{crit}_{(|g_z - s_z|)}$ (Figure 13). Subsequently, during online trajectory planning in simulated environments, UTA-4 generated collision free trajectories which had lower $|g_z - s_z|$ on average than LC-2 (Figure 17).

2) Terrain Simulation 2: HDM 2's dataset was applied to UTA-4 and compared to the reference solution generated by LC-2 (Figure 18). HDM 2 preferred to minimize platform ϕ variance during the offline simulation set (Figure 12). LC-2 has a higher preference for $\text{crit}_{(|\Delta\psi|)}$ which leads to the selection of manoeuvres which exhibit a low $c_{(|\Delta\psi|)}$ (6). This can result in the selection of primitives on the edge of the

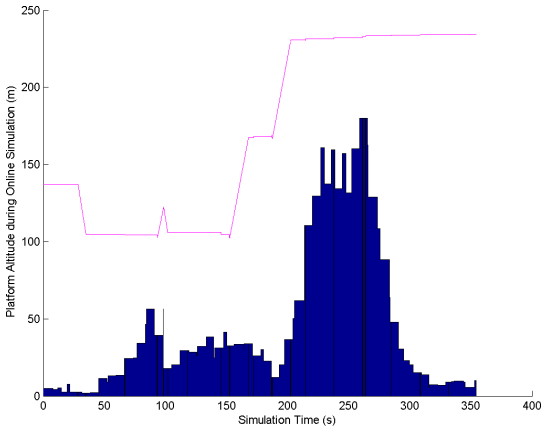


Fig. 15. UAS platform altitude during simulation (UTA-4 with HDM 3 dataset) (terrain simulation 1)

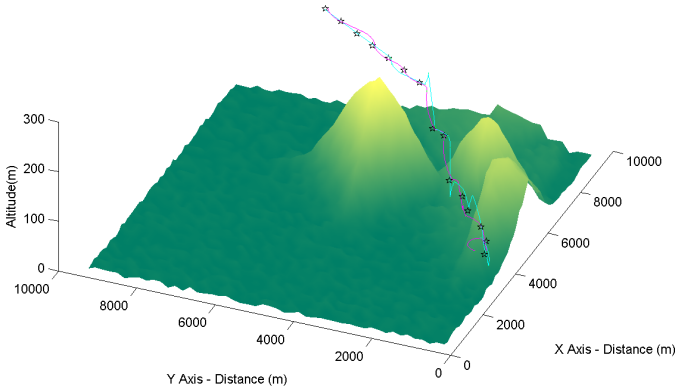


Fig. 16. Comparing trajectories from LC-2 solution and UTA-4 with HDM 3 dataset (terrain simulation 1)

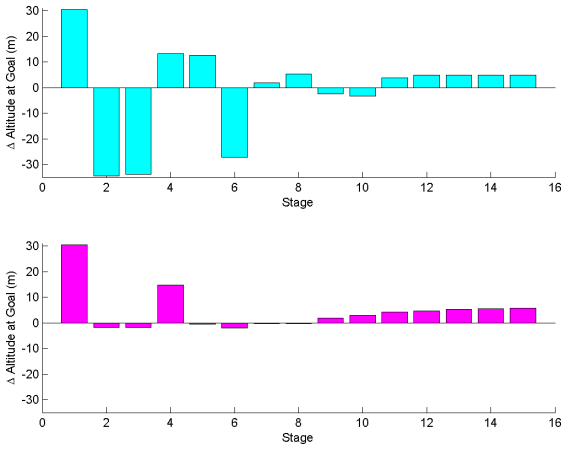


Fig. 17. Comparing UAS platform ΔAltitude at goal for LC-2 solution and UTA-4 with HDM 3 dataset (terrain simulation 1)

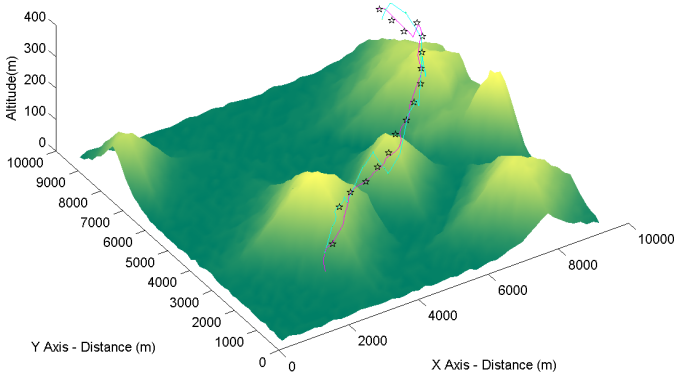


Fig. 18. Comparing Trajectories from LC-2 solution and UTA-4 with HDM 2 Dataset (terrain simulation 2)

platforms wing loading performance bounds as LC-2 does not explicitly consider $\text{crit}_{(|\Delta\phi|)}$ during optimisation. This can be viewed in Figure 19 where LC-2 exhibits higher maximum ϕ values than UTA-4.

VI. Discussion and Conclusions

This paper presented a new approach for the inclusion of human expert cognition into autonomous trajectory planning, for Unmanned Aerial Systems (UAS) operating in environments with terrain present. Expert decision data was gathered using a Graphical User Interface (GUI), allowing for the quantification of the human decision making process. Aspects of human cognition were applied to MA theory to generate feasible 3D collision free trajectories which were optimized to generate similar decisions with respect to the candidate HDM during autonomous operations.

It has been demonstrated that mission requirements and HDM decision styles can be better represented in automated trajectory planning systems through the inclusion of HDM decision data through the UTA MCDA technique. Using automated decision algorithms which apply human expert decision strategies may result in increase confidence in UAS operations over populated regions and potentially bring civilian UAVs closer to being operated autonomously in the NAS.

Acknowledgment

The author would like to thank and acknowledge the support of the Australian Research Centre for Aerospace Automation (ARCAA), the Queensland University of Technology (QUT) and Télécom Bretagne throughout this research project. The author would like thank and acknowledge Paul Wu for providing terrain data and the corresponding set of optimised mission waypoints. The author would also like to thank and acknowledge Troy Bruggeman, Duncan Greer, Ryan Fechney and Paul Zapotezny-Anderson for their participation in the collection of human expert decision data.

References

[1] S. Wegener, "Uav autonomous operations for airborne science missions," American Institute of Aeronautics and Astronautics, Tech. Rep., 2004.
[2] C. A. A. of Australia, "Advisory circular - uav operations, design specification, maintenance and training of human resources," Tech. Rep., July 2002.
[3] EUROCONTROL, "Specifications for the use of military unmanned aerial vehicles as operational air traffic outside segregated airspace," Tech. Rep., 25 April 2006.
[4] M. T. DeGarmo, "Issues concerning integration of unmanned aerial vehicles in civil airspace," MITRE, Center for Advanced Aviation System Development, Tech. Rep., 2004.
[5] P. Narayan, P. Wu, D. Campbell, and R. Walker, "An intelligent control architecture for unmanned aerial systems (uas) in the national airspace system (nas)," in *2nd International Unmanned Air Vehicle Systems Conference*, Grand Hyatt, Melbourne, Australia, 2007.

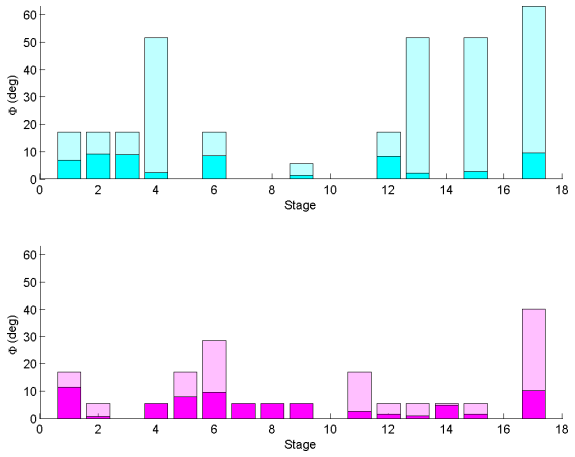


Fig. 19. Comparing UAS Platform Mean and Maximum ϕ (per stage) for LC-2 solution and UTA-4 with HDM 2 Dataset (terrain simulation 2)

[6] J. Boskovic, R. Prasanth, and R. Mehra, "A multilayer control architecture for unmanned aerial vehicles," in *Proceedings of the American Control Conference*, U. o. Cincinnati, Ed., vol. 3, pp. 1825–1830.

[7] R. P. Bonasso, D. Kortenkamp, D. P. Miller, and M. G. Slack, "Experiences with an architecture for intelligent, reactive agents," in *International Joint Conference on Artificial Intelligence*.

[8] P. Schaefer, R. Colgren, R. Abbott, H. Park, A. Fijany, F. Fisher, M. James, S. Chien, R. Mackey, M. Zak, T. Johnson, and S. Bush, "Technologies for reliable autonomous control (trac) of uavs," in *Digital Avionics Systems Conference*, vol. 1.

[9] P. Vincke, *Multicriteria Decision-Aid*. Wiley UK, 1992.

[10] J. Franke, V. Zaychik, T. Spura, and E. Alves, "Inverting the operator/vehicle ratio: Approaches to next generation uav command and control," June 2005.

[11] P. Meyer, "Progressive methods in multiple criteria decision analysis," Ph.D. dissertation, 2007.

[12] I. K. Nikolos, K. P. Valavanis, N. C. Tsourveloudis, and A. N. Kostaras, "Evolutionary algorithm based offline/online path planner for uav navigation," *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics*, vol. 33, no. 6, pp. 898–912, 2003.

[13] K. B. Judd and T. W. McLain, "Spline based path planning for unmanned air vehicles," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, vol. AIAA-2001-4238, Montreal, Canada, 2001.

[14] E. Anderson, R. Beard, and T. McLain, "Real-time dynamic trajectory smoothing for unmanned air vehicles," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 3, pp. 471–477, 2005.

[15] E. Gagnon, C. Rabbath, and M. Lauzon, "Heading and position receding horizon control for trajectory generation," in *The Proceedings of the American Control Conference*, 2005, pp. 134–139.

[16] Y. Kuwata, "Real-time trajectory design for unmanned aerial vehicles using receding horizon control," Master of Science, 2003.

[17] D. Rathbun, S. Kragelund, A. Pongpunwattana, and B. Capozzi, "An evolution based path planning algorithm for autonomous motion of a uav through uncertain environments," in *21st Digital Avionics Systems Conference. Proceedings*, vol.2 ed. USA Piscataway, NJ: IEEE, 2002.

[18] E. Frazzoli, M. Dahleh, and E. Feron, "A hybrid control architecture for aggressive maneuvering of autonomous helicopters," in *Proceedings of the 38th IEEE Conference on Decision and Control*, vol. 3, 1999, pp. 2471–2476.

[19] ——, "Maneuver-based motion planning for nonlinear systems with symmetries," *IEEE Transactions on Robotics and Automation*, vol. 21, no. 6, pp. 1077–1091, 2005.

[20] R. Bellman, "On the theory of dynamic programming." *National Academy of Sciences*, vol. 38, p. 716, 1952.

[21] T. Schouwenaars, B. Mettler, E. Feron, and J. How, "Robust motion planning using a maneuver automaton with built-in uncertainties," in *AIAA Aerospace Sciences and Exhibit*, Reno Nevada, 2003.

[22] L. Singh, J. Plump, M. McConley, and B. Appleby, "Software enabled control: Autonomous agile guidance and control for a uav in partially unknown urban terrain," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Austin, Texas, 2003.

[23] S. M. LaValle, *Planning Algorithms*. New York: Cambridge University Press, 2006.

[24] J. Figueira, S. Greco, and M. Ehrgott, *Multi Criteria Decision Analysis: State of the Art Surveys*. Boston: Springer, 2005.

[25] G. Gigerenzer, P. Todd, and A. R. Group, "Simple heuristics that make us smart," Tech. Rep., 1999.

[26] P. C. Fishburn, *Utility Theory for Decision Making*. New York: John Wiley and Sons, Inc., 1970.

[27] R. Keeney and H. Raiffa, *Decision with multiple objectives: Preferences and value tradeoffs*. Cambridge University Press, 1976.

[28] B. Roy, "Classement et choix en prsence de points de vue multiples (la mthode electre)," *la Revue d'Informatique et de Recherche Opérationelle (RIRO)*, vol. 8, pp. 57–75, 1968.

[29] E. Triantaphyllou, B. Shu, S. N. Sanchez, and T. Ray, "Multi-criteria decision making: An operations research approach," *Encyclopedia of Electrical and Electronics Engineering*, vol. 15, pp. 175–186, 1998.

[30] F. A. Administration, "Performance maneuvers," in *Airplane Flying Handbook: FAA-H-8083-3A*, 2nd ed. Aviation Supplies and Academics, Inc., 2004.

[31] C. Bana e Costa and J. Vansnick, "A theoretical framework for measuring attractiveness by a categorical based evaluation technique (macbeth)," in *XIth Int. Conf. on MultiCriteria Decision Making*, Coimbra, Portugal, 1994, p. 1524.

[32] E. Jacquet-Lagreze and J. Siskos, "Assessing a set of additive utility functions for multicriteria decision-making, the uta method," *European Journal of Operational Research*, vol. 10, pp. 151–164, 1982.

[33] P. P. Wu, D. A. Campbell, and T. Merz, "On-board multi-objective mission planning for unmanned aerial vehicles," in *IEEE Aerospace Conference*, Big Sky, Montana, 2009.

Pritesh Narayan completed his bachelors degree in Aerospace Avionics Engineering with first class honors at QUT in 2005. Pritesh is currently a PhD candidate at ARCAA QUT (Brisbane, Australia). His research investigates the inclusion of human expert cognition into autonomous trajectory planning, for UAS operating in environments with real time constraints present.

Patrick Meyer is currently an Associate Professor at Télécom Bretagne (France) and a member of the Lab-STICC laboratory of the French National Center for Scientific Research. Patrick obtained his PhD in Mathematics and Engineering Science, jointly from the Univ. of Luxembourg and the Engineering Faculty of Mons (Belgium) in 2007. He has also worked four years as a researcher at the Univ. of Liége (Belgium) and four years as a teaching and research assistant at the Univ. of Luxembourg. Patrick's interests include mathematical and methodological tools in MCDA and robustness issues in Data-Mining.

Duncan Campbell is an Associate Professor in the School of Engineering Systems at QUT and is also the Acting Director of ARCAA. Duncan is currently the President of Australasian Association for Engineering Education (AAEE) and previously served as IEEE Queensland Chair for the joint chapters of Control Systems, and Robotics and Automation (2008-2009). He also leads an international group on the internationalisation of the engineering curriculum.

Computationally Adaptive Multi-Objective Trajectory Optimization for UAS With Variable Planning Deadlines

Pritesh Narayan*, Duncan Campbell, Rodney Walker
*PhD Candidate
Queensland University of Technology
2 George Street, Brisbane
p.narayan@qut.edu.au

Abstract—This paper presents a new approach which allows for the computation and optimization of feasible 3D flight trajectories within real time planning deadlines, for Unmanned Aerial Systems (UAS) operating in environments with obstacles present. Sets of candidate flight trajectories have been generated through the application of manoeuvre automaton theory, where smooth trajectories are formed via the concatenation of predefined trim and manoeuvre primitives; generated using aircraft dynamic models. During typical UAS operations, multiple objectives may exist, therefore the use of multi-objective optimization can potentially allow for convergence to a solution which better reflects overall mission requirements. Multiple objective optimization of trajectories has been implemented through weighted sum aggregation. However, real-time planning constraints may be imposed on the multi-objective optimization process due to the existence of obstacles in the immediate path. Thus, a novel Computationally Adaptive Trajectory Decision (CATD) optimization system has been developed and implemented in simulation to dynamically manage, calculate and schedule system execution parameters to ensure that the trajectory solution search can generate a feasible solution, if one exists, within a given length of time. The inclusion of the CATD potentially increases overall mission efficiency and may allow for the implementation of the system on different UAS platforms with varying onboard computational capabilities. This approach has been demonstrated in th[1]is paper through simulation using a fixed wing UAS operating in low altitude environment[2]s with obstacles present.

TABLE OF CONTENTS

1. INTRODUCTION.....1
2. FEASIBLE TRAJECTORY REPRESENTATION 2
3. TRAJECTORY OPTIMIZATION 3
4. REAL TIME OPTIMIZATION 4
5. RESULTS 5
6. CONCLUSIONS 7
7. ACKNOWLEDGEMENTS 7
REFERENCES 8
BIOGRAPHY 8

1

[1] 1-4244-1488-1/08/$25.00 ©2009 IEEE
[2] IEEEAC paper#1237, Version 3, Updated 2008:11:03

1. INTRODUCTION

Unmanned Aerial Systems (UAS) have been previously employed in a diverse range of military applications. With respect to civilian applications, geographically sparse countries, such as Australia have great potential for utilization of UAS in asset management, search and rescue, remote sensing operations and atmospheric observation [1]. In order to realize this potential, seamless operation of UAS with the National Airspace System (NAS) is required [2, 3]; this is a difficult problem

Operation of UAS in the NAS creates a new set of challenges that are not applicable to many military applications. From a regulatory perspective, UAS need to: (i) demonstrate an Equivalent Level Of Safety (ELOS) to that of a human piloted aircraft, (ii) operate in compliance with existing aviation regulations and (iii) appear transparent to other airspace users [4].

The majority of UAS operations still require human operators to perform mission management and piloting tasks through real time communications links with the unmanned platform. This results in high operator workload and places greater reliance on the communications link. The inclusion of automated planning systems onboard can potentially improve mission efficiency and allow for continued operations in the presence of communications failures. In particular, the automation of global and local path planning components assist in ensuring that the flight occurs in accordance with the rules of the air; a key ELOS requirement.

Local path planning provides a navigation strategy for safe traversal through cluttered environments. The desired track, represented as a collision free flight trajectory, ensures that the platform remains within platform performance bounds. Automating the local path planning process is non-trivial and some challenges include: incorporation of complex platform dynamics, trajectory optimization to meet mission requirements, real-time constraints on computation time imposed by obstacles in the flight path, and the guarantee that generated trajectories are collision free.

During operations, civilian UAS may have multiple objectives to meet. The use of multi-objective optimization

allows the generation of a solution which better reflects the overall mission requirements. Additionally, if operations are undertaken at lower altitudes, the environment may present several challenges not encountered during high altitude flight. Terrain and urban structures become hazards to the safety of the UAS. The proximity of obstacles to the UAS places real-time constraints on re/planning computation time.

This paper presents a new framework for the Computationally Adaptive Multi-Objective Flight Management of UAS in civilian environments. An outline of UAS trajectory generation approaches and related work is given in section 2. Section 3 presents an overview of the trajectory optimization process, and section 4 outlines the real-time replanning requirements of UAS operating in cluttered requirements. Simulation results presented in section 4 demonstrate how the addition of the CATD can allow for the generation of feasible trajectories within given real-time deadlines. Finally, conclusions are presented in section 5.

2. FEASIBLE TRAJECTORY REPRESENTATION

A local path planning process is generally described as a system which generates a smooth trajectory representing the aircraft track through a set of mission level waypoints; typically generated by a global planner. The trajectory generated is required to be feasible and collision free to ensure that UAS flight track is safe and within platform performance bounds.

UAS Platform Constraints

The inclusion of vehicle dynamics during the trajectory planning process, allows for the generation of flight trajectories which take platform constraints into account. Vehicle dynamics are used to calculate the performance envelope which the aircraft must remain within to ensure that platform does not operate outside performance bounds. In the presence of a Stability Augmentation System (SAS) onboard, trajectories which do not consider platform performance bounds may lead to poor tracking.

Flight Trajectory Representation

Flight trajectories are generally represented through the use of either spline based or geometric approximations. Polynomial or spline based techniques [5, 6] place control points in a particular order to generate the desired trajectory. Geometric based techniques require the concatenation of aircraft flight manoeuvres to form a smooth flight track [7-10]. However, these flight manoeuvres are usually limited to cruise and constant radius turns and roll/yaw coupling effects are not considered; an essential flight characteristic of fixed wing platforms.

During the execution of a constant radius turn for a fixed wing aircraft, the consideration of roll/yaw coupling allows for the inclusion of platform roll rate as a component of the overall aircraft performance envelope. However, this requires the additional tracking of the platform attitude (roll component) during the trajectory planning process. One candidate method which allows for the inclusion of roll rate performance bounds is manoeuvre automaton theory.

Manoeuvre Automaton Theory

Manoeuvre Automaton (MA) theory, proposed by Frazzoli et al. [11, 12] can be used in the generation of feasible flight trajectories through sequential concatenation of predefined motion primitives. MA employs two types of primitives: trims and manoeuvres. Trim primitives represent the vehicle during a state of equilibrium whilst manoeuvre primitives characterize the vehicle operating outside a state of equilibrium. Primitives are generated using a dynamic model of the vehicle, thus platform stability can be implicitly guaranteed through generation of primitives which ensure that the vehicle remains within performance bounds.

Trajectory Representation Implementation

For this paper, MA theory is used to describe a time-invariant non-linear, dynamical system S, described as a set of ordinary differential equations (ODE) as:

$$\dot{x}(t) := \frac{d}{dt}x(t) = f(x(t), u(t)) \qquad (1)$$

Where u is the control input (execution time, manoeuvre type) = $\{\tau, primitive\}$ and x is the state vector.

Trim Primitive Representation

Trim Primitives represent the UAS platform operating in a state of equilibrium. Using MA theory, trim primitives can be generated by placing the body fixed roll $(\dot{\phi})$ and pitch $(\dot{\theta})$ rates to zero and maintaining a constant velocity (V), roll (ϕ) and pitch (θ) angle for the duration (τ_q) of the primitive execution. Trim primitives were generated using a 6 Degree of Freedom (DOF) flight dynamics model based on the Aerosonde UAS platform data set available in the Aerosim Blockset [13]. Six predefined trim primitives have been implemented in simulation including: cruise; coordinated turn, climb, descent, helical climb and helical descent.

The initial platform state $x(t_i) = x_i$ reaches a final state $x(t_f) = x_f$ due to the execution of a given trim primitive (q); this can be represented as:

$$x_f = x_i + \tau_q \dot{x}_q$$
$$t_f = t_i + \tau_q \quad (2)$$

Where $\{V, \phi, \theta\}$ are constants and $\{\dot{\phi}, \dot{\theta}\} = \{0,0\}$

It is of importance to note, that for a platform to enter a state of equilibrium (execution of a trim primitive), the initial platform attitude must equal the attitude requirements of the trim primitive to be executed; $\{\phi, \theta\}_i = \{\phi, \theta\}_q$. If the initial platform attitude does not equal the attitude required to execute the given trim primitive, a manoeuvre primitive must be inserted to ensure that body fixed attitude rate constraints are included as a performance bound.

Manoeuvre Primitive Representation

During the execution of a manoeuvre primitive, the UAS does not have to remain in a state of equilibrium. For a fixed wing platform, the body fixed attitude rate constraint becomes $\{\dot{\phi}, \dot{\theta}\} = \{\dot{\phi}_{\max}, \dot{\theta}_{\max}\}$. In this paper, manoeuvre primitives (p) are employed to connect two trim primitives, if required, in the formation of feasible trajectories. Furthermore, this allows for the consideration of attitude rates as an additional platform constraint during periods where the UAS is not in a state of equilibrium (e.g. switching between trim primitives where $\{\phi, \theta\}_i \neq \{\phi, \theta\}_q$).

If $\{\phi, \theta\}_i \neq \{\phi, \theta\}_q$, the UAS platform dynamic model is propagated until the platform reaches the desired state configuration $\{\phi, \theta\}_i = \{\phi, \theta\}_q$ making the execution of the next trim primitive feasible.

While $\{\phi, \theta\}_k \neq \{\phi, \theta\}_q$

$$x_{k+1} = x_k + \dot{x}_p \Delta T$$
$$t_{k+1} = t_k + \Delta t \quad (3)$$

Where $\{\dot{\phi}, \dot{\theta}\} = \{\dot{\phi}_{\max}, \dot{\theta}_{\max}\}$

The manoeuvre primitive required to switch between cruise and coordinated turn trim primitives is shown in Figure 1.

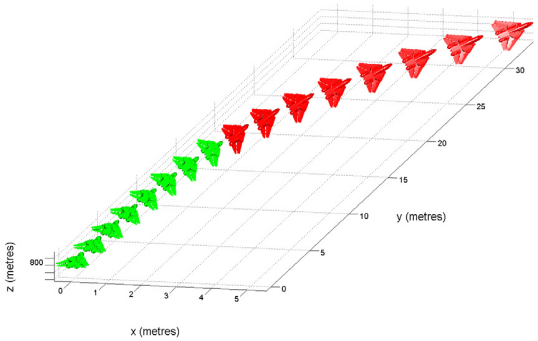


Figure 1 – Visual Representation of Trim and Manoeuvre Primitive Concatenation

Generating Collision Free Trajectories

Safe UAS operation in cluttered environments requires the generation of collision free trajectories. This has been accomplished through the inclusion of collision detection algorithms. The transition trajectory must be deemed collision free before collision detection along the candidate flight mode takes place. However, due to the sequential nature of manoeuvre concatenation, a collision free candidate trajectory does not guarantee vehicle safety during the next manoeuvre. Safe state manoeuvres [14] are executed at each sampled point along the candidate flight mode and then tested for collisions. This ensures that the UAS can enter a safe state if no collision free trajectory is determined during the optimization of the following stage (Figure 2).

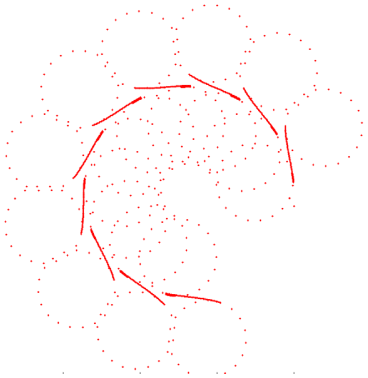


Figure 2 – Safe States Generated for a Candidate Coordinated Turn Trim Primitive

3. TRAJECTORY OPTIMIZATION

Dynamic programming (DP) [15] has been previously employed in related research [16, 17] for the optimization of feasible trajectories that have been generated using manoeuvre automaton theory. DP is a sequential optimization process where each trim primitive selected for execution can be considered a stage. Thus the final trajectory is formed through sequential concatenation of a set of selected trim primitives (and corresponding manoeuvre primitives, if required) for all stages used in the computation.

DP is a very computationally expensive algorithm for the motion planning application. In comparison to the application of DP to trajectory planning with respect to a generic scenario, the current UAS platform position can be treated as the current node. Each possible state the platform can reach through the execution of currently stored trim primitives must be treated as neighboring nodes. Expanding each neighboring node would cause the algorithm to grow exponentially in computational complexity for each additional stage considered in the overall optimization process. Additionally, due to the inclusion of manoeuvre primitives, it is difficult to calculate how many stages are required before a solution is found (if one exists).

In a typical UAS scenario, constant trajectory replanning maybe required if operations take place in partially known environments (e.g. active onboard sensing is predominantly used for navigation). To decrease the computational complexity and resulting time to plan during DP optimization over multiple stages, hybrid architectures involving DP with Rapidly Exploring Random Trees (RRT) [11] and DP with Model Predictive Control (MPC) [18] have been implemented.

The research presented in this paper uses DP search algorithm but limits the search to single stage optimization. This converts to a greedy search algorithm which essentially chooses the most optimal trim primitive, trim execution time and manoeuvre execution time required to execute the optimal trim primitive for each stage. The UAS position after execution of the optimal trim primitive is taken as the next node for expansion, and continues until a solution is found (**Error! Reference source not found.**).

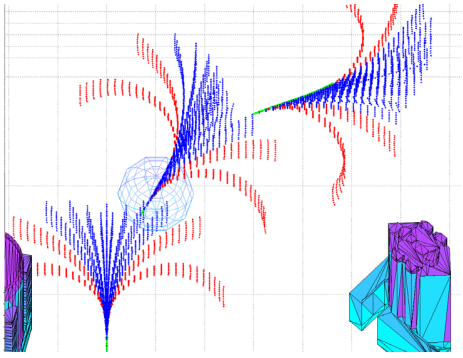


Figure 3 - Greedy Search Algorithm Implementation

Executing a DP search algorithm iteratively over each stage significantly decreases search time. However, not considering all stages during the optimization process means that global solution optimality and completeness cannot be guaranteed. Additionally, this may lead to scenarios where the platform becomes trapped in local minima. UAS motion planning in 3D space has the advantage for allowing the execution of certain motion primitives (e.g. helical ascent) to escape local minima and continue operations [19]. In addition, during operations in dynamic and partially known environments, a greedy motion planning implementation can suffice as it may not be possible to find a global solution (e.g. due to limited environment representation). Furthermore searching for a globally optimal solution may be infeasible as there can be real-time constraints placed on the replanning time, imposed by obstacles in the flight path.

Multiple Objective Optimization Process

During operations, civilian UAS may have multiple objectives to meet including platform safety; successful completion of the mission; minimizing fuel, time, and/or distance; or minimizing deviation from the current path. The use of multi-objective optimization allows for the generation of a solution which may better reflect the overall requirements of the mission. For example, by placing greater emphasis on safety, operations in populated environments may benefit from the inclusion of additional objectives which minimize platform control loss.

During each stage, the utility value is calculated using a weighted sum aggregation for all feasible trim primitives. The objectives included in the optimization process are, minimization of distance to goal and minimization of vehicle heading with respect to goal. Two additional objectives have been included to generate trajectories which are less likely to lead to loss of platform control. These objectives include: minimizing wing loading; and minimizing transition length required to execute next flight mode. The optimal solution for each stage is the trim primitive with the highest aggregated weighted sum value.

$$\mu_T = \sum_{i=1}^{n} w_i \mu_i \tag{4}$$

Where μ_T is the total utility value, w_i is the objective weighting and μ_i is the utility value objective utility value.

The following section provides an overview of the need for the inclusion of real time deadlines into the optimization.

4. Real Time Optimization

In the presence of real time deadlines, there is a finite length of time available (Finite Planning Window) for the UAS to complete the trajectory solution search before a predefined safety manoeuvre must be executed to ensure collision free flight. Convergence to a solution, if one exists, within this Finite Planning Window (FPW) is dependent on current system execution parameters and computational power available.

The time required to perform an optimal trajectory solution search during manoeuvre generation is dependent on system

execution parameters such as search resolution (number of primitives available); manoeuvre resolution (number of points representing primitive). Scenarios may occur where a feasible solution cannot be generated within the FPW if the search and resolution settings are too great. Consequently, solution completeness may be further diminished if the settings are too low.

A novel Computationally Adaptive Trajectory Decision (CATD) optimization system has been developed and implemented in simulation to dynamically manage, calculate and schedule system execution parameters. This ensures that the trajectory generator can complete the trajectory solution search and generate a feasible solution, if one exists, within the FPW.

CATD is an expert system which composed of two components. The offline component benchmarks the computational performance of the system using sets of predefined execution parameters. The computational performance can be estimated as the algorithm is deterministic in nature. However, the offline component must be re-executed if the computation capabilities of the system are modified.

The online component dynamically computes the most optimum set of execution parameters with respect to the available computational power and FPW. Multi-objective theory is used to find a best compromise solution where the conflicting objectives are maximization of search and resolution and minimization of search time.

The inclusion of the CATD potentially increases overall mission efficiency and may allow for the implementation of the system on different UAS platforms with varying onboard computational capabilities. The following section presents the results for the generation of feasible trajectories with the CATD both enabled and disabled.

5. RESULTS

A 3D environment representation was setup in MATLAB to simulate an urban scenario where the UAS assignment included safe and efficient navigation through a set of predefined mission level waypoints. The FPW is calculated as the time taken to complete the current stage. During the simulation the platform operates at a constant velocity of 30 m/s. The simulation has been performed on a computer with an Intel Core 2 quad core processor operating at 3.4GHz to simulate the how the inclusion of the CATD can allow for the generation of feasible trajectories within a given FPW. The FPW value is has a maximum value of ranging from 3 to 5 seconds to simulate a finite horizon (FH) between 90 and 150m

Simulated Results – CATD Not Enabled

The first set of results show the algorithms performance without the CATD enabled for each computing setup. The manoeuvre generation algorithm finds a feasible solution (Figure 4 and Figure 5) using a predefined set of manoeuvre and search resolution parameters (Table 1).

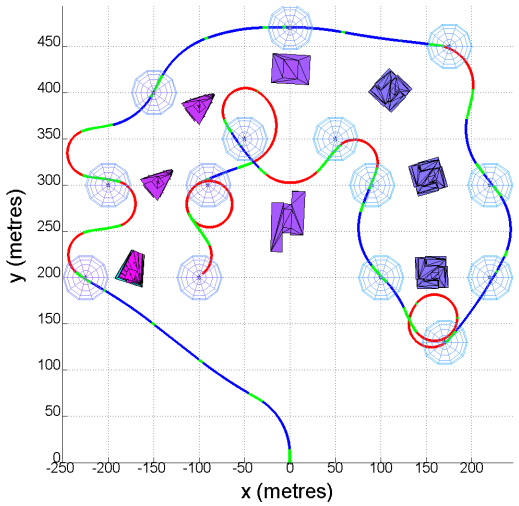


Figure 4 –Top View of Trajectory

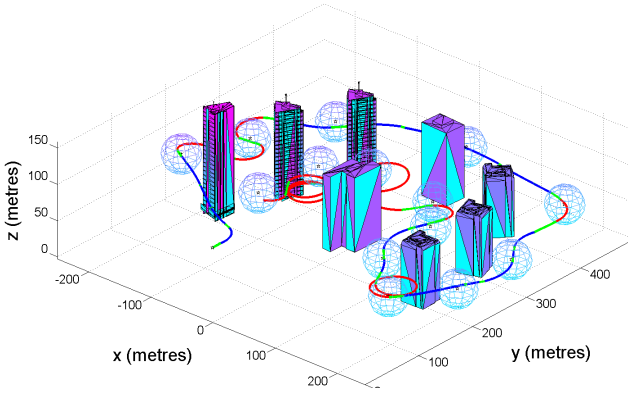


Figure 5 – 3D View of Trajectory

Table 1 – Algorithm Run Time: CATD Not Enabled

FH (m)	Manoeuvre Resolution	Search Resolution	Average Utility Value	Minimum FPW (s)
90	80	89	0.52	-0.7
120	80	89	0.52	0.1
150	80	89	0.52	1.2

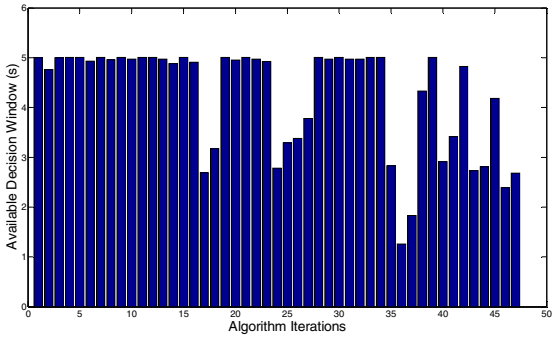


Figure 6 – FPW per Iteration (FH = 150m)

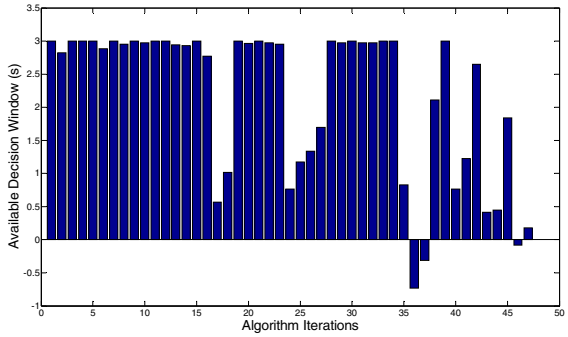


Figure 7 - FPW per Iteration (FH = 90m)

Without the CATD enabled, there is not guarantee that feasible trajectories will be generated within a given FPW. Using predefined search and manoeuvre resolution parameters may use of the computation time available inefficiently in scenarios where the FH is relatively large (Figure 6). In scenarios, where the given FH is shorter (Figure 7), the platform may not be able to compute a feasible solution within the available FPW.

Simulated Results – CATD Enabled

Enabling the CATD dynamically adjusts the manoeuvre and search resolutions with respect to the available FPW. Table 2 presents the results for the simulated results with the CATD Enabled.

Table 2 - Algorithm Run Time - CATD Enabled

FH (m)	Manoeuvre Resolution	Search Resolution	Average Utility Value	Minimum FPW (s)
90	Dynamic (Figure 10)	Dynamic (Figure 11)	0.93	1.6
120	Dynamic	Dynamic	0.93	0.3
150	Dynamic (Figure 14)	Dynamic (Figure 15)	0.9	0.5

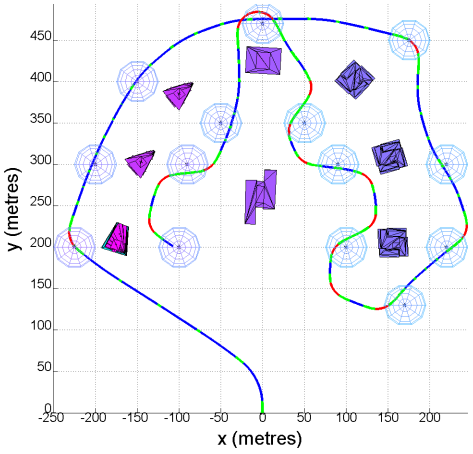


Figure 8 –Top View of Trajectory (FH = 150m)

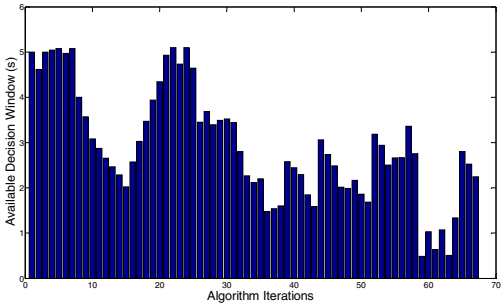


Figure 9 - FPW per Iteration (FH = 150m)

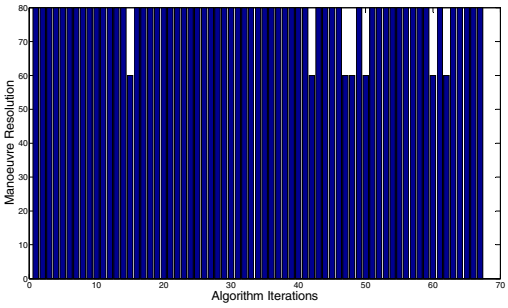


Figure 10 - Manoeuvre Resolution (FH = 150m)

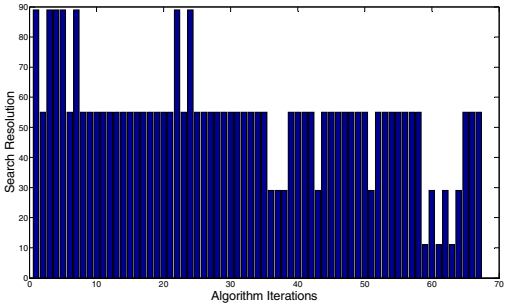


Figure 11 - Search Resolution (FH = 150m)

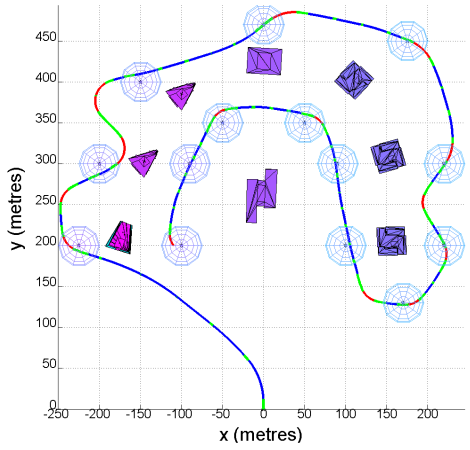


Figure 12 -Top View of Trajectory (FH = 90m)

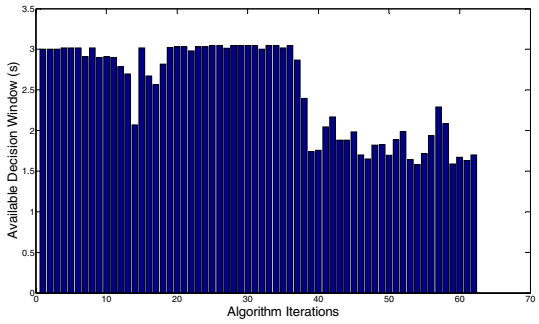


Figure 13 - FPW per Iteration (FH =90m)

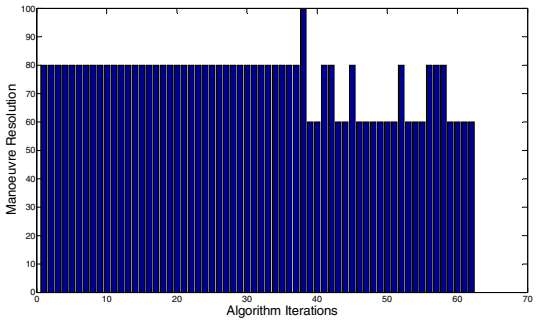


Figure 14 - Manoeuvre Resolution (FH = 90m)

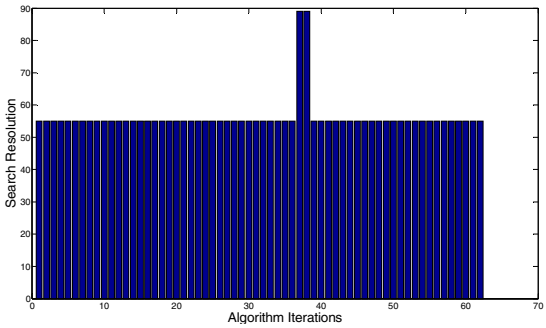


Figure 15 - Search Resolution (FH = 90m)

The inclusion of the CADT ensures that a feasible solution is generated within the given FPW. By dynamically adjusting the search and manoeuvre resolution parameters, the system compromises search completeness for time required to generate a solution. However, systems with greater onboard computational capabilities and/or longer FH (simulating onboard sensors) (Figure 9), benefit from the ability to complete a search at a higher resolutions. Systems without lower computational resources and/or Shorter FH can continue to generate feasible trajectory solutions (Figure 13) within the given FPW. This requires the search to be conducted at lower resolutions.

6. CONCLUSIONS

This paper has presented a new framework which allows for the computation and optimization of feasible 3D flight trajectories within real time planning deadlines, for UAS operations in cluttered environments. A novel real time flight management subsystem (CATD) was implemented to dynamically adjust manoeuvre and search resolution parameters to ensure that a feasible trajectory solution could be generated (if one existed) within a given FPW.

The inclusion of the CATD coupled to a multi-objective manoeuvre automaton based trajectory planner can potentially allows for more efficient use of the computational time available. Additionally, the utilization the offline component of the CATD to evaluate the performance of a given system, may potentially allow for the implementation of CATD on different platforms with varying onboard computational capabilities and Finite Planning Windows.

7. ACKNOWLEDGEMENTS

The author would like to thank and acknowledge the support of the Australian Research Centre for Aerospace Automation (ARCAA) and the Queensland University of Technology (QUT) throughout this research project.

REFERENCES

[1] S. Wegener, "UAV Autonomous Operations for airborne Science Missions," American Instititute of Aeronautics and Astronautics 2004.

[2] "Advisory Circular - Unmanned Aerial Vehicle Operations, Design Specification, Maintenance and Training of Human Resources," Civil Aviation Authority of Australia July 2002.

[3] "EUROCONTROL Specifications For The Use Of Military Unmanned Aerial Vehicles As Operational Air Traffic Outside Segregated Airspace," EUROCONTROL 25 April 2006.

[4] M. T. DeGarmo, "Issues Concerning Integration of Unmanned Aerial Vehicles in Civil Airspace," MITRE, Center for Advanced Aviation System Development, McLean, Virginia 2004.

[5] K. B. Judd and T. W. McLain, "Spline based path planning for unmanned air vehicles " in *AIAA Guidance, Navigation, and Control Conference and Exhibit,*. vol. AIAA-2001-4238 Montreal, Canada, 2001

[6] I. K. Nikolos, K. P. Valavanis, N. C. Tsourveloudis, and A. N. Kostaras, "Evolutionary algorithm based offline/online path planner for UAV navigation," *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics,* vol. 33, pp. 898-912, Dec 2003.

[7] E. P. Anderson, R. W. Beard, and T. W. McLain, "Real-time dynamic trajectory smoothing for unmanned air vehicles," *IEEE Transactions on Control Systems Technology,* vol. 13, pp. 471-477, 05 Jan 2005.

[8] E. Gagnon, C. A. Rabbath, and M. Lauzon, "Heading and position receding horizon control for trajectory generation," in *The Proceedings of the American Control Conference*, 2005, pp. 134-139.

[9] Y. Kuwata, "Real-time Trajectory Design for Unmanned Aerial Vehicles using Receding Horizon Control," in *Aeronautics and Astronautics* Massachusetts: Massachusetts Institute of Technology, 2003.

[10] D. Rathbun, S. Kragelund, A. Pongpunwattana, and B. Capozzi, "An evolution based path planning algorithm for autonomous motion of a UAV through uncertain environments," in *21st Digital Avionics Systems Conference. Proceedings (Cat. No.02CH37325)*, vol.2 ed USA Piscataway, NJ: IEEE, 2002, pp. 8D2-1-12.

[11] E. Frazzoli, M. A. Dahleh, and E. Feron, "A hybrid control architecture for aggressive maneuvering of autonomous helicopters," in *Proceedings of the 38th IEEE Conference on Decision and Control*, 1999, pp. 2471-2476 vol.3.

[12] E. Frazzoli, M. A. Dahleh, and E. Feron, "Maneuver-based motion planning for nonlinear systems with symmetries," *Robotics, IEEE Transactions on [see also Robotics and Automation, IEEE Transactions on],* vol. 21, pp. 1077-1091, 2005.

[13] "Aerosim Blockset," Unmanned Dynamics, 2003.

[14] T. Schouwenaars, J. How, and E. Feron, "Receding horizon path planning with implicit safety guarantees," in *Proceedings of the American Control Conference*, 2004, pp. 5576-5581 vol.6.

[15] S. M. LaValle, *Planning Algorithms*. New York: Cambridge University Press, 2006.

[16] T. Schouwenaars, B. Mettler, E. Feron, and J. How, "Robust Motion Planning Using a Maneuver Automaton with Built-In Uncertainties," in *AIAA Aerospace Sciences and Exhibit*, Reno Nevada, 2003.

[17] L. Singh, J. Plump, M. W. McConley, and B. D. Appleby, "Software Enabled Control: Autonomous Agile Guidance and Control for a UAV in Partially Unknown Urban Terrain," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Austin, Texas, 2003.

[18] T. Schouwenaars, B. Mettler, E. Feron, and J. How, "Hybrid Model for Trajectory Planning of Agile Autonomous Vehicles," *Journal of Aerospace Computing, Information and Communication,* vol. 1, Dec 2004.

[19] D. Anisi, J. Robinson, and P. Ögren, "On-Line Trajectory Planning for Aerial Vehicles: A Safe Approach with Guaranteed Task Completion," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, Colorado, 2006.

BIOGRAPHY

Pritesh completed his bachelor's degree in Aerospace Avionics Engineering with first class honors at QUT in 2005. His final year project included the development of autonomous capabilities for unmanned airborne platforms. His PhD research is focused on the generation of feasible flight trajectories with multi-objective optimization in scenarios with finite replanning time deadlines.

Unmanning UAVs – Addressing Challenges in On-Board Planning and Decision Making

Pritesh Narayan, Paul Wu, Duncan Campbell

Australian Research Centre for Aerospace Automation (ARCAA)
Queensland University of Technology
2 George St, Brisbane, Australia
da.campbell@qut.edu.au

Abstract. Planning and decision making, especially the planning of dynamically negotiable collision free paths, is an integral part in the operation of Unmanned Aerial Vehicles (UAVs). Effective path planning ensures that the UAV operates safely, and conforms to the rules and regulations governing flight within the National Airspace System (NAS). To demonstrate an Equivalent Level Of Safety (ELOS) to that of piloted aircraft for certification purposes, UAVs must demonstrate a high level of autonomy without a human in the loop. This research surveys the literature as to how human experts perform planning tasks and forms a framework which promotes shared authority of UAV mission (re)planning and path planning, and can adopt sole authority should the UAV communications link fail or the human operator relinquishes decisions. It has been demonstrated through simulation that the optimization of flight manoeuvre sets using multiple objectives allows for convergence to a solution which better represents civilian mission requirements whilst emulating common flight patterns of trained pilots. These initial findings highlight the challenges involved in replicating the skills of human pilots onboard a UAV. It is revealed that UAV planning and decision making is a multi-disciplinary problem that combines the fields of path planning (search optimization), trajectory generation, and human cognition

1 Introduction

Unmanned Aerial Vehicles (UAVs) have been employed, with great effectiveness, in a diverse range of military applications. However, geographically sparse countries, such as Australia, have great potential for utilization of UAVs in a wide range of civilian applications. These include asset management, search and rescue, and remote sensing. In order to realise this potential, it is necessary to gain access to the National Airspace System (NAS).

Operation of UAVs in the NAS creates a set of challenges not applicable to many military applications. From a regulatory perspective, UAVs need to: (i) demonstrate an Equivalent Level Of Safety (ELOS) to that of a human piloted aircraft, (ii) operate in compliance with existing aviation regulations and (iii) appear transparent to other airspace users [1]. Additionally, for the majority of current UAV operations, the

human operator acts as both the mission manager and the pilot using a real time communications link [2]. This results in high operator workload and places great reliance on the communications link.

Path planning assists in ensuring that the flight is operated in accordance with the rules of the air. The inclusion of automated planning systems onboard can potentially improve mission efficiency and reduce the need for laborious input from a ground-based human operator. This avoids problems associated with communications link failures and operator fatigue. UAV path planning can be considered in terms of global (mission) planning and local (trajectory) planning. This paper outlines the challenges involved in both types of planning and reviews studies on how human pilots currently perform these tasks. In light of these findings, candidate planning algorithms are identified to replicate human planning and decision making.

2 Global Planning

Global planning is concerned with finding a flight plan that minimises a cost function. Flight plans typically follow the standard profile shown in **Fig. 1**. The en-route flight plan comprises a series of waypoints, assumed to be joined by straight line trajectory segments, originating at the climb phase just after takeoff and terminating at the descent phase prior to approach. For the purposes of this paper, it is assumed that the UAV operates under Visual Flight Rules (VFR) as many civil applications (e.g. crop dusting) are performed under VFR [3].

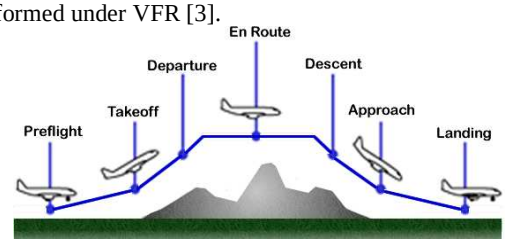


Fig. 1. Standard flight profile [4]

2.1 Path Planning and Sequential Decision Making

It has been shown that the path planning problem is of PSPACE complexity [5]. This complexity arises due to the exponential increase in memory and computation time with dimensionality. In 3D flight planning, the problem is further compounded by the size of the search space (due to the flight range of UAVs) and the need to optimise for multiple objectives (such as fuel, risk and rules of the air) [6]. Therefore, it is of value to study and replicate the cognitive skills of human expert pilots given their proficiency at flight planning [7]. Conventional path planners are complex, incomplete and computationally costly [8]. Replication of decision strategies (as

opposed to direct replication of human knowledge which is difficult [9]) of human experts can help create a planning framework that is more efficient. Additionally, this provides a high degree of cognitive compatibility which increases the system's usefulness in terms of design and operation.

The flight planning problem can be modelled as a sequential decision process where actions are chosen to maximally satisfy multiple designated objectives [5, 8]. These decisions are not independent as later decisions are constrained by earlier decisions. Furthermore, the decisions need to be made in real time [8].

Typical path planning methods model this sequential decision process through the dynamic programming recurrence equation [5]:

$$g(s_{k+1}) = g(s_k) + c(s_k, s_{k+1}) \quad \textbf{(1)}$$

where $s \in S$ is a node in the 3D search space, s_{k+1} is a child node to s_k (the parent), g is the total cost to reach a node from the start node s_1, and $c(s_k,s_{k+1})$ is the edge cost, i.e. the transition cost of moving from s_k to s_{k+1}. Methods such as A* iteratively evaluate nodes in the search space and calculate the cost g to neighbouring nodes until the minimum cost g^* for the goal node s_g has been found. From (1), it can be seen that the cost of each node s is a summative accumulation of individual decision outcomes from the start node.

Ideally, the multi-objective sequential decision making process should be conducted in decision space (which includes, in addition to x, y, z, variables like fuel and risk). Unfortunately, this is computationally challenging on small aircraft due to the PSPACE complexity of path planning [5]. It is common practice (e.g. [10, 11]) to "aggregate" the decision variables into a single cost variable. Thus, the optimal path is in actuality the least aggregated cost path.

However, the majority of human pilots, when equipped with the appropriate decision interface, are capable of planning satisficing paths that are at worst 5% more expensive from an "optimal" path generated by a computer [8]. Therefore, it is instructive to examine the cognitive strategies of human pilots for the purpose of flight planning.

2.2 Pilot Decision Model

It has been found that human pilot decision making can be described with 'non-rational' or naturalistic decision making models [12]. This form of decision making is characterised by the concept of *bounded rationality* [13]. Studies have shown that humans characteristically focus only on three to four categories of attributes (less than ten variables), adopt *non-compensatory* decision strategies (especially when under duress), and process only a few decision alternatives [14].

Additionally, studies have revealed that expert pilots predominantly employ intuition based decision making but also include some elements of analytical decision making

[9]. Intuition can be defined as "knowledge based on experiences and acquired through sensory contact" [9]. One way of characterising this form of decision making is through the *Recognition Primed Decision Making* (RPDM) model. This model is in actuality an intuitive form of diagnosis and prediction which can be surmised as (i) recognition (pattern matching), (ii) serial evaluation (generating situational awareness) and (iii) mental simulation. Thus, the expert pilot employs pattern matching using experience honed cues (effectively a form of *a priori* knowledge) to structure the decision process. This then activates conditional IF THEN rules which produce the final decision outcome [9, 12].

It has been observed that human pilots frequently make use of rules and procedures in their decision making processes [7, 9, 12]. This in part stems from the vigorous training of procedures and aviation rules. Additionally, human pilots also manage the weighting and selection of rules, attributes and even search cues based on the overall situational awareness; this is known as meta-cognition [12, 15]. Rasmussen's model [16] provides a holistic framework that captures both the RPDM and meta-cognitive elements of human pilot cognition. The CASSY [17] aviation decision support system is based on Rasmussen's model.

Using Rasmussen's model, the decision making component (each evaluation of (1)) of the flight planning task can be described as shown in **Fig. 2**. At each increment in the flight plan, decision variables are extracted from sensor data, Geographic Information Systems (GIS), weather and air traffic information and from the aggregated cost of previous decisions $g(s_k)$. These variables form the antecedents for IF THEN rules for RPDM and meta-cognition. Therefore, the cost function needs to be a multi-objective evaluation function capable of implementing multiple rules in a hierarchical manner. A candidate method for this would be fuzzy inferencing [6].

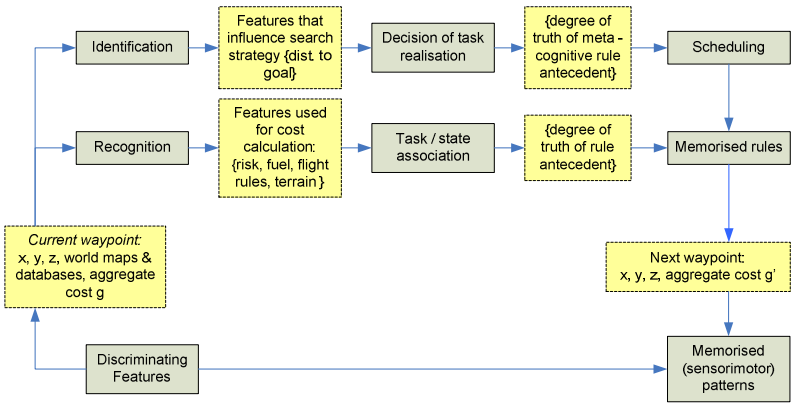


Fig. 2. Rasmussen's 3 layer model [16] depicted as a data flow diagram for flight planning

An important component in the decision making process above is the use of heuristics. Human pilots often employ heuristics, a category of cognitive processes whose primary role is to reduce the search space and thus speed up the decision process [12]. The heuristic is a meta-cognitive approach that can be used to prioritise the sequential decision process (i.e. choose which regions of the search space to explore first). Some heuristics, such as representativeness, availability and bias can adversely affect the solution outcome [18]. A useful heuristic, however, is adjustment and anchoring. With this heuristic the search process is seeded with an initial guess which is then adjusted based on available situational awareness information. Adjustment and anchoring is well suited to flight planning as flight plans predominantly follow the standard flight profile as shown in **Fig. 1** [18].

The anchoring and adjustment heuristic can be implemented with a heuristic search algorithm such as A*. In A*, the search process is prioritised according to a heuristic cost term:

$$f(s) = g(s) + h(s, s_g) \tag{2}$$

where h is a heuristic estimate of the cost to go from s to the goal s_g and f is the total cost (s_I to s_g). Therefore, through careful selection of h, it is possible to bias the search towards the standard flight profile.

2.3 Using Cognitive Techniques in Path Planning

The previous review of literature concerning pilot decision making has established three key points: (i) pilots tend to find a satisficing rather than an optimal path, (ii) pilots employ pattern matching (IF THEN rules, or production rules [9]), and (iii) heuristics aid in culling the search space. Therefore, heuristic search algorithms such as A* can be used as a suitable starting point for replicating human pilot planning. These algorithms have been used extensively in mobile robotics [19]. However, anytime replanning variants of A*, such as ARA*, are even better suited as, through adjustment of a heuristic inflation factor ε, it is possible to quickly find a satisficing solution. Furthermore, the solution path has a total cost of at most ε-times the optimal path cost [20]. Thus, if time is available, it is possible to iteratively decrease ε until ε<1 which gives the optimal solution.

Fig. 3 depicts an implementation of A*, showing a solution path in a complex environment. The decision variables for this investigation, based on VFR operation, are: (i) altitude Above Ground Level (AGL), (ii) airspace type, (iii) population risk (fatality risk per flight hour presented to people on the ground [21]), (iv) fuel consumed, and (v) weather (wind and storm cells). The search algorithm uses the framework presented in [6] for integrating a multi-criteria cost function into a path planner.

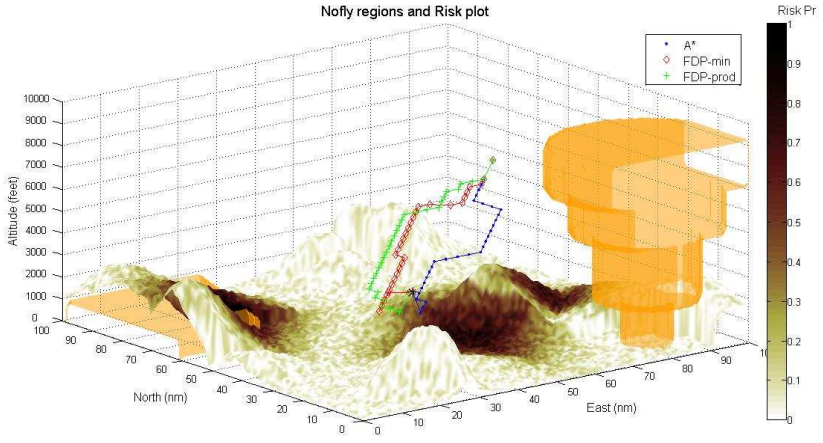


Fig. 3. Example flight paths using A*, Fuzzy Dynamic Programming (FDP) with min t-norm, and with product t-norm. Controlled airspace, and population risk shown.

The problem with A* like algorithms is identified in (1). The summative aggregation of prior decision outcomes means that decisions are aggregated using a disjunctive operator [22]. Therefore, as is highlighted in **Fig. 3**, there are cases where A* chooses a path with highly undesirable segments (i.e. high incremental cost) because the resultant summed cost is low. Oftentimes, it is desirable to avoid these high incremental cost paths unless if no other alternatives exist.

One method for addressing this shortcoming is to employ Fuzzy Dynamic Programming (FDP) [22]. Here, the sequential decision process is tracked using a conjunctive or t-norm operator:

$$\mu_{G_{i+1}}\left(x_{i+1}\right) = \max_{u_i}\left(\mu_{C_i}\left(u_i\right) \wedge \mu_{G_i}\left(x_i\right)\right) \tag{3}$$

At each decision step along a path, the utility value $\mu_G\left(x_{i+1}\right)$ of a state x_{i+1}, is found by the t-norm ($\wedge$) of the parent utility value $\mu_{G_i}\left(x_i\right)$, and the state transition action u_i, that gives maximal $\mu_G\left(x_{i+1}\right)$. The transition action u is transformed into a utility value using a constraint Membership Function (MF) $\mu_{C_i}\left(u_i\right)$. Note that fuzzy dynamic programming is cast in terms of a utility value, which is simply the negative of the cost.

Two FDP t-norm operators are evaluated against A* - the min and the product operators. The resultant paths are also shown in **Fig. 3**; note these paths avoid the higher risk regions. Over a number of simulations, it is unsurprising to find that the FDP methods find paths with lower maximum incremental path costs (**Fig. 4**). However, when using the min t-norm, the solution paths are significantly longer. This occurs because the min operator is more pessimistic and does not allow for

compensation between the constraints and the goals [22]. On the other hand, the product operator tends to find paths with a better balance between incremental path cost and path length. Unfortunately, both FDP methods take longer computation time than A*, and this is due to the fact that the current FDP framework doest not include a heuristic component to guide the search.

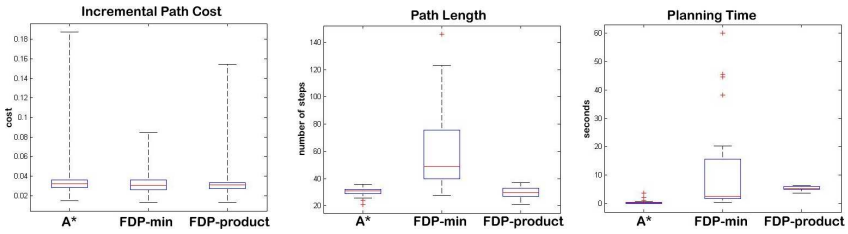


Fig. 4. Box and whiskers plots, showing inter-quartile range for incremental path costs, path length, and planning time

This preliminary investigation into automation of UAV flight planning has revealed that there are benefits in replicating human expertise. A survey of existing studies on human pilot cognition reveals that pilots predominantly rely on RPDM (pattern matching) and heuristics. This can be modelled specifically for flight planning using an adaptation of Rasmussen's three layered cognitive model. In turn, this sequential decision model can be replicated using A* or fuzzy dynamic programming; by using a product t-norm operator, a path that mimics human expectations is found. However, there remain many challenges that need to be addressed. These include evaluation of suitable multi-criteria cost functions (e.g. [6]), study of suitable heuristics and incorporation of heuristics into fuzzy dynamic programming. Unlike A*, the existing fuzzy dynamic programming framework does not include a heuristic term (3).

3 Local Planning

Local planning provides a navigation strategy for safe traversal through cluttered environments. This can be represented as a collision free flight trajectory which ensures that the platform remains within performance bounds. The implementation of local planning systems onboard UAV platforms has numerous benefits including overcoming potential ground station link issues. However, automating the local planning process is non-trivial and some challenges include: incorporation of complex platform dynamics, optimisation of trajectory to meet mission requirements, real-time constraints on computation time imposed by obstacles in the flight path, and the guarantee that trajectories generated are collision free. The following section presents a brief overview on flight trajectory representation.

3.1 Flight Trajectory Representation

A flight trajectory typically represents the desired motion of the aircraft during transversal between two points in airspace (i.e. current and goal position). The inclusion of vehicle dynamics during the trajectory planning process, allows for the generation of flight trajectories which take platform constraints into account.

Vehicle dynamics are used to calculate the performance envelope which the aircraft must remain within to ensure vehicle stability during flight. The types of aircraft performance bounds which can be included during the trajectory planning process is dependent on the number of states used for trajectory representation (e.g. position, velocity, acceleration, attitude, attitude rates). A 3 Degree Of Freedom (DOF) trajectory representation can allow for the inclusion of multiple aircraft performance bounds including: min (stall) and max velocities, min turn radius, and max climb and descent rates. However, a more complex 6 DOF trajectory representation is required for the inclusion of attitude rate constraints (e.g. max roll rate).

An example of flight trajectory representation is through the use of polynomial or spline based techniques [23, 24], where control points can be placed in a certain order to generate the desired trajectory. The use of polynomial or spline curve approximation limits trajectory representation to only 3 DOF. Without attitude and attitude rate state information, it is not possible to guarantee that the aircraft motion remains within platform performance bounds; in particular, the attitude rate constraints.

3.2 Trajectory Generation using Manoeuvre Automaton Theory

Manoeuvre Automaton theory is a published approach [25], where smooth feasible flight trajectories are formed via concatenation of predefined trim and manoeuvre primitives. Generating trajectories using manoeuvre automaton theory allows the inclusion of attitude information (roll, pitch and yaw) for trim manoeuvres and attitude rate information for manoeuvre primitives. This ensures that the trajectory generated is within vehicle performance bounds. Furthermore, trim and manoeuvre primitives can be configured to emulate flight manoeuvres performed by trained pilots (e.g. coordinated turn). The following sections outlines the implementation of manoeuvre automaton theory to generate smooth trajectories for fixed wing UAS in 3D space.

3.2.1 Trim Primitives

Six predefined trim primitives (referred to as flight modes) have been implemented in simulation including: cruise; flat turn, climb, descent, helical climb and helical descent. The flight dynamics model is based on the Aerosonde UAV data set available in the Aerosim Blockset [26].

3.2.2 Transition Primitives

A transition primitive has been implemented to ensure that the platform remains with performance boundaries while switching between flight modes. The UAV platform dynamic model is propagated until the UAV reaches the desired state configuration for execution of the next flight mode. The transition manoeuvre required to switch from cruise to coordinated turn flight modes in shown in **Fig. 5.**

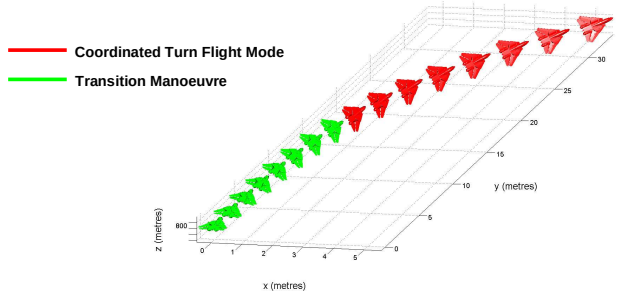


Fig. 5. Transition Manoeuvre Linking Cruise and Coordinated Turn Flight Mode

3.3 Trajectory Optimisation

Dynamic programming has been previously employed in related research [27, 28, 29] for the optimization of feasible trajectories that have been generated using manoeuvre automaton theory. Dynamic programming is a sequential optimization process and is appropriately suited to this particular optimization problem (referred to as manoeuvre generation) since only one flight mode can be executed at any one time.

Traditionally, trajectory generation techniques converge to near/optimal solutions by minimizing a singular cost function (e.g. fuel, time, distance). However, during each mission; civilian UAS may have multiple objectives to meet including platform safety; successful completion of the mission; minimizing fuel, time, and/or distance; or minimizing deviation from the current path. The use of multi-objective optimization allows the generation of a solution may better reflect the overall requirements of the mission.
The manoeuvre generation process was implemented in simulation using MATLAB to demonstrate how the inclusion additional objectives can potentially lead to the generation of trajectories which better represent overall mission requirements. A 3D environment representation was setup to simulate an urban scenario, where the UAV assignment included safe and efficient navigation through a predefined set of waypoints.

3.3.1 Single Objective Optimisation

To ensure mission completion; single objective optimization of trajectories generated through manoeuvre generation have been limited to distance minimization. Essentially, the optimal solution (per iteration) is the candidate flight manoeuvre which, once executed, minimizes the distance required to travel to the goal.

Simulated results for a single objective manoeuvre generation scenario are presented in (**Fig. 6)**.

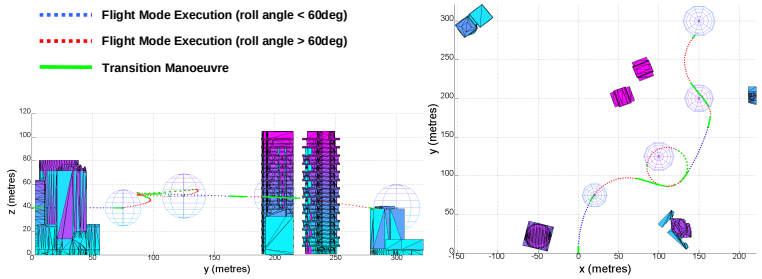


Fig. 6. Simulated Results for 3D Manoeuvre Generation using single objective optimisation

Single objective optimization during manoeuvre generation only considers the distance remaining to goal after flight mode execution. This may lead to the generation of trajectories which do not adequately satisfy mission requirements. For example, the trajectory generated in simulation (**Fig. 6)** requires the execution of flight modes approaching the performance limits of the platform; placing the vehicle at greater risk to loss of controllability. Thus, the solution generated may not be deemed acceptable if flight safety was an important mission requirement. The inclusion of additional objectives during the optimization process can potentially provide a better representation of overall mission requirements. The following section presents simulated results for multi-objective optimization of manoeuvre generation with respect to civilian operations.

3.3.2 Multi-Objective Optimisation

Loss of platform control can potentially result in collision with the surrounding environment. The consequences may be greater if UAV operations are undertaken in populated regions. Thus, operations in populated environments may benefit from the inclusion of objectives place a greater emphasis on safety by minimizing platform control loss during manoeuvre generation.

Two additional objectives have been included in the optimisation process of the simulated urban scenario to generate trajectories which are less likely to lead to loss of platform control. These objectives include: minimizing wing loading; and minimizing transition length required to execute next flight mode. The wing loading minimization objective gives a greater utility value to candidate flight modes which maintain a lower roll angle during execution since more controller power is available to recover from unexpected disturbances (e.g. wind gust). The transition length minimization objective gives a greater utility value to candidate flight modes which require shorter transition manoeuvres before execution, thus potentially decreasing platform instability due to coupling between lateral and longitudinal responses [30].

Fig. 7 presents simulated results after the inclusion of wing loading minimization objective to the single objective optimization process. Additionally, **Fig. 8** presents

simulated results for the inclusion of transition length minimization to the single objective optimization process. Finally, **Fig. 9** presents simulated results for the inclusion of both objectives to the optimisation process.

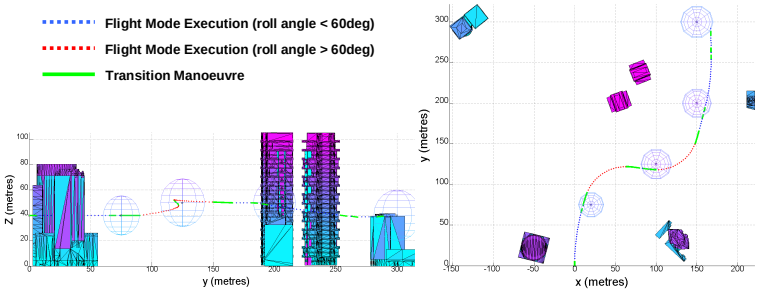


Fig. 7. Inclusion of wing loading minimization objective to optimisation process

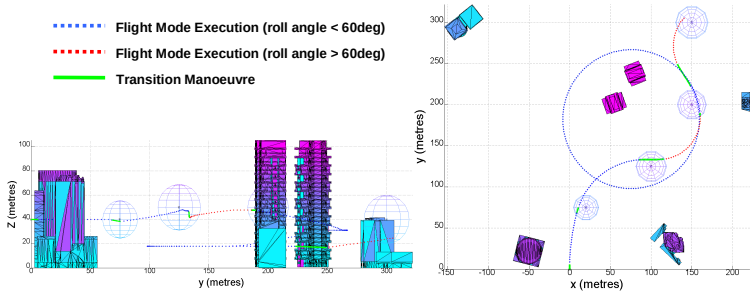


Fig. 8. Inclusion of transition length minimization objective to optimisation process

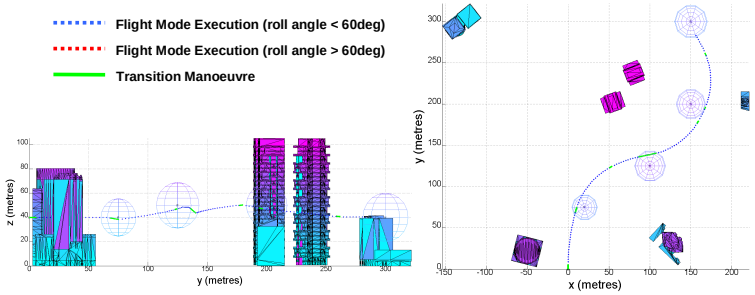


Fig. 9. Inclusion of wing loading and transition length minimization objectives

4 Conclusions

The research presented in this paper demonstrates the multi-disciplinary nature of UAV planning and decision making. Despite the complexity of flight planning and trajectory generation, human pilots perform such tasks with proficiency. A survey of existing studies on human pilot cognition revealed that human cognition can be modelled using Rasmussen's three layered structure. The paper presented some initial findings in replicating this model using A* and fuzzy dynamic programming. Additionally, it has been shown through simulation that optimization of flight manoeuvres can be used to emulate common flight patterns of trained pilots. Inclusion of multiple objectives mimicking human decision making results in trajectories that better match mission requirements.

This initial work presented here paves the way for future research into replication and modelling of human cognition with planning algorithms for UAV operation. Future work includes evaluation of suitable multi-criteria cost functions, study of suitable heuristics and incorporation of heuristics into fuzzy dynamic programming.

Bibliography

1. DeGarmo, M.T.: Issues Concerning Integration of Unmanned Aerial Vehicles in Civil Airspace. MITRE, Center for Advanced Aviation System Development, McLean, Virginia (2004)
2. Wegener, S.S., Schoenung, S.S., Totah, J., Sullivan, D., Frank, J., Enomoto, F., Frost, C., Theodore, C.: UAV Autonomous Operations for Airborne Science Missions. AIAA 3rd "Unmanned Unlimited" Technical Conference, Workshop and Exhibit. American Institute of Aeronautics and Astronautics, Chicago, Illinois (2004)
3. McManus, I.: A Multidisciplinary Approach to Highly Autonomous UAV Mission Planning and Piloting for Civilian Airspace. School of Engineering Systems. QUT (2004)
4. Kayton, M., Fried, W.: Avionics Navigation Systems. John Wiley & Sons, New York (1997)
5. LaValle, S.M.: Planning Algorithms. Cambridge University Press, New York (2006)
6. Wu, P., Clothier, R., Campbell, D., Walker, R.: Fuzzy Multi-Objective Mission Flight Planning in Unmanned Aerial Systems. IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making, Honolulu, Hawaii (2007)
7. Deitch, E.L.: Learning to Land: A Qualitative Examination of Pre-Flight and In-Flight Decision-Making Processes in Expert and Novice Aviators. Virginia Polytechnic Institute and State University, Falls Church, Virginia (Nov 21, 2001)
8. Cummings, M.L., Bruni, S.: Collaborative Human-Computer Decision Making in Network Centric Warfare. International Journal of Aviation Psychology (Accepted)
9. Adams, R.J.: How Expert Pilots Think: Cognitive Processes in Expert Decision Making. FAA, Washington D.C. (1993)
10. List, G.F., Mirchandani, P.B., Turnquist, M.A., Zografos, K.G.: Modeling and Analysis for Hazardous Materials Transportation - Risk Analysis, Routing Scheduling and Facility Location. Transportation Science **25** (1991) 100-114
11. Rathbun, D.: An Evolution based Path Planning Algorithm for Autonomous Motion of a UAV through Uncertain Environments. (2002)

12. Simpson, P.A.: Naturalistic Decision Making in Aviation Environments. Air Operations Division, Aeronautical and Maritime Research Laboratory (2001) 28
13. Coppin, G., Cadier, F., Lenca, P.: Some considerations of cognitive modeling for collective decision support. 40th Annual Hawaii International Conference on System Sciences, 2007. HICSS 2007. (2007) 10
14. Graham, H., Coppin, G., Cummings, M.L.: The PRIM: Extracting Expert Knowledge for Aiding in C2 Sense & Decision Making. 12th ICCRTS "Adapting C2 to the 21st Century". CCRP, Newport, Rhode Island (2007)
15. Cohen, M.S., Freeman, J.T.: Thinking Naturally About Uncertainty. 40th Annual Meeting of the Human and Ergonomics Society, Vol. 1, Philadelphia, PA (1996) 179-183
16. Rasmussen, J.: Skills, rules, and knowledge; Signals, signs, and symbols, and other distinctions in human performance models. IEEE Trans. on Systems, Man and Cybernetics **13** (1983) 257-266
17. Onken, R., Walsdorf, A.: Assistant systems for aircraft guidance: cognitive man-machine cooperation. Aerospace Science and Technology **5** (2001) 511-520
18. Gerd Gigerenzer, P.M.T.: Simple Heuristics that Make Us Smart. Oxford University Press, New York (1999)
19. Ferguson, D., Likhachev, M., Stentz, A.T.: A Guide to Heuristic-based Path Planning. Int. Conf. on Automated Planning and Scheduling (ICAPS), Monterey, CA (2005)
20. Likhachev, M.: Search-based Planning for Large Dynamic Environments. School of Computer Science. Carnegie Mellon University, Pittsburgh, PA (2005) 143
21. Clothier, R., Walker, R., Fulton, N., Campbell, D.: A Casualty Risk Analysis for Unmanned Aerial System (UAS) Operations over Inhabited Areas. Twelfth Australian International Aerospace Congress (AIAC12), Melbourne, Australia (2007)
22. Kacpryzk, J.: Mutistage Fuzzy Control. John Wiley and Sons, West Sussex, UK (1997)
23. Judd, K.B., McLain, T.W.: Spline based path planning for unmanned air vehicles AIAA Guidance, Navigation, and Control Conference and Exhibit,, Vol. AIAA-2001-4238 Montreal, Canada (2001)
24. Singh, L., Fuller, J.: Trajectory generation for a UAV in urban terrain, using nonlinear MPC. American Control Conference, 2001. Proceedings of the 2001, Vol. 3 (2001) 2301-2308 vol.2303
25. Frazzoli, E., Dahleh, M.A., Feron, E.: Maneuver-based motion planning for nonlinear systems with symmetries. Robotics, IEEE Transactions on [see also Robotics and Automation, IEEE Transactions on] **21** (2005) 1077-1091
26. Aerosim Blockset. Unmanned Dynamics (2003)
27. Schouwenaars, T., How, J., Feron, E.: Receding horizon path planning with implicit safety guarantees. American Control Conference, 2004. Proceedings of the 2004, Vol. 6 (2004) 5576-5581 vol.5576
28. Singh, L., Plump, J., McConley, M.W., Appleby, B.D.: Software Enabled Control: Autonomous Agile Guidance and Control for a UAV in Partially Unknown Urban Terrain. AIAA Guidance, Navigation, and Control Conference and Exhibit, Austin, Texas (2003)
29. Schouwenaars, T., Mettler, B., Feron, E., How, J.: Robust Motion Planning Using a Maneuver Automaton with Built-In Uncertainties. AIAA Aerospace Sciences and Exhibit, Reno Nevada (2003)
30. Singh, S.N.: Ultimate boundedness control of uncertain systems with application to roll coupled aircraft maneuver. Proceedings of the 28th IEEE Conference on Decision and Control (1989) 1708-1713 vol.1702

Multi-Objective UAS Flight Management in Time Constrained Low Altitude Local Environments

Pritesh Narayan[1], Duncan Campbell[2] and Rodney Walker[3]
Australian Research Centre for Aerospace Automation (ARCAA), Queensland University of Technology (QUT), Brisbane, Australia

This paper presents a new framework for Multi-Objective Flight Management of Unmanned Aerial Systems (UAS), operating in partially known environments, where planning time constraints are present. During UAS operations, civilian UAS may have multiple objectives to meet including: platform safety; minimizing fuel, time, distance; and minimizing deviation from the current path. The planning layers within the framework use multi-objective optimization to converge to a solution which better reflects overall mission requirements. The solution must be generated within the available decision window, else the UAS must enter a safety state; this potentially limits mission efficiency. Local or short range planning at low altitudes requires the classification of terrain and infrastructure in proximity as potential obstacles. The potential increase in the number of obstacles present further reduces the decision window in comparison to high altitude flight. A novel Flight Management System (FMS) has been incorporated within the framework to moderate the time available to the environment abstraction, path and trajectory planning layers for more efficient use of the available decision window. Enabling the FMS during simulation increased the optimality of the output trajectory on systems with sufficient computational power to run the algorithm in real time. Conversely, the FMS found sub-optimal solutions for the system with insufficient computational capability once the objective utility threshold was decreased from 0.95 to 0.85. This allowed the UAS to continue operations without having to resort to entering a safe state.

I. Introduction

In recent times, UAS have been employed in an increasingly diverse range of applications. Numerous UAS market forecasts portray a burgeoning future, including predictions of a USD10.6 billion market by 2013[1]. Within the civilian realm, UAS are expected to be useful in performing a wide range of airborne missions such as disaster monitoring, search and support, and atmospheric observation[2]. However, to realize these civilian applications, seamless operation of UAS within the NAS will be required; this is a difficult problem.

Most literature[3, 4] indicate that an equivalent level of safety (ELOS) to that of a human pilot will be one of the requirements for integration of UAS into the NAS. The ELOS requirement indicates that the system must be capable of replicating some of the capabilities of a human pilot; this leads to the need for a higher degree of onboard autonomy.

Automation assists in overcoming restrictions commonly found on current Remotely Piloted Aerial Vehicles (RPV). For example: Limited RPV range due to signal limitations; the need to stay within line of sight of remote pilot; decrease in pilot reaction; and pilot fatigue. A higher degree of onboard autonomy includes the ability to respond automatically to hardware failures and respond to changes in the environment through onboard replanning and execution. These tasks are routinely performed by human pilots; automating these tasks onboard results in a more robust UAS that is not as susceptible to onboard failures. Such autonomy could potentially lead to a decrease in operational costs.

[1] PhD Candidate, Engineering Systems, p.narayan@qut.edu.au
[2] Associate Professor, Engineering Systems, da.campbell@qut.edu.au
[3] Professor, Engineering Systems, ra.walker@qut.edu.au

Low altitude UAS operations present further challenges not encountered in high altitude flight. Terrain and urban structures may become hazards to the safety of the UAS, and must be treated as obstacles. The inclusion of terrain and urban structures as obstacles potentially increases the overall obstacle density within a given mission environment; conversely, the distance between obstacles is decreased. Thus, UAS operating at low altitudes may have less time available (shorter decision window) to generate and perform the appropriate manoeuvres for successful obstacle avoidance.

Traditionally, local path planning and trajectory generation techniques converge to near/optimal solutions by minimizing only one cost function (e.g. fuel, time, or distance). However, during each mission; civilian UAS may have multiple objectives to meet including and not limited to: safety of vehicle, the immediate environment and the public at all times; successful completion of the mission; minimizing fuel, time, and/or distance; and minimizing deviation from the current path. The use of multi-objective optimization allows the generation of a solution which better reflects the overall requirements of the mission. For example, multi-objective optimization may allow UAS operating partially known environments to perform collision avoidance whilst optimizing the solution to also meet other objectives, such as mission completion; thus potentially increasing mission efficiency. However, the solution must still be generated within this limited decision window; otherwise the platform must resort to entering a safe state.

UAS vehicles can be broadly categorized into two types, rotary and fixed wing. Rotary UAS traveling at low velocities have the capability to brake and hover if the planner does not converge to a solution within the available decision window, thus averting a potential collision. Fixed wing and Rotary UAS traveling at higher velocities can offer increased mission efficiency, but an alternative collision avoidance strategy must be available if a solution is not available within the decision window. The collision avoidance strategy can be in the form of predefined non-holonomic safety manoeuvres[5, 6]. A collision avoidance strategy implicitly guarantees vehicle safety, however mission efficiency decreases each time the planner cannot converge to a solution within the decision window. Decreasing the frequency of which safety manoeuvres are required during operations can potentially lead to an increase in mission efficiency.

This paper presents a new framework for Multi-Objective Flight Management of UAS operating in partially known environments whilst addressing replanning time constraints. An outline of UAS local path planning approaches in partially known environments and related work is given in section II. Section III presents an overview of the proposed framework, while simulation results in section IV show how the addition of an FMS can increase mission efficiency. Finally, conclusions are presented in section V.

II. Problem Formulation

A local path planning system is generally described as a system which generates a smooth trajectory for a UAS to follow through a set of mission level waypoints. At higher altitudes and typically remote operating locations; UAS are not constantly required to avoid static or dynamic obstacles. Therefore a trajectory generator may be all that is required to generate a smooth trajectory through mission level waypoints.

During low altitude local path planning however, the environment may present several challenges not encountered in high altitude flight. Terrain and urban structures become hazards to the safety of the UAS, and must be treated as obstacles. Due to the limited distances between objects, UAS have a limited decision window to generate and perform the appropriate manoeuvres for successful obstacle avoidance. Low altitude local path planning may require the additional inclusion of a local waypoint planner to generate a collision free path between mission level waypoints first.

If UAS possess the capability to safely navigate low altitude environments, additional civil applications can potentially include: traffic surveillance; response to emergency situations; assisting search and rescue efforts and aerial mapping.

A. Related work

This section provides an overview of relevant local path planning systems presented in literature.

1. Known Environments

Singh[7] presents a 2D local path planning algorithm, which generates an optimal trajectory through a predefined set of waypoints in an environment known *a priori* using Model Predictive Control (MPC) techniques. This algorithm performs the planning component off-line, thereby limiting UAS operations to purely static environments.

Schouwenaars[8, 9] presents a 2D MPC based local path planning algorithm which takes into account a static 2D environment known *a priori*. The solution is optimized using Mixed Integer Linear Programming (MILP). The safe state component of the algorithm ensures vehicle safety is preserved if solution is not generated within a specified deadline.

Other research into planning in known environments has been presented by Rathbun[10] (genetic algorithms) and Pettersson *et al.* from the Wallenberg Information Technology and Autonomous Systems (WITAS)[11] (probabilistic planning).

Navigation in known environments implies the use of high resolution maps. This may not be feasible for some forms of UAS (e.g. mini or micro variants) due to: cost; computational; or payload limitations. An alternative is to use active or passive onboard sensors to perform online mapping; this is generally referred to as planning in partially known environments.

2. Partially known Environments

Sebastian *et al.*[12] present a local planning system which constructs a partially known 3D environment online using LAser Detection And Ranging (LADAR) information. A Laplacian (a type of potential field implementation), drives the UAS towards the goal until an obstacle is detected by onboard sensors. A reactive collision avoidance system, entitled the dodger is activated once an obstacle is detected. The obstacle avoidance manoeuvre is limited to either moving around, or over an obstacle.

Griffiths *et al.*[13] present another local planning system which generates an approximate 3D representation of the environment using low resolution map data. An initial path is constructed using a rapidly exploring random tree (RRT) algorithm. Similarly to Sebastian[12], if the UAS encounters an obstacle which has not been planned for, an obstacle avoidance algorithm (using static LADAR sensing data) is activated to perform collision avoidance.

Other research into planning in partially known environments has been presented by Shi[14, 15] (MILP optimization of LADAR sensing data) and Nikolos[16] (Evolutionary optimization of simulated sensing data)

Planners onboard UAS operating in partially known environments generally overcome the possibility of becoming trapped in local minima (it is still possible though), by planning in 3D. However, if a separate collision avoidance algorithm is activated when an unforeseen obstacle is detected; the safety is of UAS usually becomes the only priority. This can potentially lead to sub-optimal results since the optimal path to the goal may not be considered during the obstacle avoidance scenario. Additionally, the capability to consider multiple objectives could potentially benefit UAS operations in this scenario.

Manoeuvre Generation; developed by Frazzoli[17, 18] refers to the generation of a smooth trajectory over a set of waypoints through concatenation of predefined trim and manoeuvre primitives. Various UAS flight modes including: cruise; coordinated turn; climb or descend; and fixed wing safety manoeuvres (e.g. loiter) can be represented through trim and manoeuvre primitives.

Richards[19] presents a local path planning system which applies Frazzoli's[20] manoeuvre generation technique to low altitude 3D collision avoidance scenarios. A modified A* algorithm is used to generate a set of waypoints. Sub-optimal trajectories are generated using manoeuvre automaton which explicitly takes UAS flight envelope and non-linear motion constraints into account. Similarly, Singh[21] and Schouwenaars[8] have applied manoeuvre generation to the local path planning problem.

B. Unresolved Local Path Planning Considerations

There are two research challenges which have not been explicitly considered in the local path planning systems presented: optimization with respect to multiple objectives; and the more efficient use of the available decision window to generate an optimal solution.

Local planning systems presented in this section optimize a solution by minimizing only one cost function (e.g. fuel, time, or distance). However, during each mission; civilian UAS may have multiple objectives to meet. Multi-objective optimization allows convergence to a solution which takes numerous aspects of the mission into account. Additionally, each cost function to be met can be given a weighting to provide an indication of the importance placed on each objective. For example, during operations in collision free environments, greater weight can be placed on fuel, time and distance objectives, whereas operations in environments with obstacles present may require greater weighting to be placed on safety cost functions.

UAS operating without mapping sensors are restricted to operations strictly within known regions available through onboard maps. Mapping sensors allows operations outside known regions and may decrease overall payload requirements since onboard maps are optional. However, planning in partially known environments requires processing of sensor information and potentially; fusion of sensor data with onboard maps if available. The computational complexity of this process is not explicitly taken into account by any of the local path planning systems presented. It is generally implied that sufficient processing power is available that this process occurs instantaneously. With limited onboard computational resources; environment abstraction will take a finite length of time; thus decreasing the overall time available for the path planning and trajectory generation algorithms to converge to a solution within the available decision window.

Environment abstraction, path planning and trajectory generation layers each require a "slice" of the available decision window assuming that sufficient computational power is available to converge to a solution within the planning time available. To the author's knowledge, if the available computational power is insufficient, no research in literature explicitly attempts to moderate the time available to each layer to generate a partial solution. If the flight management can provide a partial or sub-optimal solution within the decision window, this allows the UAS to continue operations without having to resort to entering a safe state.

This concludes the overview of related work in the field of low altitude local path planning. The proposed solution presented in the next section incorporates multi-objective optimization into the local path planning process. Additionally the proposed solution identifies the computational complexity of environment abstraction and planning and attempts to generate a partial solution if there is insufficient time for the planning algorithms to converge to an optimal solution.

III. Proposed UAS Framework

In general, the local path planning process can be described as an iterative procedure (Figure 1), where current sensor data is fused with onboard mapping information (if available) to form an abstraction of the environment. The environment abstraction is used by an intermediate path planner to generate a set of collision free waypoints between two mission level waypoints. Finally, a smooth trajectory is generated through the intermediate waypoint set by a trajectory generation algorithm.

This entire process must be completed to ensure that the local path planning system converges to a solution within the finite decision window (Figure 1). In situations where the planner cannot converge to a solution within the time available; a partial solution should be available so the UAS can continue with the mission without having to resort to entering a safe state[5]. Potential benefits from more efficient use of the available decision window are increased mission efficiency and reduced operational costs.

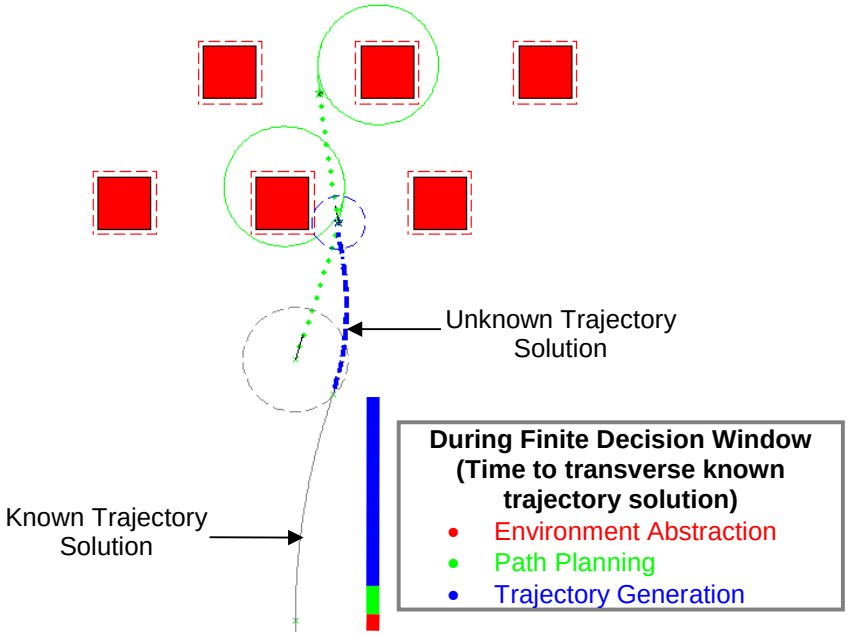


Figure 1. Finite Decision Window during Local Path Planning

A. Proposed Architecture

The architecture[22] (Figure 2) presented in this section is suited to UAS operations in partially known environments, and potentially offers greater mission efficiency and mission completion opportunities in comparison to the current approaches presented in Section II, Subsection A.

The inclusion of an FMS can provide greater mission efficiency through more resourceful use of the available decision window. The FMS dynamically allocates a finite "slice" to the environment abstraction, path and trajectory generation layers, with the length of time dependent on available onboard computational resources and overall decision window length.

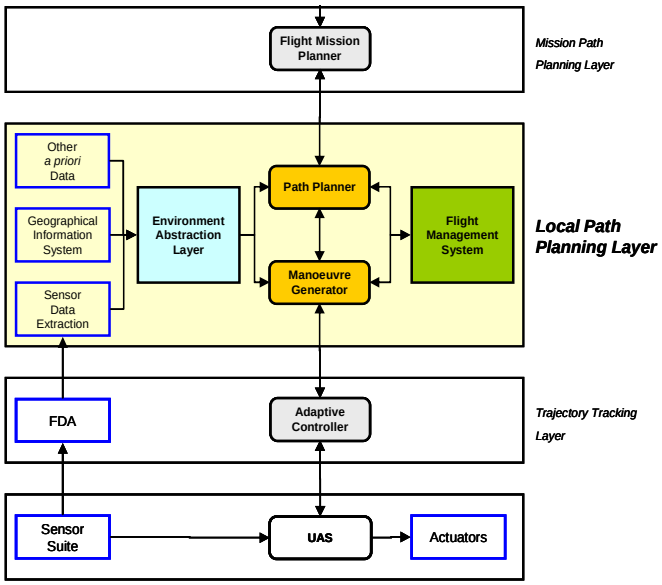


Figure 2. Proposed Architecture for Local Path Planning Concept Presented

1. Flight management layer

The FMS is an expert system which manages and schedules the execution parameters of the environment abstraction, path planning and trajectory generation layers. In scenarios with approaching real time deadlines, there is a limited amount of time available to the UAS to converge to a new feasible solution before a safety manoeuvre must be executed ensuring the safety of the vehicle.

The environment abstraction layer requires a finite length of time to generate a representation of the environment. The time remaining is then allocated to the path planning and manoeuvre generation layers. In a worst case scenario, the path planning algorithm must be terminated while enough time remains to generate a manoeuvre between two waypoints. To ensure that the FMS can moderate the length of time allocated to each layer, certain limitations must be emplaced on the: environment abstraction, path planning and trajectory generation layers. These limitations are discussed in the following sections.

2. Environment abstraction layer

The environment abstraction layer uses available sensor; map and other onboard data to create a representation of the immediate environment. Environment abstraction must be performed first since trajectory and path planning layers must have knowledge of possible hazards within proximity before a suitable navigation strategy can be devised. Additionally, if the environment abstraction layer does not output a situational representation within the time allocated by the FMS, its operations are deferred so the planning layers can attempt to generate a feasible solution within the time remaining.

3. Trajectory generation layer

The trajectory generation layer creates a feasible trajectory through a set of mission level waypoints whilst meeting dynamic and kinematic constraints of the UAS platform. This is sufficient for operations in obstacle free environments however, in the presence of obstacles; the path planning layer must be initialized to generate a set of intermediate waypoints representing a collision free path between mission level waypoints. Additionally, the trajectory generation algorithm should possess the capability to output a solution which is either partial, sub-optimal or both.

4. Path planning layer

During operations with obstacles in proximity, the path planning layer is initialized to generate a set of waypoints which represent a safe feasible path from the current position to the next mission level waypoint. The planner must take platform kinematic and dynamic constraints into account to ensure that waypoints generated within the platform performance envelope.

For the path planning algorithm to output a solution within a predefined set of time, it is desirable for it to display anytime qualities, where either a partial and/or sub-optimal solution can be output whenever required. Additionally, the path planner can operate in parallel to the environment abstraction trajectory generation layers, but must generate the forthcoming intermediate waypoint before the trajectory generator is initialized. The trajectory generator requires this information to calculate the exit attitude of the UAS when generating a trajectory between two waypoints.

This concludes the overview of the proposed planning framework for UAS operations in partially known environments. The following section provides an implementation overview and subsequent results to demonstrate the feasibility of the framework presented in III.

IV. Demonstration of Framework Feasibility

The framework was implemented using MATLAB to demonstrate its potential to improve overall mission efficiency during operations in partially known environments. The following section provides an overview of the implementation regarding the: FMS; environment abstraction; path planning and trajectory generation layers.

A. Framework Implementation Details

A 2D environment representation was setup to simulate an urban scenario (Figure 3) where the UAS assignment included safe and efficient navigation through a set of predefined mission level waypoints. The finite decision window is calculated as the time taken to complete the current stage, where each waypoint pair is regarded as a single stage. Additionally a fixed wing platform is used during simulation due to their incapacity to brake and hover. If no solution is available once the decision window comes to an end, the UAS permanently enters a safe state using a loiter manoeuvre.

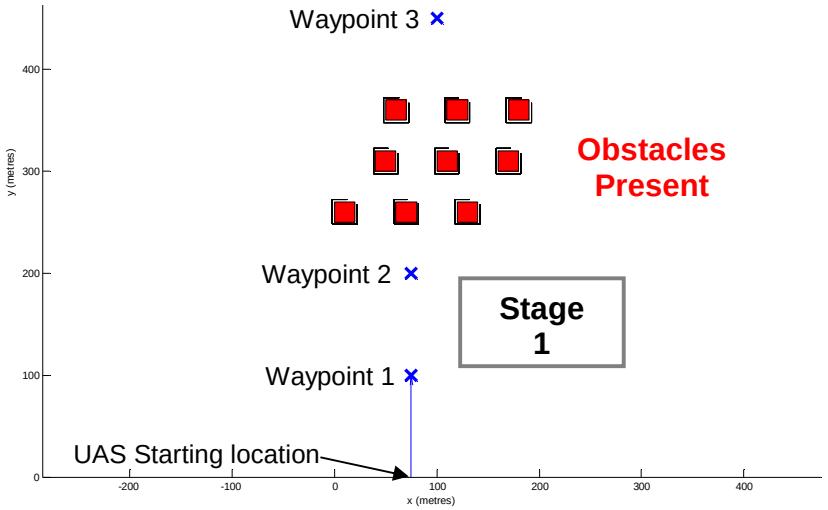


Figure 3. Environment Setup for Simulation

The path planning algorithm (Figure 4) implemented is based on Smith's fuzzy logic path planning algorithm[23]. The iterative nature of Smith's algorithm makes it quite suitable for local path planning as a partial solution is available if the algorithm is terminated by the FMS before completion; it also performs planning with respect to multiple mission objectives.

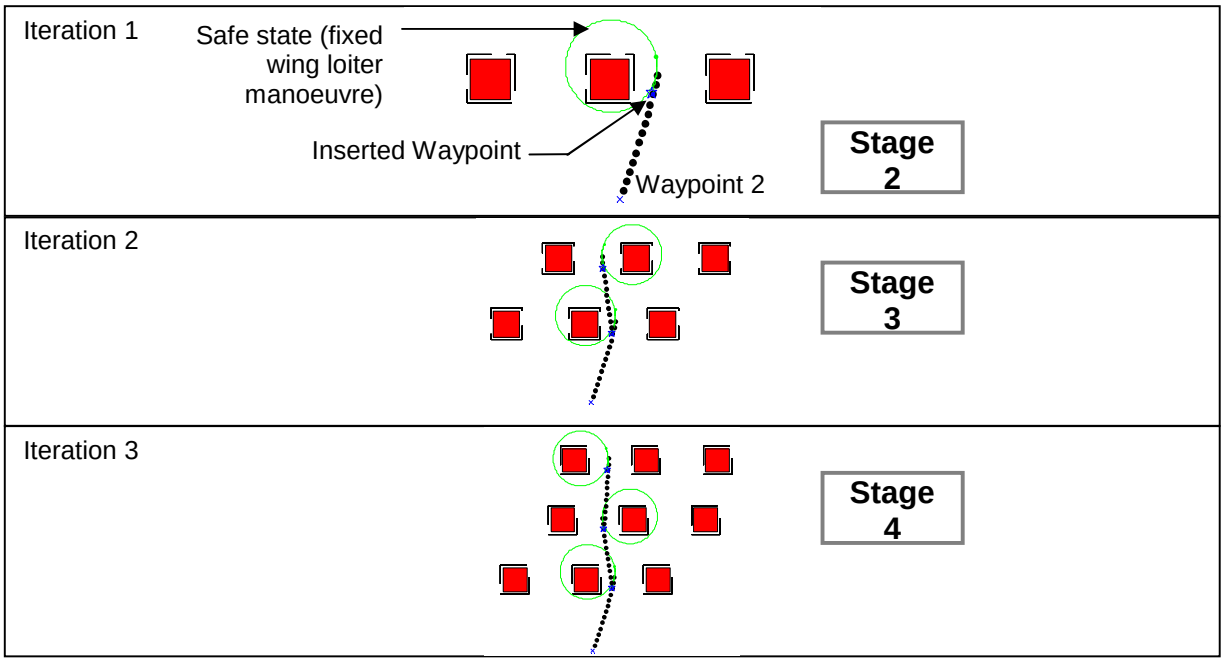


Figure 4. Path Planning Algorithm

The trajectory generation algorithm (Figure 5) is based on Frazzoli's[20] manoeuvre generation framework. Frazzoli[20] states that pre-defined discrete classes of manoeuvres can be concatenated together to create smooth trajectories through a set of waypoints. Representing aircraft motion as classes of manoeuvres has previously been demonstrated in[17, 18, 21, 24]. Frazzoli's manoeuvre generation research and subsequent work has been limited to rotary UAS only; thus new trim and transition manoeuvre sets for fixed wing UAS operating in cruise and coordinated turn flight modes were created for simulation.

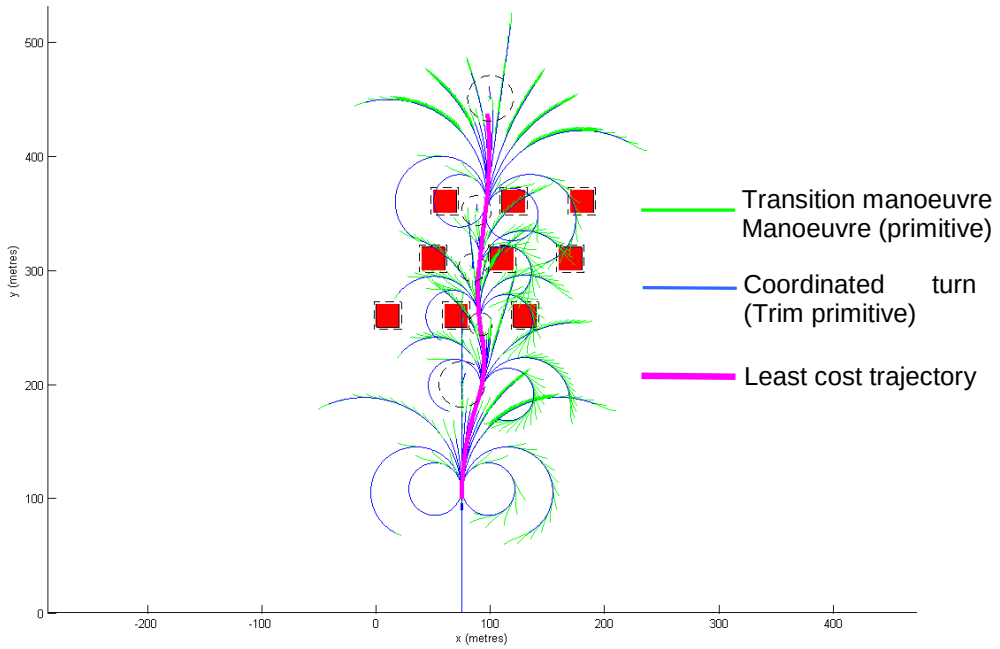


Figure 5. Trajectory Generation Algorithm (six predefined coordinated turn trim primitives)

Multi-objective optimization is applied to both path planning and trajectory generation algorithms. Fuzzy multi-objective optimization is already a component of Smith's path planning algorithm[23]; however a simpler aggregation of the utility of multiple objectives (a utility of one denotes a cost of zero) has been applied to the trajectory solution for computational efficiency.

The FMS initializes the environment abstraction layer to generate a representation of the environment within the current stage. If obstacles are detected, the path planner is initialized to generate a set of waypoints within the stage to reach the next mission level waypoint. The time remaining is allocated to trajectory generation layer which iteratively finds a more optimal solution until there is insufficient time left. If excess time remains after the trajectory generator outputs a solution, this time is allocated the environment abstraction; path planning and trajectory generation layers to generate solutions for future stages. Conversely, if no solution is available then the UAS resorts to entering a safe state indefinitely.

B. Simulation Setup

The Aerosonde UAS has been used as the vehicle platform for the simulation results presented in the following section. During the simulation the platform operates at a constant velocity of 15 m/s in either cruise or coordinated turn flight modes. The maximum roll angle is set to 45 degrees; this has been verified using the 6 degree of freedom Aerosonde UAS model available with the Aerosim Blockset for MATLAB. The objectives chosen for simulation include: distance minimization, meeting yaw angle requirement (generated by path planner) at goal location and distance of candidate solution from the goal location; all objectives have equal weighting.

The simulation has been performed using three computers with varying processing capabilities (Table 1) to simulate the how an FMS can potentially increase mission efficiency of the same UAS with different computational capabilities.

Computer	Processor	Memory (RAM)	MATLAB Version	Operating System
A	Core 2 Duo @ 3.2 GHz	2 GB	7	Windows XP
B	Core 2 Duo @ 2.13 GHz	2 GB	7	Windows XP
C	Centrino Duo @ 2 GHz	2GB	7	Windows Vista

Table 1. Available Computing Power of Candidate Computers

C. Simulated Results – Section 1: FMS Not Enabled

The first set of results show the algorithms performance without the FMS enabled for each computing setup. The algorithm finds a feasible path using a combination of: cruise; six coordinated turn trims; and the resulting transitions between candidate trim manoeuvres.

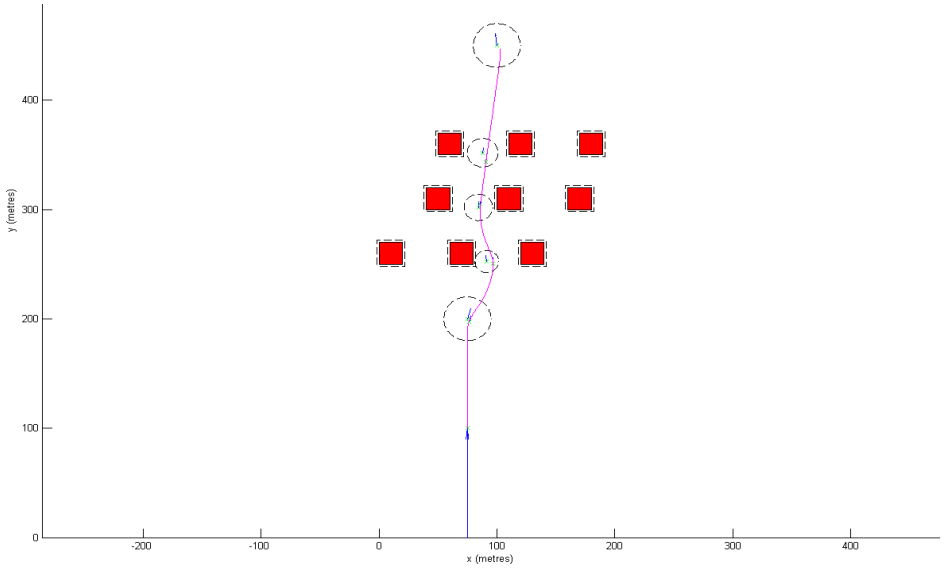


Figure 6. Least Cost Trajectory Solution generated using Cruise and six Coordinate Turn Trims

The mean results for a Monte Carlo setup (100 algorithm iterations) are presented (Table 2). The total simulation run time is given in conjunction with average run times for each layer. The decision window represents the time available all layers for planning (Figure 7). If the decision window remaining at the end of the simulation is positive, then a potentially more optimal trajectory could have been generated through more efficient use of the decision window. Conversely if the remaining decision window is negative, insufficient time (or processing power) was available to generate the solution in real time. However, it may still be possible to find a less optimal path within the given decision window.

Computer	Algorithm Run Time [mean (std dev)] (seconds)				Decision Window Remaining	Utility Threshold (Upper Bound)
	Env Abstraction Layer	Path Planning Layer	Trajectory Generation Layer	Total Run Time		
A	0.706 (0.234)	1.345 (0.048)	7.899 (0.102)	9.95 (0.261)	15.136 (0.227)	0.95
B	1.318 (0.346)	2.013 (0.053)	12.066 (0.098)	15.396 (0.329)	9.790 (0.274)	0.95
C	4.893 (0.797)	3.624 (0.883)	25.356 (2.768)	33.872 (3.785)	-8.095 (3.719)	0.95

Table 2. Algorithm Run Time for Cruise and six Coordinated Turn Trim Manoeuvres (FMS not enabled)

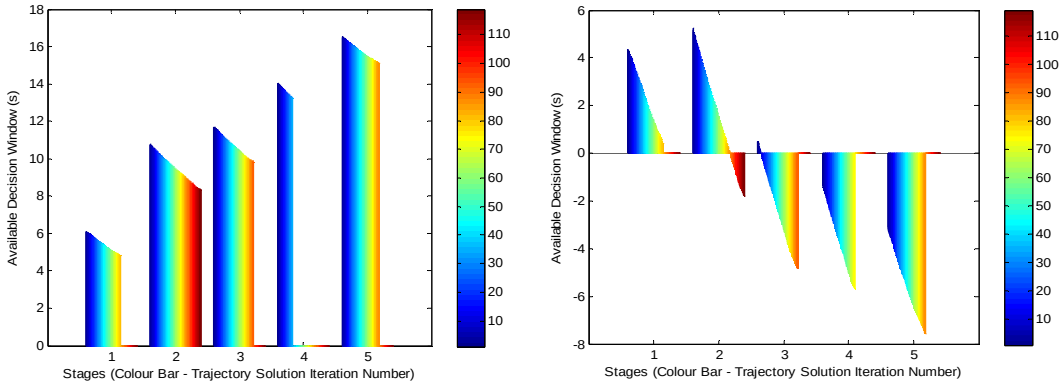


Figure 7. Available Decision Window during Simulation (Computer A - left) (Computer C – Right)

Table 3 presents the optimality of the solution found in terms of its utility rather than cost. Where an optimal solution in terms of least cost solution approaches zero, the normalized utility of the solution approaches one. To generate solutions more efficiently, an upper bound of 0.95 has been set. Once a feasible solution is found which exceeds this value, the trajectory generator stops looking for other possible solutions for the current stage and moves to the next stage. This prevents the trajectory generator continuously searching for other solutions when an acceptable solution has been previously discovered. This can be seen in Figure 7 where the trajectory generator finds a solution with a utility above 0.95 relatively fast for stage 4 and immediately proceeds to find a feasible trajectory solution for stage 5.

Computer	Utility Value of Output Trajectory (max value = 1)						Utility Threshold (Upper Bound)
	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5	Average	
A, B and C	0.872	0.863	0.933	0.977	0.891	0.907	0.95

Table 3. Utility Value of Output Trajectory (FMS not enabled)

D. Simulated Results – Section 2: FMS Enabled

The second set of results present an overview of the algorithms performance once the FMS has been enabled.

Computers A and B have sufficient processing power to generate a solution for the given scenario in real time. Since the time required finding a feasible solution was less than the UAS flight time, this resulted in a positive decision window remaining at the end of the non FMS enabled simulation (Table 2). Enabling the FMS results in more efficient use of the decision window (Figure 10) and subsequently, a more optimal solution is found (Figure 8) (Table 4).

Computer C has insufficient processing power available to generate a solution in real time; thus the remaining decision window is negative (Table 2). The FMS attempts to find a sub-optimal solution, however no feasible solution can be found within the given decision window of stage 2 (Table 4) (Figure 11); the UAS must then resort to entering a safe state (Figure 9). A feasible solution was discovered by the FMS by decreasing the utility threshold to 0.85 (Table 5). The resulting solution was less optimal in comparison, but allowed the UAS to continue operations without having to initiate a safe state manoeuvre (Figure 10).

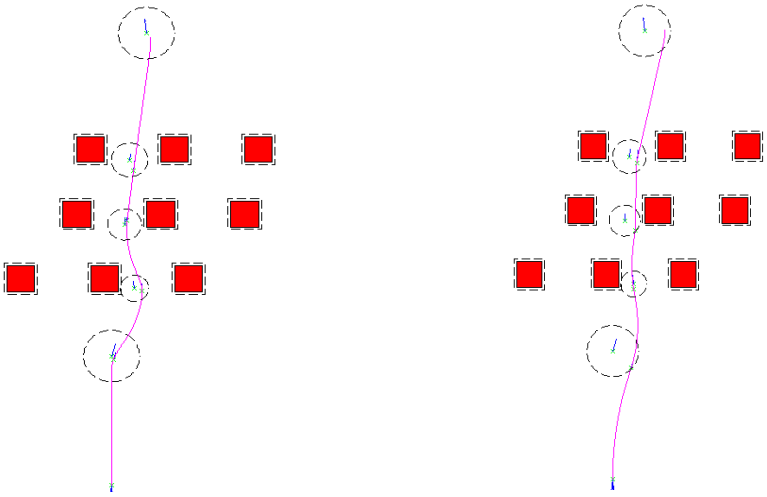


Figure 8. Trajectory Solution Generated (No FMS - left) (Computer B with FMS enabled – right)

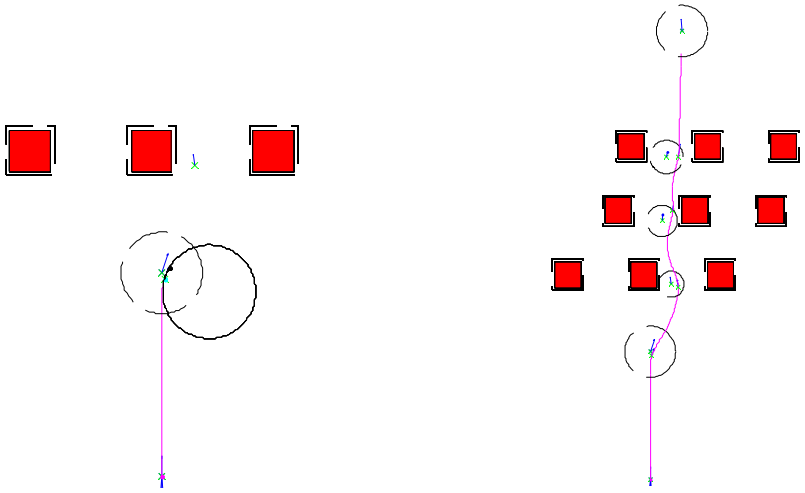


Figure 9. Trajectory Solution Generated (FMS Enabled) (Computer C: utility threshold 0.95 - left) (Computer C: utility threshold 0.85 - right)

Computer	Algorithm Run Time [mean (std dev)] (seconds)				Decision Window Remaining	Utility Threshold (Upper Bound)
	Env Abstraction Layer	Path Planning Layer	Trajectory Generation Layer	Total Run Time		
A	1.194 (0.349)	1.531 (0.05)	23.236 (0.2889)	25.961 (0.067)	0.344 (0.017)	0.95
B	1.394 (0.353)	2.107 (0.067)	22.322 (0.306)	25.823 (0.07)	0.071 (0.011)	0.95
C	2.105 (0.46)	2.211 (0.121)	9.452 (0.635)	13.77 (0.67)	0.033 (0.046)	0.95
C	4.114 (1.0)	3.276 (0.264)	16.071 (0.742)	23.462(1.191)	1.897 (1.008)	0.85

Table 4. Algorithm Run Time for Cruise and Coordinated Turn Trim Manoeuvres (FMS Enabled)

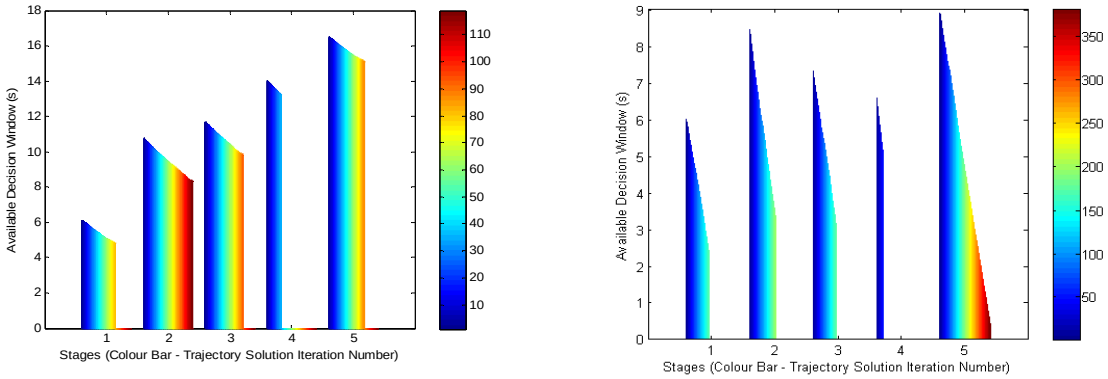


Figure 10. Available Decision Window during Simulation (Computer A with FMS Disabled - left) (Computer A with FMS enabled – Right)

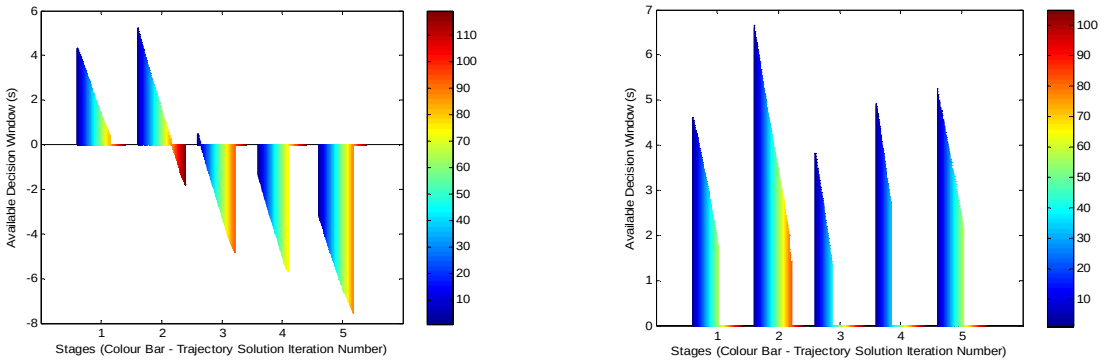


Figure 11. Available Decision Window during Simulation (Computer C: FMS not enabled - left) (Computer C: utility threshold 0.85 - right)

Table 5 presents an overview of the utility value of the output trajectory after the FMS has been enabled. It can be seen that candidate systems possessing sufficient processing power to compute a solution in real time benefit from an increase in the average utility value of the output trajectory once the FMS is enabled. Additionally computer C is able to find a feasible solution in real time once the utility threshold is reduced to 0.85. The utility threshold is currently set manually. Implementing a variable utility threshold has the potential to further increase the effectiveness of the FMS.

Computer	Utility Value of Output Trajectory (max value = 1)						Utility Threshold (Upper Bound)
	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5	Average	
A	0.972	0.958	0.952	0.965	0.932	0.956	0.95
B	0.972	0.958	0.936	0.924	0.867	0.931	0.95
C	0.8715	0	0	0	0	0	0.95
C	0.8715	0.8625	0.8845	0.9063	0.8759	0.8801	0.85

Table 5. Utility value of Output Trajectory (FMS Enabled)

V. Conclusions

This paper has presented a new framework for multi-objective flight management in time constrained low altitude local environments. A finite length of time defined as the limited decision window was dynamically distributed among the: environment abstraction; path planning; and manoeuvre generation layers by the FMS. In a particular scenario where the UAS does not possess sufficient processing capabilities to generate a full solution within the time available, a partial and/or sub-optimal solution was found in several scenarios. This allows the UAS to continue the mission without having to resort to entering a safe state; thus potentially increasing mission efficiency.

It is expected that in future, the overall capabilities of the framework implementation will be extended in several areas. 3D planning and trajectory generation can be employed through the implementation of additional flight modes, for example climb and descend. Additionally, the implementation of a variable utility threshold may increase the effectiveness of the FMS further.

Acknowledgments

The author would like to thank and acknowledge the support of ARCAA and QUT throughout this research project.

References

[1]Cox, T., "Civil UAV Capability Assessment (Draft Version)," 2004

[2]Wegener, S., "UAV Autonomous Operations for airborne Science Missions," 2004

[3]Civil Aviation Authority of Australia, "Advisory Circular - UAV Operations, Design Specification, Maintenance and Training of Human Resources," 2002

[4]EUROCONTROL, "Specifications for the use Of Military Unmanned Aerial Vehicles as Operational Air Traffic Outside Segregated Airspace," 2006

[5]Schouwenaars, T., How, J. and Feron, E., "Receding horizon path planning with implicit safety guarantees," *American Control Conference Proceedings*, 2004

[6]Anisi, D., Robinson, J. and Ögren, P., "On-Line Trajectory Planning for Aerial Vehicles: A Safe Approach with Guaranteed Task Completion," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, Colorado, 2006

[7]Singh, L. and Fuller, J., "Trajectory generation for a UAV in urban terrain, using nonlinear MPC," *American Control Conference Proceedings*, 2001

[8]Schouwenaars, T., Mettle, B., Feron, E. and How, J., "Hybrid Model for Trajectory Planning of Agile Autonomous Vehicles," *Journal of Aerospace Computing, Information, and Communication*, 1, 2004

[9]Schouwenaars, T., Valenti, M., Feron, E. and How, J., "Implementation and Flight Test Results of MILP-based UAV Guidance," *IEEE Aerospace Conference*, 2005

[11]Doherty, P., Granlund, G., Kuchcinski, K., Sandewall, E., Nordberg, K., Skarman, E. and Wiklund, J., "The WITAS Unmanned Aerial Vehicle Project," *14th European Conference on Artificial Intelligence*, Amsterdam, 2000

[12]Sebastian, S., Singh, S., Chamberlain, L. and Saripalli, S., "Flying Fast and Low Among Obstacles," *IEEE International Conference on Robotics and Automation*, 2007

[13]Muscettola, N., Nayak, P. P., Pell, B. and Williams, B. C., "Remote Agent: to boldly go where no AI system has gone before," *Artificial Intelligence*, 103, 1-2, 1998

[14]Shi, D., "Aerial Robot Navigation in Cluttered Urban Environments," College of Engineering, Florida, 2006

[15]Shi, D., Selekwa, M. F., Collins, E. G., Jr. and Moore, C. A., "Fuzzy behavior navigation for an unmanned helicopter in unknown environments," *IEEE International Conference on Systems, Man and Cybernetics*, 2005

[16]Nikolos, I. K., Valavanis, K. P., Tsourveloudis, N. C. and Kostaras, A. N., "Evolutionary algorithm based offline/online path planner for UAV navigation," *Ieee Transactions on Systems Man and Cybernetics Part B-Cybernetics*, 33, 6, 2003

[17]Frazzoli, E., Dahleh, M. A. and Feron, E., "Real-time motion planning for agile autonomous vehicles," *American Control Conference Proceedings*, 2001

[18]Frazzoli, E., Dahleh, M. A. and Feron, E., "Maneuver-based motion planning for nonlinear systems with symmetries," *IEEE Transactions on Robotics*, 21, 6, 2005

[19]Richards, N. D., Sharma, M. and Ward, D. G., "A hybrid A*/automaton approach to on-line path planning with obstacle avoidance," *Collection of Technical Papers - AIAA 1st Intelligent Systems Technical Conference; Collection of Technical Papers - AIAA 1st Intelligent Systems Technical Conference*, 1, 2004

[20]Frazzoli, E., Dahleh, M. A. and Feron, E., "Maneuver-based motion planning for nonlinear systems with symmetries," *Robotics, IEEE Transactions on [see also Robotics and Automation, IEEE Transactions on]*, 21, 6, 2005

[21]Singh, L., Plump, J., McConley, M. and Appleby, B., "Software Enabled Contro: Autonomous Agile Guiance and Control for a UAV in Partially Unknown Urban Terrain," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Austin, Texas, 2003

[22]Narayan, P., Wu, P., Campbell, D. and Walker, R., "An Intelligent Control Architecture for Unmanned Aerial Systems (UAS) in the National Airspace System (NAS)," *Australian International Aerospace Conference (AIAC)*, Melbourne, 2007

[23]Smith, E. B. and Langari, R., "Fuzzy Multiobjective Decision Making for Navigation of Mobile Robots in Dynamic Unstructured Environments," *Journal of Intelligent and Fuzzy Systems*, 14, 2003
[24]Richards, N. D., Sharma, M. and Ward, D. G., "A hybrid A*/automaton approach to on-line path planning with obstacle avoidance," *Collection of Technical Papers - AIAA 1st Intelligent Systems Technical Conference*, 1, 2004

AN INTELLIGENT CONTROL ARCHITECTURE FOR UNMANNED AERIAL SYSTEMS (UAS) IN THE NATIONAL AIRSPACE SYSTEM (NAS)

Pritesh Narayan*, Paul Wu, Duncan Campbell, Rodney Walker

** p.narayan@qut.edu.au, PhD research student, Australian Research Centre for Aerospace Automation (ARCAA), Queensland University of Technology (QUT), Brisbane, Australia*

Abstract

In recent times, Unmanned Aerial Systems (UAS) have been employed in an increasingly diverse range of applications. Numerous UAS market forecasts portray a burgeoning future, with many applications in both the military and civilian domains. Within the civilian realm, UAS are expected to be useful in performing a wide range of missions such as disaster monitoring (e.g. wildfires, earth-quakes, tsunamis and cyclones), search and support, and atmospheric observation.

However, to realise these civilian applications, seamless operation of UAS within the National Air Space (NAS) will be required. Increasing the levels of onboard autonomy will help to address this requirement. Additionally, increased autonomy also reduces the impact of onboard failures, potentially lower operational costs, and decrease operator workload.

Numerous intelligent control architectures do exist in the literature for mobile robots, space based robots and for UAS. These include: the WITAS project, Open Control Platform, Remote Agent and TRAC/ReACT. However, none of these are specifically targeted at providing the required support for a wide range of civilian UAS missions. Operation of UAS in the NAS for civil applications require robust methods for dealing with emergency scenarios such as performing forced landings and collision avoidance to preserve the safety of people and property.

This paper presents a new multi layered intelligent control architecture. The highest layer provides deliberative reasoning and includes situational awareness and mission planning subsystems. The middle layers deals with navigational aspects (such as path planning and manoeuvre generation). Finally, there is a functional control layer which comprises sensor and actuator subsystems and provides reactive functionality to enable forced landings and collision avoidance. Collision avoidance and forced landing technologies are currently under development at the Australian Research Centre for Aerospace Automation (ARCAA).

Biography

Pritesh Narayan received a Bachelors degree in Aerospace Avionics Engineering with first class honours from QUT in 2005. He is currently completing a PhD in UAV navigation with ARCAA at QUT. The PhD research is focused on reactive UAS flight management in uncertain urban environments.

Paul Wu completed both his Bachelors degree in Electrical and Computer Engineering and his Masters degree in Computer and Communications Engineering from QUT in 2005. He is currently working towards a PhD degree with ARCAA at QUT. His research interests include multi-objective path planning, fuzzy logic and embedded systems.

A/Prof Duncan Campbell is with the School of Engineering Systems at QUT and a member of ARCAA. He has research expertise in real-time computational intelligence, intelligent control, embedded systems and decision support systems. A/Prof Campbell has commercial research linkages in agricultural, food and beverage, and information and communication technologies (ICT) industries. He has a career total of 60 internationally refereed scholarly papers and book chapters, and international patents on an intelligent control application for beverage production.

A/Prof Rod Walker is Director of ARCAA, which is a $12M facility supported by QUT, CSIRO and the Queensland State Government. Rod has a diverse aerospace research background that has involved electromagnetic propagation modelling, GPS multipath modelling, aerospace image processing, aerospace risk assessment and fault tolerant control. He has consulted widely on aerospace automation issues with clients such as: the mining industry, Airservices Australia, Boeing Phantom Works, DSTO and the Federal Government. He is the author of over 70 scientific papers and book chapters and has appeared in over 20 media articles.

Introduction

In recent times, UAS have been employed in an increasingly diverse range of applications. Numerous UAS market forecasts portray a burgeoning future, including predictions of a USD10.6 billion market by 2013 [1]. Within the civilian realm, UAS are expected to be useful in performing a wide range of airborne missions such as disaster monitoring, search and support, and atmospheric observation [2].

However, to realise these civilian applications, seamless operation of UAS within the NAS will be required; this is a difficult problem. Most literature [3, 4] indicate that an equivalent level of safety (ELOS) to that of a human pilot will be one of the requirements for integration of UAS into the NAS. The ELOS requirement, indicates that the system must be capable of replicating some of the capabilities of a human pilot; this leads to the need for a higher degree of onboard autonomy.

A higher degree of onboard autonomy includes the ability to respond automatically to hardware failures and respond to changes in the environment through onboard replanning and execution. These tasks are routinely performed by human pilots; automating these tasks results in a more robust UAS that is not as susceptible to onboard failures. Furthermore, it reduces the human operator's workload and therefore potentially allows a single human to operate multiple unmanned aircraft instead of many human operators controlling one aircraft. As well, it allows the operator to focus on the mission rather than piloting aspects. Such autonomy could potentially lead to a decrease in operational costs.

However, taking the human pilot out of an aircraft removes much sensory and decision making capability. To demonstrate that a UAS still has an equivalent level of safety to a human piloted aircraft, this capability must be automated. For this to occur, UAS will need to possess greater "intelligence" than they do today, aspiring to acquire the traits of the human pilot. The UAS will need to acquire the capacity to monitor the vehicle's internal systems and the outside world, and to detect any changes that affect the mission safety and mission outcome. With this information, the UAS must then make rational decisions and take the necessary actions to preserve safety and achieve mission objectives. This capability can be implemented through the use of an intelligent control architecture.

Defining Intelligent Control

Intelligent control is a multi disciplinary field (***Figure 1***) that involves the use of techniques from the fields of Artificial Intelligence and Control within the context of the Operational Requirements of the task. Intelligent control systems are generally structured in a hierarchical manner. High level (Complex and abstract) tasks are decomposed into a series of time critical low level tasks (data rich and precise). This obeys the so called "principle of increasing precision with decreasing intelligence" [5].

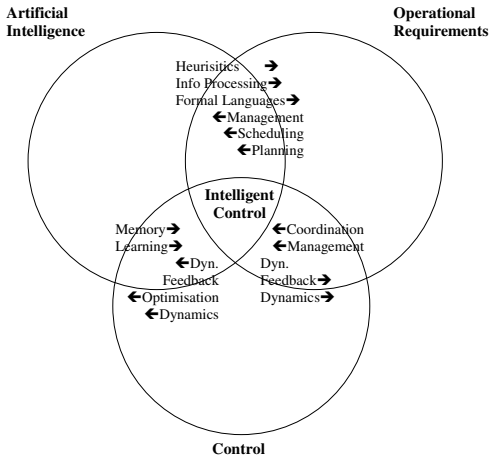


Figure 1 - Definition of Intelligent Control Discipline [6]

Intelligent Control and the Human Pilot

Replicating the capabilities of a human pilot is not a trivial task. For example, during a routine manned flight in civilian airspace, the pilot uses available data (e.g. terrain maps), sensor readings and instructions from air traffic management (ATM) to fly the aircraft safely to its destination. The pilot is capable of dealing with varying situations including and not limited to: turbulence, onboard failures (e.g. actuator, sensor, engine), performing a forced landing and avoiding potential collisions with terrain and other aircraft.

To encapsulate the qualities of a human pilot within UAS, the intelligent control architecture must accurately model a pilot's decision making process. An example of aircraft pilots cognitive process [7] during routine flight is shown in ***Figure 2***. The cognitive model is relatively complex but the reader should note that human pilots have their own sensors (e.g. vision, touch) and actuators (e.g. hands, feet). Pilots use their own perception (e.g. recognition of obstacles) in conjunction with memory (prior experiences) to take appropriate actions in a broad range of scenarios.

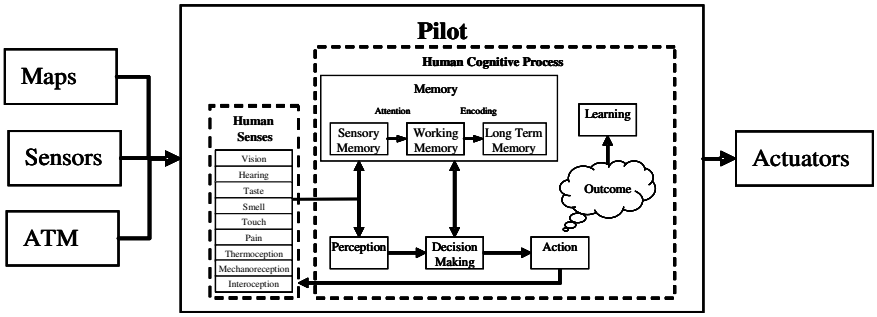


Figure 2 - Cognitive Model of an Aircraft Pilot's Decision Making Process during flight

The purpose of this paper is to combine the principles of the human cognitive process and the field of intelligent control to encapsulate the qualities of a human pilot. In order to achieve this, a review of existing architectures is presented in the next section.

Review of Intelligent Control Architectures

An overview of existing architectures in robotics, spacecraft and UAS is presented in this section to identify relevant architecture design methodologies and the benefits and shortcomings of different architectures.

Robotics Architectures

Traditionally, many architectures in the field of robotics have made use of the state-action model. The state action model is based on the idea that the system can be described as a set of states [8]. The agent (e.g. a ground based robot) transitions from one state to another through actions. This is under the assumption that the environment remains constant unless acted upon by the robot.

Bonasso [9] is the pioneer of a three tiered (3T) intelligent control architecture which has been successfully implemented on a variety of robotics platforms (***Figure 3***). The deliberation layer evaluates goals, resources and timing constraints and outputs a partial list of ordered tasks called a Reactive Action Package (RAP). The sequencing layer decomposes a selected RAP, into a sequence of skill sets (basic agent commands e.g. move left) which are activated and deactivated to accomplish tasks. This architecture does not provide any way of enforcing hard real time limits on these specific skill sets. Furthermore, since all replanning (mission level and reactive) is performed by the deliberation layer, it is difficult to calculate the time required to generate a RAP as deliberative planners are generally symbolic in nature. This may not pose a problem for slow moving robots, but is a critical problem in UAS operations (e.g. reactive sense and avoid).

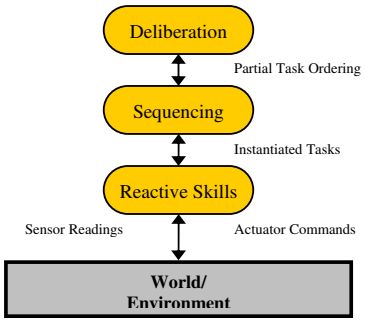


Figure 3 - 3T Intelligent Control Architecture [9]

The ATLANTIS architecture by Gat [8] is very similar to Bonasso's 3T architecture [9]. ATLANTIS also includes planning and reactive skills to allow the robot to operate in dynamic environments. The main difference is that ATLANTIS leaves the overall control of the system to the sequencing layer. Deliberation is treated as an activity that is scheduled by the sequencing layer. In situations where the computational urgency of the reactive component is greater (e.g. obstacle avoidance), the sequencer can temporarily suspend other ongoing deliberative activities. However, ATLANTIS also lacks a method for enforcing real time constraints.

Noreils [10] developed a three layer architecture with the aim of improving overall system reactivity. The highest two layers (planning and control) correspond approximately to the top two layers of Bonasso's 3T architecture [9] (Deliberation and Sequencing layers). However, the Functional (reactive) layer of Noreils' architecture includes sub-modules (***Figure 4***) which can independently trigger the appropriate actions (e.g. obstacle avoidance, target tracking) if the required command is not provided by higher layers in time. As a result, the use of independent sub-modules increases the extensibility of the architecture. Specialised modules are very beneficial for performing specific tasks which must meet real time deadlines. However, using specialised modules also results in added complexity as adding additional functionality requires the addition of new sub-module

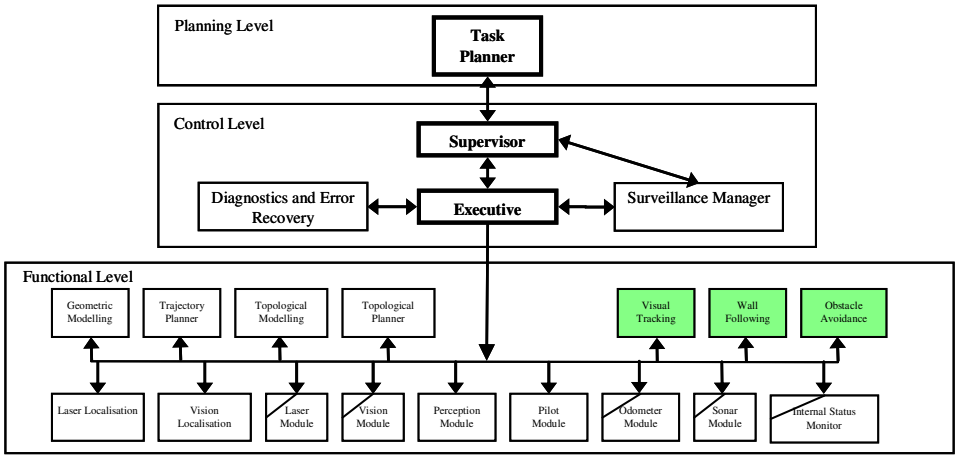


Figure 4 - Noreil's Architecture [10]

Arkin [11] presents an alternate architecture to the 3T architecture [9], entitled AURA. The deliberative layer here consists of a mission planner, navigator and pilot sub-system. The navigator performs mission level path planning; the pilot then constructs a linear path through this free space by calling upon available sensors and motor schemas (motor schemas comprise of low level navigational tasks e.g. trajectory tracking; similar to sub-modules found within Noreils' Functional layer [10]). The use of motor schemas limits the functionality of the agent to navigation only. No mention has been made about the inclusion of onboard payload activity (e.g. onboard camera control) scheduling; an important component for civil UAS operations (e.g. surveillance, disaster monitoring).

Brooks [12] presents another architecture known as Subsumption, which decomposes the control system problem into multiple modules (a module is an independent subsystem which is focused on completing a specific task), also referred to as behaviours. The simplest module is implemented first, and subsequently more complex modules are then implemented above it, providing more functionality (***Figure 5***). As more functionality is added to the robot, the system can quickly become very complicated. Furthermore; this architecture lacks flexibility, where once a complete system is implemented, it becomes very difficult to change the system functionality as each module is very task specific.

In addition, there are no provisions for fault detection and accommodation (FDA) in the robotics architectures reviewed here. This may not an issue for operations in controlled environments, but is a critical issue for UAS operations over populated environments (e.g. urban terrain).

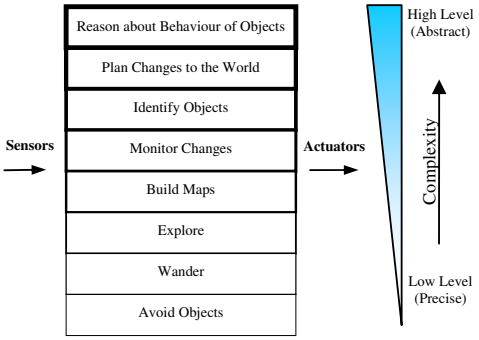


Figure 5 – Brooks' Subsumption Architecture [12]

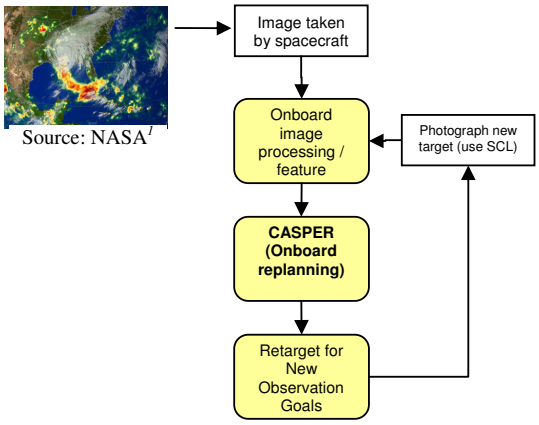
The architectures reviewed in this section represent the most common architectures used in robotics. A review of the architectures used in space based robotics is presented in the following section.

Space Based Architectures

A wide variety of architectures for onboard intelligence have been developed for space based systems. The operation of spacecraft bears many similarities to that of UAS; both deal with a remote semi-autonomous system that operate in the natural environment and must therefore deal with dynamic changes and uncertainty. The need for robustness in space-based applications is important, due to the significant financial cost of failure. Furthermore, both are constrained by finite resources (such as fuel and battery energy) and must meet stringent real time computational requirements. Both UAS and spacecraft are currently heavily reliant on human operation and receive commands for low level control (such as manoeuvre control) from manned ground stations. Consequently, both fields can benefit greatly from increased onboard autonomy. A brief overview of several key projects in space-based automation is presented here.

NASA's Autonomous Science-craft Experiment (ASE) [13] demonstrated automated scheduling and planning routines on the EO-1 Satellite launched in late 2000. This was the first time a space based mission was conducted autonomously using an intelligent control architecture (***Figure 6***) implemented onboard the spacecraft.

The cornerstone of the ASE architecture is the *Continuous Activity Scheduling Planning Execution and Replanning* (CASPER) [13] module which employs a repair based technique to: create a plan (which resolves conflicts that violate spacecraft constraints); propose a set of resolutions for a chosen conflict using a genetic algorithm; and choose the desired solution using heuristics. This process occurs iteratively until no more conflicts remain. The Spacecraft Command Language (SCL) uses rule based checking to convert this high level plan into low level commands. Therefore, even though the general concept is useful, the architecture itself is not focused on path planning and is instead concerned with scheduling of activities; which are critical aspects of UAS operations.



Source: NASA[1]

Figure 6 - NASA ASE architecture [13]

Another NASA based architecture is the Automated Planning/Scheduling Environment (ASPEN) [14] system which is essentially a software based application framework. It is an object oriented framework based on C++ that can intelligently schedule activities onboard the spacecraft. Activities are represented using state action models with the actual scheduling decisions performed using a parameter dependency network. This is similar in concept to a temporal constraint network [15] but extends its capabilities to include physical resources. A temporal constraint network is a graph based method for scheduling where nodes represent instances in time and edges represent time delays. ASPEN has been tested in numerous scenarios including onboard the New Millennium Earth Observing One (NM EO-1) satellite and Navy UHF Follow On One (UFO-1) satellite. Again, this architecture is targeted at the scheduling of activities rather than the path planning and execution problems, so important for UAS.

NASA's Remote Agent [16] is another intelligent control architecture designed for use onboard satellites. It employs a three tiered hierarchy similar to that presented by Bonasso [9]. The remote agent contains a set of decision making and scheduling tools to synthesise responses to unexpected situations which may arise during the mission (***Figure 7***). The Mission manager determines achievable goals both in the long term and current short term. The Planner/Scheduler takes these goals and uses a heuristic guided backtracking search to create an execution schedule of activities. Like ASPEN, plans are generated using temporal constraint network related methods. The Smart Executive plays a similar role to the sequencing layer in Bonasso's model and also makes use of Reactive Action Packages to implement these activities. Additionally, there is a Mode Identification and Reconfiguration subsystem that provides dynamic information on the status of the spacecraft. This provides an added layer of robustness as the execution of plans is dynamically modified by the perceived health of the spacecraft. Unlike the ASE and ASPEN scheduling systems, this architecture is more comprehensive as it includes methods for handling changes in the spacecraft's internal state as well as the external state. However, again, it is more focused on activity scheduling rather than path planning.

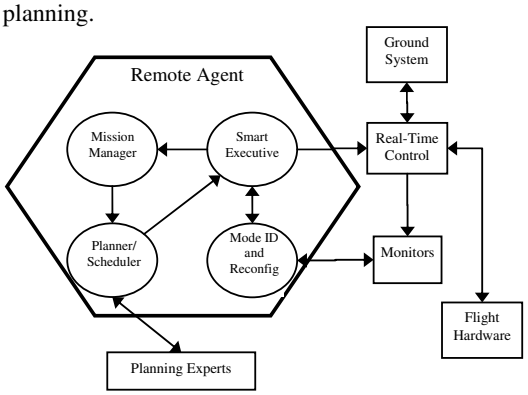


Figure 7 - NASA's Remote Agent Architecture [16]

The architectures reviewed in this section represent the most common architectures used in spaced based operations. A review of the architectures used within UAS is presented in the following section.

[1] *http://www.nasa.gov/mission_pages/hurricanes/multimedia/index.html*

UAS Architectures

Intelligent control architectures implemented onboard UAS are generally extensions of architectures found in robotics. However, many robotics architectures cannot be directly implemented in UAS. Firstly, UAS operate in highly dynamic environments where atmospheric changes can occur almost instantaneously; therefore, the agent's response must meet real time constraints. This is further compounded by the fact that aircraft typically travel at much greater velocities than ground based robots. Secondly, UAS dynamics can be highly non-linear and thus require careful consideration in the controller design and how this will interface to other subsystems onboard the aircraft. Finally, failures onboard UAS can be catastrophic and result: in loss of the UAS; property damage; and in the worst case, loss of human life as these robots are exposed to the general public.

A UAS can be thought of as a special type of robot that takes directives asking it to move from one location to another within a certain timeframe. Generally, there are two types of UAS: rotary/helicopter UAS that have the ability to brake and hover, and fixed wing UAS. Rotary UAS generally have shorter flight times while fixed wing UAS often have greater endurance but must always maintain some minimum (greater than stall) velocity.

Various architectures have been proposed that are specifically targeted at UAS. Schaefer [17] for example, presents a multi-layered UAS decision making architecture known as "Technologies for Reliable Autonomous Control (TRAC)". This is a variation of the 3T architecture pioneered by Bonasso [9] that has been augmented with another layer known as the Meta-Executive layer. The meta-executive layer is used to coordinate and synchronise interactions between the deliberative (which is goal driven e.g. performing a set of tasks based on accomplishing a particular goal) and execution (which is event driven e.g. performing a set of tasks based on a schedule) layers.

The TRAC architecture (***Figure 8***) revolves around a central data communications and storage module named the Active State Cache. The topmost, deliberative layer is called Closed Loop Execution and Recovery (CLEaR). This is responsible for high level mission management and task sequencing. Plans created by CLEaR are acted upon by the Autonomous Command Executive (ACE) which oversees the execution of mission plan elements. Beacon-based Exception Analysis (BEAM) and Spacecraft Health Inference Engine (SHINE) are subsystems that monitor the health of the unmanned vehicle in real time. The TRAC architecture is an extension of that developed in the NASA Remote Agent project. Significant emphasis has been placed on the importance of low level fault detection and Identification (FDI) through the inclusion SHINE and BEAM subsystems. The ACE subsystem can deal with some reactive situations, but this is limited to terrain avoidance and stability corrections during wind gusts [18]. There is no specific subsystem onboard to deal with reactive collision avoidance, a necessity for flight operations within the NAS.

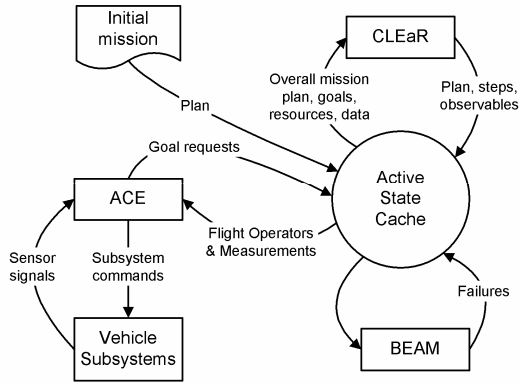


Figure 8 - TRAC System Structure [19]

The NASA APEX software robotics Architecture [19] is also based on Bonasso's 3T architecture [9]. This architecture has been successfully applied to a range of applications, notably that of NASA's Autonomous Rotorcraft Project. The upper two layers of APEX are collectively referred to as Reasoning and Control Services (RCS). This architecture emphasises reusability through modularity and thus separates RCS procedures (which are the most reusable) from lower layer modules which are less reusable.

Boskovic [20] presents a UAS architecture which is optimized for navigation, in similar fashion to the upper layers of AURA [11]. The layers within this architecture are defined with respect to specific UAS functionalities rather than generic functions in robotics. The highest layer (***Figure 9***) in this hierarchal four layered model is the Decision Making layer. This layer uses *a priori* information in conjunction with information obtained from sensors to make appropriate decisions to achieve mission goals. The next level is the Path Planning Layer which generates mission waypoints. If an obstacle is detected that was not known *a priori*, then the waypoints are recomputed online. There is no communications subsystem to give the operator any decision making capability throughout the mission. Again, there is no specific subsystem to deal with reactive collision avoidance.

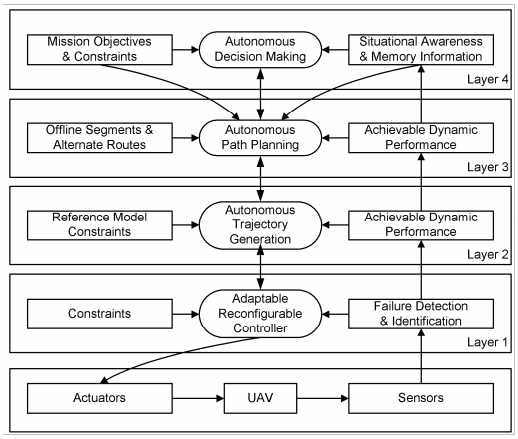


Figure 9 - Boskovic's UAS Decision Making Architecture [20]

The WITAS UAS project [21] presents yet another intelligent control architecture which has deliberative, reactive (like sequencing) and control layer components. It is best to view this as a reactive concentric architecture as the individual processes at the various levels of abstraction are executed concurrently at different latencies (high level path planner runs at higher latency than low level controllers). The deliberative layer here contains a collection of path and trajectory planners, predictors and recognition packagers. A set of flight control modes such as hovering, dynamic path following and take-off and landing, are automated using sets of Task Procedures (similar to a RAP in 3T). However, in order to switch between autonomous flight modes, it is necessary for the UAS to brake and hover before executing the next Task Procedure. Doing so decreases the operational efficiency the UAS; furthermore, this strategy is infeasible for fixed wing UAS. Finally, there is no explicit provision for handling collision avoidance.

The open control platform (OCP) [22] is similar to APEX in that it too is a software robotics architecture based on Bonasso's 3T architecture [9], which can potentially be applied to UAS operations. The top layer here comprises of supervisory tasks such as: data management; event detection and situation awareness. The middle layer (reconfigurable control) performs mode transitioning stability control (e.g. use of an adaptive control algorithm during transition between approach and landing phases) whilst the lower level is dedicated to trajectory tracking (low level controller implantation). All internal communications makes use of middleware (CORBA) with custom extensions implemented to ensure hard real time execution of commands. This also allows the architecture to operate as a distributed network (similar to the WITAS Project) and different components can be written in different languages. Similar to other UAS architectures, there are no specific subsystems to deal with external communications.

Summary of Findings

From the literature review, it was found that the vast majority of architectures were hierarchical. This approach was often used to separate slower, deliberative planning processes from faster, time-critical hardware control systems [9]. Additionally, it allows for abstraction of complexity from one layer to the next; this is useful not only in reducing subsystem complexity, but also helps in software reusability [19]. The vast majority of architectures employed some variation of Bonasso's 3T hierarchy [9] which had separate layers for deliberation, sequencing of actions and control execution.

Ideally, a human operator should only need to interact with the high level deliberative layer. In this scenario, the operator performs high level tasks such as specifying mission goals and the schedule associated with these goals. In these instances, there is a need for a communications subsystem that provides the link between the remote agent and the ground station. Such a communications module is incorporated into the ASE, APEX and Remote agent architectures [13, 16, 19].

It was found that in many UAS and spacecraft based architectures that an important capability was a method for monitoring the agent's internal state (i.e. the health of the vehicle) and its impact on vehicle performance. This was implemented as a form of Fault Detection and Accommodation (FDA) in TRAC, Remote Agent and OCP and in Boskovic's architecture [16, 17, 20, 22].

At the same time, it is important to have accurate knowledge of the external environment in which the agent is situated. It was found that even though all architectures made provisions for a sensing mechanism, very few explicitly explored the computational complexities involved in processing sensor information for use in higher level planning algorithms. Obviously, very little processing is required for low level control systems as raw data, such as position and velocity can feed directly into an actuator control module. However, avoiding dynamic obstacles when generating manoeuvres requires predictions of the current and future state of the dynamic obstacle. Therefore, analysis of sensor data is required to transform it into a form usable by the higher level algorithms through sensor fusion. There is currently no UAS architecture which explicitly separates the sensor data requirements between lower (raw sensor data) and higher layers (accurate state information calculated through sensor fusion methods).

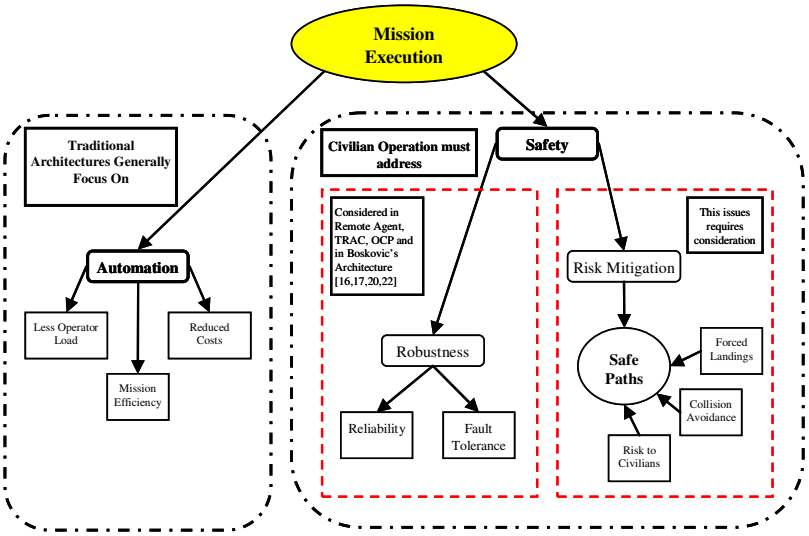


Figure 10 - Summary of the functional requirements for an architecture during civilian UAS operations

As the majority of UAS operations require positioning the aircraft in the right place at the right time, it can be seen that there is a need for a robust architecture that provides this path planning and execution functionality. It was found that the majority of architectures focus on only the mission execution component of UAS operations and do not explicitly provide for a method of ensuring the safety of the aircraft and the minimisation of risk to other aircraft and people on the ground. There has been some work in the areas of fault tolerance and reliability, but risk mitigation (actions to reduce the impact of a risk) has not fully been addressed (***Figure 10***).

Furthermore, the vast majority of architectures do not provide a complete end to end architecture from goal deliberation to planning to action execution (with the exception of Boskovic [20]). However, Boskovic's architecture does not address the problem of managing risks in during UAS operations. There is no communications subsystem to give the operator any decision making capability throughout the mission. Also, there is no specific subsystem to deal with collision avoidance. A proposed architecture is presented in the following section addressing these critical issues.

Proposed Architecture

We propose an architecture for civil UAS operations based on Boskovic's [20] architecture. This architecture not only accommodates path planning and FDA, but also includes provisions for intelligent execution of activities not explicitly involved in path planning. As well, it also encompasses modules dedicated to ensuring the safety of the aircraft. At this point in time, all aspects of high level decision making however, are left to the responsibility of the Human Operator (e.g. choosing which goals to pursue). In terms of efficiency, this architecture provides the potential to reverse the current Civilian UAS trend from many operators monitoring a single UAS, to a single operator monitoring multiple UAS.

To allow the human operator interaction with the onboard high level deliberative layer, a communications subsystem has been included to allow real time interaction between a human operator and UAS activities (e.g. uploading new mission goals) during the flight operation.

In the previous section, it was concluded that, none of the current UAS architectures explicitly separate the sensor data requirements between lower and higher layers. In the proposed architecture, raw sensor data is forwarded to lower layers (which have real time requirements) in an effort to minimise any lag which may occur with processed data. Raw sensor data is also forwarded to a sensor fusion subsystem which generates an accurate approximation of the aircraft state. This data is stored within the Integrated Shared State Memory (ISSM – similar to Schaefer's Active State Cache [17]) which can be accessed by the relevant layer.

To incorporate risk mitigation strategies within the proposed architecture, specific modules (similar to Noreil's Architecture [10]) to deal with sense and avoid and forced landing situations were used. The sense and avoid module performs detection of obstacles within the immediate vicinity of the aircraft. This data is used by the manoeuvre generation layer to perform the appropriate collision avoidance manoeuvres. Likewise, the forced landing site classifier module is used to detect potential landing sites during critical onboard failures.

Currently, Boskovic's architecture does not include any functionality for scheduling and control of payloads (e.g. camera, lights) as it is focussed purely on the navigational aspects of UAS operations. The proposed architecture includes an activity scheduler and controller. The activity scheduler creates a schedule of payload activities by synchronising start and finish times using mission time and aircraft state. The activity controller activates and deactivates the relevant payload. This feature allows the UAV to perform a range of missions including and not limited to surveillance, disaster monitoring and search and support. A detailed representation of the Proposed UAS Architecture is presented in ***Figure 11***

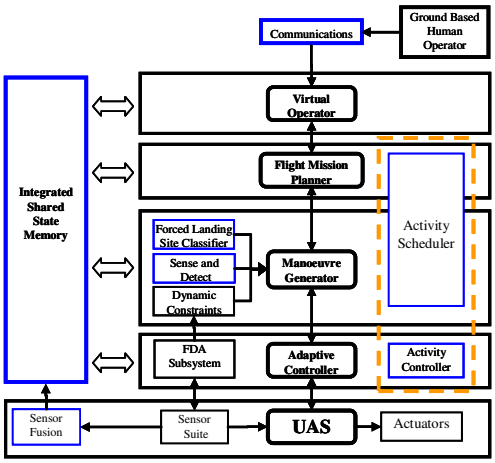


Figure 11 - Proposed Civilian UAS Architecture

Virtual Operator

The Virtual Operator (VO) is concerned with providing mission goals to the lower layers and simultaneously monitoring the state of the UAS in assessing whether these goals have been achieved or not. When communicating with the mission flight planner, the VO provides a set of prioritised goals which may be defined in terms of spatial position, velocity, aircraft orientation and time. The mission flight planner calculates a path based on these goals and returns the costs (in terms of fuel and time for example). With this information, the VO can then make a decision as to whether to continue the mission using the current plan or to reject this plan and create a new one by changing one or more mission goals. A similar process is necessary for interaction with the Activity Scheduler. When communicating with the Activity Scheduler, the goals the VO provides are instead activities that need to be performed when the UAS reaches certain states. For example, an activity could be to turn on the camera and begin capturing images when the aircraft is at a specified location.
In the case that the VO is the human operator, the operator has the full authority (and accountability) to choose the order of operations. Once the operator has decided on an operations schedule, the schedule can be uploaded to the unmanned aircraft via the communications channel and stored in onboard memory. The VO has the authority to update the operations schedule and the maps used for planning (e.g. terrain maps) throughout the mission. In the absence of certifiable decision making techniques that can meet the requirements of such a VO, it is envisaged that a semi-autonomous VO module, coupled to a Human Machine Interface (HMI) and human operator would constitute the VO shown in this architecture.

Mission Flight Planner

The Mission Flight Planner is in essence a multi-objective path planner that evaluates multiple criteria in determining an optimal path for the aircraft. It receives goals, which may comprise multiple prioritised waypoints from the VO. As well, it obtains information about the environment through multi-resolution maps stored in the memory module. These maps could include terrain data, risk data (risk associated with overflying certain areas) and predictions of dynamic obstacles (such as other aircraft).

Additionally, the planner also obtains from this dynamic memory the current aircraft position, fuel load and other related constraints (such as the need to maintain within line of sight of the operator). The VO is informed of the costs involved in reaching each waypoint and of any unreachable waypoints. At the same time, the mission flight planner passes to the Manoeuvre Generation subsystem a path which effectively consists of a series of intermediate goals (or waypoints). When there are changes to the environment (which is reflected in the data obtained from dynamic memory), or significant deviation of the aircraft from the planned route as reported by the Manoeuvre Generation subsystem, the mission flight planner replans a new flight path. If any of the intermediate goals are unachievable, the Manoeuvre Generation subsystem modifies the maps in memory and marks these unachievable regions as no-go.

Manoeuvre Generator

The Manoeuvre Generation Layer generates a feasible local path between a set of intermediate waypoints given by the Flight Mission Planner. A feasible path is one that is collision free and which satisfies aircraft dynamic and kinematic constraints.

Basic Manoeuvres (e.g. flying straight and level, pitching, rolling and yawing motion) can be

combined together to create more complex manoeuvres. Representing aircraft motion as a set of manoeuvres is essential for flight in civilian airspace. For example, consider a scenario whereby the UAS is instructed to reach a higher altitude to avoid other aircraft, but is not physically able to do so due to the constraints of restricted airspace and the limitations of the aircraft's manoeuvrability (insufficient rate of climb). A candidate solution in this situation is to perform a spiral manoeuvre.

A sense and detect capability within UAS is essential for flight in segregated airspace. The sense and detect subsystem uses onboard sensors (e.g. vision) for detection of obstacles (static and moving). The manoeuvre generation algorithm then uses the data provided by the sense and detect unit to generate collision free path segments, and to perform emergency collision avoidance manoeuvres if necessary. Carnie [23] is currently investigating the use of machine vision to provide sense and detect capabilities onboard UAS at the Australian Research Centre for Aerospace Automation (ARCAA).
The forced landing classifier is used to detect potential landing sites during flight, in case the UAS needs to perform an emergency landing due to onboard failures which cannot be accommodated (e.g. engine failure). This information is input to the Manoeuvre Generation algorithm, which provides the adaptive controller with a suitable landing trajectory for tracking. Fitzgerald [24] has conducted research into detection and classification of potential forced landing sites at ARCAA.

Adaptive Controller

The low level controllers are designed to ensure aircraft stability at all times. A broad range of techniques are available to create the desired response including: Proportional, Integral and Differential (PID); State Space; Fuzzy; Optimal; to mention a few [6].

Small scaled UAS generally, do not have the available onboard payload capacity to include sensor redundancy. If sensor failure occurs without detection, this can lead to critical failure as the stability controllers will receive incorrect or no state information. Critical Failure can also occur if an actuator becomes inoperable.

Fault Detection and Accommodation algorithms (FDA) are used to detect if a particular sensor is providing erroneous data, and allowing continuation of operations by disable the erroneous sensors operation. This however leads to reduction in aircraft performance, as fewer sensors are now available to provide an estimation of the UAS state. This information is conveyed to the Dynamic Constraints Subsystem for recalculation of new UAS dynamic and kinematic constraints. Cork [23] is currently investigating FDA techniques to reduce the effects of erroneous sensors at ARCAA.

Concluding Remarks

It is apparent that the operation of UAS in civilian applications requires an equivalent level of safety to that of manned aircraft. Achieving this level of safety requires, in addition to system robustness, an intelligent system that is capable of both tactical and strategic planning to minimise the risk involved when undertaking a mission. At the same time, the system must also be able to execute emergency procedures in the event of hardware failures.

Through an investigation of existing architectures in unmanned aircraft, space based systems and robotics, it was found that few offered a framework that catered for the path planning and manoeuvre generation aspects of onboard intelligence in light of the needs of sensor integration. Many have considered mission scheduling and fault detection and accommodation, but few have integrated this with the aforementioned path planning and execution elements with a focus on emergency scenarios; including and not limited to collision avoidance and forced landings. Furthermore, even fewer have considered the multiple criteria, in terms of airspace regulations, mission objectives and mission safety that must be considered in civil UAS operations.

To address these deficiencies, an intelligent control architecture for UAS was devised that addresses the requirements of intelligent planning, execution and handling of emergency scenarios. This architecture encompasses many subsystems that are currently being developed at ARCAA. It is envisaged that the integration of the various components in this architecture would help increase the level of intelligence onboard unmanned aircraft in terms of mission efficiency and increased safety. This is not only paramount to the acceptance of UAS in the NAS, but will also allow for decreased operator workload and thus reduce operational cost.

Acknowledgements

The authors wish to acknowledge the support of the Queensland University of Technology (QUT), the Australian Research Centre for Aerospace Automation (ARCAA) and the Commonwealth Scientific and Industrial Research Organisation (CSIRO).

Bibliography

1. Cox, T., Civil UAV Capability Assessment (Draft Version). 2004, NASA. p. 103.
2. Wegener, S., UAV Autonomous Operations for airborne Science Missions. 2004, American Institute of Aeronautics and Astronautics.
3. Australia, C.A.A.o., Advisory Circular - UAV Operations, Design Specification, Maintenance and Training of Human Resources. 2002.
4. EUROCONTROL, Specifications for the use Of Military Unmanned Aerial Vehicles as Operational Air Traffic Outside Segregated Airspace. 2006.
5. Tzafestas, S., ed. Methods and Applications of Intelligent Control. 1997, Kluwer Academic: Boston.
6. Gupta, M.S., N, ed. Intelligent Control Systems - Theory and Applications. 1996, IEEE Press: New York. 820.
7. Roberts, P., et al., En-Route Human Pilot Automation Considerations for future UAV Systems. 2005, Queensland University of Technology: Brisbane.
8. Gat, E. Integrating planning and reacting in a heterogenous asynchronous architecture for controlling real-world mobile robots. in National Conference for Artificial Intelligence (AAAI). 1992.
9. Bonasso, R.P., et al. Experiences with an Architecture for Intelligent, Reactive Agents. in International Joint Conference on Artificial Intelligence. 1995.
10. Noreils, F.R. and R.G. Chatila, Plan execution monitoring and control architecture for mobile robots. Robotics and Automation, IEEE Transactions on, 1995. 11(2): p. 255-266.
11. Arkin, R. Motor schema based navigation for a mobile robot: An approach to programming by behavior. in IEEE International Conference on Robotics and Automation. Proceedings. 1987.
12. Brooks, R., A robust layered control system for a mobile robot. Robotics and Automation, IEEE Journal of [legacy, pre - 1988], 1986. 2(1): p. 14-23.
13. Rabideau, G. Mission Operations with Autonomy: A Preliminary Report for Earth Observing-1. in 4th International Workshop on Planning and Scheduling for Space. 2004. Darmstadt, Germany.
14. Fukunaga, A., et al. Towards an application framework for automated planning and scheduling. in IEEE Aerospace Conference. 1997.
15. Schwalb, E. and R. Dechter, Processing disjunctions in temporal constraint networks. Artificial Intelligence, 1997. 93(1-2): p. 29-61.
16. Muscettola, N., et al., Remote Agent: to boldly go where no AI system has gone before. Artificial Intelligence, 1998. 103(1-2): p. 5-47.
17. Schaefer, P., et al. Technologies for reliable autonomous control (TRAC) of UAVs. in Digital Avionics Systems Conferences. 2000.
18. Johnson, T.L., et al. The TRAC mission manager autonomous control executive. in IEEE Aerospace Conference. 2001.
19. Freed, M., et al., An Architecture for Intelligent Management of Aerial Observation Missions. AIAA, 2005. 2005-6938.
20. Boskovic, J.D., R. Prasanth, and R.K. Mehra. A multilayer control architecture for unmanned aerial vehicles. in American Control Conference. 2002.
21. Doherty, P., et al. A Distributed Architecture for Autonomous Unmanned Aerial Vehicle Experimentation. in 7th International Symposium on Distributed Autonomous Systems. 2004. Toulouse, France.
22. Wills, L., et al., An open platform for reconfigurable control. Control Systems Magazine, IEEE, 2001. 21(3): p. 49-64.
23. Cork, L., R. Walker, and S. Dunn. Fault Detection, Identification and Accommodation Techniques for Unmanned Airborne Vehicles. in Australian International Aerospace Congress. 2005. Melbourne.
24. Fitzgerald, D., R. Walker, and D. Campbell. A Vision Based Forced Landing Site Selection System for an Autonomous UAV. in Intelligent Sensors, Sensor Networks and Information Processing Conference. 2005.