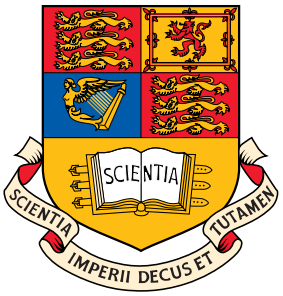
Prohibited Volume Avoidance for Aircraft

by

Rushen Bhikhubhai Patel



Department of Aeronautics
Imperial College London

This thesis is submitted for the degree of Doctor of Philosophy

2010

QINETIQ/TIS/PTS/JP1100002

. . . to the British taxpayer.

Abstract

This thesis describes the development of a pilot override control system that prevents aircraft entering critical regions of space, known as prohibited volumes. The aim is to prevent another 9/11 style terrorist attack, as well as act as a general safety system for transport aircraft.

The thesis presents the design and implementation of three core modules in the system; the trajectory generation algorithm, the trigger mechanism for the pilot override and the trajectory following element. The trajectory generation algorithm uses a direct multiple shooting strategy to provide trajectories through online computation that avoid pre-defined prohibited volume exclusion regions, whilst accounting for the manoeuvring capabilities of the aircraft. The trigger mechanism incorporates the logic that decides the time at which it is suitable for the override to be activated, an important consideration for ensuring that the system is not overly restrictive for a pilot. A number of methods are introduced, and for safety purposes a composite trigger that incorporates different strategies is recommended. Trajectory following is best achieved via a nonlinear guidance law. The guidance logic sends commands in pitch, roll and yaw to the control surfaces of the aircraft, in order to closely follow the generated avoidance trajectory. Testing and validation is performed using a full motion simulator, with volunteers flying a representative aircraft model and attempting to penetrate prohibited volumes.

The proof-of-concept system is shown to work well, provided that extreme aircraft manoeuvres are prevented near the exclusion regions. These hard manoeuvring envelope constraints allow the trajectory following controllers to follow avoidance trajectories accurately from an initial state within the bounding set. In order to move the project closer to a commercial product, operator and regulator input is necessary, particularly due to the radical nature of the pilot override system.

Acknowledgements

First and foremost, I would like to thank my supervisor Dr. Paul Goulart for his help and guidance over the course of this PhD. Paul has always had his door open, and without his knowledge and experience, the journey would have undoubtably been more painful. Working with Paul has reinvigorated my interest in science and mathematics, and for that I will always be grateful. I would also like to thank Dr. Varnavas Serghides for his input, particularly in the modeling and testing phases of the work. My thanks go to Rod Angel of QinetiQ for valuable discussions and feedback throughout the project. I enjoyed helping Devang Patel and Paul Gilmartin with their Masters projects and learnt some valuable lessons along the way. Dr. Eric Kerrigan was my first point of contact at Imperial, and thanks go to him for directing me towards this research topic. Generous financial support from the Engineering and Physical Sciences Research Council and QinetiQ is gratefully acknowledged.

A number of people from the Department of Aeronautics have made my time at Imperial an enjoyable one. There are those who were here before I arrived: Adelaide de Vecchi, Bruno Carmo, Gustavo Assi, Alaa Qubain, Peter Vos and the indomitable Ian Couchman; those I started with: Xuerui Mao, Julien Hoessler and Oliver Rupert Hadlow Buxton; and those who came after: David Pearson, Alessandro Bolis, Duncan McNae, Mark Potter and Nick Fogell. I wish you guys all the best in the future.

Finally, I would like to thank my parents who taught me the importance of education and have supported my endeavours wholeheartedly. I could not have asked any more of them.

I hereby declare that this thesis, and the research to which it refers, are the product of my own work, and has not been submitted for any other award. Ideas and quotations from the work of other people, published or otherwise, are fully acknowledged.

Rushen B. Patel
London
October 1, 2010

Contents

Abstract **2**

Table of Contents **5**

List of Figures **8**

List of Tables **11**

1 Introduction **16**

1.1 Description of PV Avoidance Problem and Solution Strategy 17

1.2 PV Avoidance System Specifications 18

1.3 PV Avoidance System Control Structure 20

1.4 Considerations for the PV Avoidance System 20

1.5 Application to Unmanned Aircraft Systems 21

1.6 Organisation and Highlights 22

1.7 Related Publications 23

2 Background, Part I: Aircraft Systems **25**

2.1 Aircraft Avionics Structure 25

2.1.1 Flight Control Systems 25

2.1.2 PV Avoidance System, Inputs and Outputs 27

2.2 Design Considerations for Flight Control Systems 28

2.3 Aircraft Conflict Detection and Resolution Systems 29

2.3.1 Ground Proximity Warning System 29

2.3.2 Traffic Collision Avoidance System 30

2.3.3 Review of Conflict Detection and Resolution Methods 30

2.3.4 H-metaphor and Parallels with the PV Avoidance System 30

3 Background, Part II: Control and Modeling Concepts **32**

3.1 Path Planning 32

3.1.1 Definitions and Terminology 32

3.1.2 Solution Categories . . . 33
3.2 Convex Analysis and Optimization . . . 34
3.2.1 Convex Sets and Convex Functions . . . 35
3.2.2 Polar, Support and Gauge Functions . . . 38
3.3 Optimal Control . . . 41
3.4 System Dynamics: Aircraft Equations of Motion . . . 42
3.4.1 3DOF Dynamics . . . 43

4 Trajectory Generation Scheme 47
4.1 Modelling PV Regions . . . 47
4.2 Optimal Control Problem and Solution Strategy . . . 49
4.2.1 Discretizing the Optimal Control Problem . . . 51
4.2.2 State Continuity and Rate Constraints . . . 51
4.2.3 Problem Formulation . . . 52
4.2.4 Non-differentiable State Exclusion Regions . . . 54
4.2.5 The Relationship Between Mixed Integer Linear Programming and Dualization methods for Disjunctive Programming . . . 58
4.2.6 Trajectory Generation Outer Loop Controller . . . 61
4.3 Extension of OCP2 to the Infinite Horizon . . . 61
4.3.1 Terminal Invariance Condition . . . 62
4.3.2 Warm Starting . . . 67

5 Implementation and Results for Trajectory Generation 69
5.1 Choice of Integration Scheme and Error Bounds . . . 69
5.2 Integration Scheme - Numerical Experiments . . . 74
5.3 Software Implementation . . . 76
5.4 Trajectory Generation Results . . . 77
5.5 Algorithm Performance . . . 78
5.5.1 Single Hemispherical PV . . . 79
5.5.2 Hemispherical and Cylindrical PVs . . . 80
5.5.3 Polyhedral PV . . . 81
5.5.4 Urban Environment Simulation . . . 81
5.6 Warm Starting . . . 82
5.7 Extension to Unmanned Aircraft . . . 83
5.7.1 UAV Model . . . 84
5.7.2 UAV Path Planning Problem Formulation . . . 85
5.7.3 Results . . . 88
5.7.4 Conclusions and Future Work in UAV Guidance . . . 91

6 Override Trigger Mechanism **94**

6.1 Problem Formulation and Relationship to Trajectory Generation Outer Loop Controller . . . 94

6.2 Two-Solution Method . . . 95

6.3 Exact Penalty Function Method . . . 95

6.3.1 Exact Penalty for a General Optimization Problem . . . 96

6.3.2 Exact Penalty for the Aircraft-Specific Problem . . . 98

6.4 PV Entry Method . . . 100

6.5 Trigger Mechanism Results . . . 103

6.5.1 Two-Solution Method . . . 103

6.5.2 Exact Penalty Function Method . . . 104

6.5.3 PV Entry Method . . . 105

6.6 Choice of Trigger Mechanism . . . 106

7 Trajectory Following Inner Loop Controllers and Testing **108**

7.1 Simulation Software . . . 108

7.2 Control System Architecture . . . 109

7.3 Set Point Tracking . . . 111

7.4 Cartesian Trajectory Tracking . . . 117

7.5 Control from Extreme Attitudes . . . 122

7.6 Testing . . . 126

7.6.1 Procedures . . . 126

7.6.2 Flight Simulation Results . . . 127

8 Conclusions for the PV Avoidance System **131**

8.1 Major Contributions of the Thesis . . . 131

8.2 Directions for Future Work . . . 132

A Linearized Model Optimization **135**

A.1 Optimization of the Linearized System . . . 135

A.2 Test Cases . . . 136

A.3 Quantifying the Maximum Trajectory Prediction Error due to Linearization . . . 138

B X-Plane Simulation Software **140**

B.1 X-Plane Flight Model . . . 140

B.2 External Communication with X-Plane . . . 141

B.3 Simulation Screenshots . . . 141

C Boeing 767 Flight Model Development **143**

C.1 Calculation of Lift and Drag Polars Using XFoil 143

C.2 Calculation of Parasite Drag, C_{D_0} 145

C.3 Thrust Characteristics 147

Bibliography **149**

List of Figures

1.1 PV Avoidance System. 19
1.2 System flow diagram. 20
1.3 Schematic of control system design. 21

2.1 Generic flight control system for a modern transport aircraft. Note the number of sections and positioning of control surfaces is dependent on aircraft type. Based on [64]. 26
2.2 Input and output diagram from PV Avoidance System. 27
2.3 Flight control law design process. Based on [71]. 28

3.1 Examples of convex and non-convex sets. 35
3.2 A hyperplane defined by $a^T x = b$ in $\mathbb{R}^2$ determines two halfspaces. 36
3.3 Polytope. 37
3.4 Norm balls in $\mathbb{R}^2$. 37
3.5 Ellipsoid in $\mathbb{R}^2$. 38
3.6 Examples of polar sets. The sets shown are polars of each other. 39
3.7 Support and gauge Functions. 40
3.8 Coordinate system for the 3DOF aircraft model. 44

4.1 Avoidance trajectory. 49
4.2 Multiple shooting 50
4.3 The subdifferential of some scalar convex functions as a function of the argument x. Based on [8]. 55
4.4 The unit square and its polar. The vertices correspond to the binary variables in the big-M formulation. 61
4.5 Polyhedral PV and scaling via the gauge function. If $\lambda > 1$ the aircraft will remain outside the PV. 65
4.6 Projected velocity far from PV. 66
4.7 Projected velocity touching PV. 66

5.1 Integration test cases. 75
5.2 The maximum integration error for x (blue), y (red) and z (green) states over five simulations. There are 50 integration steps with a step size of 0.4s. 76
5.3 Left: relative computation time vs integration discretization. Right: 2-norm error in position vs integration discretization. Euler is shown in blue and RK2 in red. 77
5.4 Software Structure. 78
5.5 Solve time = 0.24s, 13 IPOPT iterations. 80
5.6 Solve time = 0.44s, 21 IPOPT iterations. 80
5.7 Solve time = 0.65s, 27 IPOPT iterations. 81
5.8 Solve time = 0.51s, 22 IPOPT iterations. 81
5.9 Two avoidance trajectories with a 1.2s time interval. 82
5.10 Comparison of reference trajectory initialized and shift-initialized problems. The trajectory used for the initialization is shown on the right. 83
5.11 Time-optimal problem formulation. 87
5.12 Fixed time-to-target problem. Solve time = 0.25s. 89
5.13 Fixed time-to-target problem with obstacle constraints. Solve time = 0.87s. . . . 89
5.14 Time-optimal problem with no obstacle constraints. Solve time = 0.77s. 90
5.15 Time-optimal problem with obstacle constraints. Solve time = 1.47s. 91
5.16 Left: initialized with the reference trajectory (shown in green). Right: initialized with the trajectory obtained when the computation is first run with no obstacles. 91
5.17 Time-optimal problem with a moving obstacle constraint (50m separation). Solve time = 1.51s. 92
5.18 Time-optimal problem with a moving obstacle constraint (80m separation). Solve time = 1.45s. 92
5.19 Autonomous control scheme for unmanned vehicles. 93

6.1 Two solution method trigger mechanism. 96
6.2 Top - pilot applies extreme input and penetrates the PV. Bottom - override activated before PV penetration can occur. 101
6.3 Two solution trigger method. Solve time = 1.1s. 104
6.4 PV entry problem for a 10s time horizon. Solve time = 0.32s. 105
6.5 Schematic of control system design with composite trigger. The override is triggered if *either* the PV Entry computation converges to a solution *or* the input soft constraint is violated in the trajectory generation computation. 107

7.1 Schematic of control system design. 110
7.2 Set point tracking 111

7.3 Simulation result using PID tracking of (α, μ). 113
7.4 Simulation result using PID tracking of (γ, μ). 114
7.5 Simulation result using PID tracking of (γ, μ) with a large initial yaw input applied. 115
7.6 Simulation result using PID tracking of (γ, μ) with a 30 knot crosswind. 116
7.7 Nonlinear guidance law 117
7.8 Linear analysis for following a perturbed line 118
7.9 Simulation result using Cartesian trajectory tracking with a 30 knot crosswind. . 120
7.10 Simulation result using Cartesian trajectory tracking (increased integral gain for lateral guidance) with a 30 knot crosswind. 121
7.11 Simulation result using Cartesian trajectory tracking. The initial aircraft state is outside the conventional operating bounds of the control system. 124
7.12 Simulation result using Cartesian trajectory tracking. The aircraft is in an extreme initial state and the trajectory following method is unable to fly the generated trajectory resulting in loss of control. 125
7.13 PV Avoidance System testing in the full motion simulator. 126
7.14 Input time series for a typical PV penetration attempt. 128
7.15 Flight simulation results for an extreme manoeuvre performed on entry into the envelope trigger cylinder. Note the poor tracking of the avoidance trajectory due to the high initial roll angle. 130

A.1 Linearized solution: solve time = 0.24s, 16 IPOPT iterations. 137
A.2 Linearization error corresponding to solution of Figure A.1 137
A.3 Linearized solution: solve time = 0.38s, 23 IPOPT iterations. 137
A.4 Linearization error corresponding to solution of Figure A.3. 138

B.1 X-Plane screenshot of the Boeing 767 model in flight 141
B.2 Pilot's cockpit view in X-Plane. 142

C.1 Lift and drag curves, NACA 65-412 corrected at $Re = 10.5 \times 10^6$ 144

List of Tables

5.1 Integration error bounds for the Euler and RK2 methods. 74
5.2 Comparison between conservative bound and actual simulation values of integration error. 76
5.3 Computation times for forward problem. 83

6.1 Results for increasingly difficult problems as the aircraft closes on a PV. 105

7.1 State and input bounds. 122
7.2 Envelope trigger bounds. 123
7.3 Flight simulation results. 127

C.1 Component breakdown method for parasite drag. 146
C.2 PW4056 engine parameters. 147
C.3 List of terms. 148

Nomenclature

Abbreviations

BVP	Boundary Value Problem
CDR	Conflict Detection and Resolution
CFIT	Controlled Flight Into Terrain
DOF	Degrees of Freedom
EGPWS	Enhanced Ground Proximity Warning System
FBW	Fly-By-Wire
FPGA	Field Programmable Gate Array
GPS	Global Positioning System
GPWS	Ground Proximity Warning System
HJB	Hamilton-Jacobi-Bellman
MILP	Mixed Integer Linear Programming
MPC	Model Predictive Control
NLP	Nonlinear Programming
ODE	Ordinary Differential Equation
PD	Proportional-Derivative
PDE	Partial Differential Equation
PID	Proportional-Integral-Derivative
PMG	Permanent Magnet Generator
PV	Prohibited Volume
RAT	Ram Air Turbine
SLAM	Simultaneous Localization and Mapping
TA	Traffic Advisory
TCAS	Traffic Collision Avoidance System
UAV	Unmanned Aerial Vehicle

Aircraft Symbols

α	angle of attack
χ	heading angle
γ	flight path angle
μ	roll angle
ξ	closure angle
a_{lat}	lateral acceleration
a_{long}	longitudinal acceleration
D	drag force
k_I	integral gain
L	lift force
L_{lat}	lookahead distance for lateral tracking
m	mass
R	turning radius
T	engine thrust
t_p	throttle setting
V	ground velocity
v	true airspeed
x_c	x-position of aircraft in Cartesian coordinates
y_c	y-position of aircraft in Cartesian coordinates
z_c	z-position of aircraft in Cartesian coordinates

Control Theory Symbols

η	dual variable for target set condition
$\gamma_C(\cdot)$	gauge function
$\operatorname{int} C$	interior of set C
κ_f	terminal control law
$\mathbf{u}$	control sequence
$\mathbf{x}$	state sequence
$\mathcal{L}(\cdot)$	Lagrangian
$\mathcal{P}(\cdot)$	optimal control problem
$\mathcal{X}_\mathcal{N}$	set of feasible initial conditions
$\mathcal{X}_f$	target set
$\phi(\cdot)$	propagation of system dynamics

C°	polar set
$\sigma_C(\cdot)$	support function
τ	trigger override parameter
$\mathcal{B}_p$	p-norm unit ball
$\|(\cdot)\|_Q$	weighted 2-norm
C_j	state exclusion set (prohibited volume)
$f(\cdot)$	system dynamics
$f^*(\cdot)$	Fenchel conjugate
$J(\cdot)_N$	objective function with horizon N
M	number of state exclusion sets
N	number of multiple shooting time steps
P	terminal cost matrix (state)
p	position vector at a given state
Q	stage cost matrix (state)
R	stage cost matrix (input)
t_f	time horizon for optimization
u_i	input vector at time step i
v	velocity vector at a given state
x_i	state vector at time step i

Chapter 1

Introduction

This thesis describes the development of a counter-hijack safety system to prevent a pilot, hostile or otherwise, from flying a civilian airliner into critical infrastructure, such as buildings, bridges and landmarks. The intention is not to protect every building in a city, nor to prevent the aircraft from being flown into the ground, but instead to protect certain high profile artefacts from terrorist attack. These might include single large buildings such as skyscrapers; complexes of buildings such as power stations, Parliament and Buckingham Palace; or areas of grouped potential targets such as Canary Wharf. Virtual protective zones known as *prohibited volumes (PVs)* are defined around those artefacts which are to be protected.

A question that arises is whether such a system is truly needed. Due to heightened security measures in airports and restricted aircraft cockpit access the likelihood of a situation similar to that of 9/11 occurring might appear slim. However, freight aircraft have the potential to be used in a similar fashion and, with the number of aircraft in the skies rapidly increasing as developing countries mature and global population expands, the risk is still present. Indeed, in February 2003, CNN reported that America had installed anti-aircraft batteries and regular patrols of fighter jets to ensure the safety of high profile targets and would continue to do so on occasions where the terrorist threat assessment is deemed to be high [25]. As Peter Huber of the Manhattan Institute Think Tank pointedly described in October 2001 [45]:

> A fully fueled jumbo jet is about as lethal an instrument as ever gets entrusted to civilian hands. Nuclear power plants and big hydroelectric dams are far safer. They don't have to fly, so they can be encased in vast excesses of concrete, and indeed they are. Assaults with nuclear or biological weapons can't begin with cardboard cutters. They require substantial factories somewhere in the background, which can, we must hope, be identified and knocked out well before they are up and running. There are many other potential instruments of terror, but none quite so essential, ubiquitous and now terrifying, as civilian jets.

It is important to develop the technological ability to prevent such attacks. At the very least this gives policy makers the choice of whether or not to implement a preventative system of this type.

1.1 Description of PV Avoidance Problem and Solution Strategy

The PV avoidance problem can be stated simply as:

> *During all phases of flight (take-off, cruise and landing) the aircraft must remain outside a set of predefined prohibited volumes, despite any actions taken by the pilot that would otherwise result in penetration of one or more of these prohibited volumes.*

There are a number of potential strategies for ensuring an aircraft remains outside PVs. Three possibilities are:

- A *ground based override* where control is seized from the onboard pilot and handed to a pilot who flies the aircraft from a ground station. In this strategy the onboard pilot no longer has any control of the aircraft.
- A *biased controls* approach where the pilot's inputs to the aircraft are progressively biased to 'push' the aircraft away from a PV. This is related to the pilot through haptic feedback in the flying controls. In this strategy a pilot flying directly towards a PV would find that the controls automatically bias the aircraft to fly a divergent path. In this approach the pilot still retains the control authority to manoeuvre away from a PV.
- An *onboard override* approach where the pilot's input is overriden by a control system on the aircraft that automatically manoeuvres the aircraft away from the PV. In this strategy the pilot no longer has any control of the aircraft.

The biased controls approach has been explored previously by Lee [61] in 2001. In this work a game-theoretic approach is used with two players (the pilot and the controller). It requires a solution over the entire state space which in turn requires large amounts of computational power and storage capacity. Although feasibility was demonstrated for a simple aircraft model operating in a 2D environment, higher order would be computationally prohibitive. A similar biased controls approach was explored by Grootendorst [38, 14] for lateral evasive manoeuvres when approaching a straight line no-go boundary. Simulation results showed increased safety with the system implemented. However, the haptic element caused some frustrations for the test pilots. Another concept which is currently being explored by NASA, known as the H-metaphor, involves a combination of biased controls and direct override. This work is reviewed

in detail in Section 2.3.4. A ground based override (known as the Robolander system) was discussed frequently after 9/11[27, 28]. The basic technology is already feasible. However, the system has not been implemented to date.

A ground override approach has the advantage of ensuring the safety of the passengers since the aircraft can be landed at an appropriate location by the ground based pilot. However, a serious drawback is the requirement for safety critical external communication with the aircraft. Interference, deliberate or otherwise, could potentially have disastrous consequences, particularly if large fleets of aircraft operate with this strategy.

The biased controls approach is likely to compare favourably when compared to a complete override in terms of pilot acceptance. However the work of Lee and Grootendorst have only shown feasibility in simplistic environments with the aircraft operating in two dimensions. Basic avoidance manoeuvres are used (fixed radius turn) and it remains to be seen whether these methods could work in a representative environment, where many PV constraints and possible trap situations are present.

In this thesis the third approach of a complete pilot override is adopted. The potential feasibility of this approach, whilst not assured, is greater than in the biased controls approach, and the strategy does not suffer from the inherent dangers associated with a ground based override.

The override system proposed in this thesis, referred to as the *PV Avoidance System*, will not guarantee the lives of innocent people onboard a hijacked aircraft and will return control back to the pilot once there is no longer any danger of PV penetration. There is no mechanism for preventing a hostile pilot from crashing the aircraft into the ground or continuing to fly the aircraft until it runs out of fuel. The primary aim is to protect structures encapsulated by the prohibited volumes (see Figure 1.1). In principle, these other safety mechanisms are possible but are not addressed in the research scope of this thesis. As a goal for future generations of the system, capabilities could be enhanced to include landing at the nearest safe airfield.

A set of outline PV Avoidance System specifications were generated with input from the project sponsor QinetiQ and are outlined in the following section.

1.2 PV Avoidance System Specifications

The PV Avoidance System proposed in this thesis should prevent an aircraft from penetrating a PV. This can be decomposed into the following functional specifications:

- Analyze the current aircraft trajectory in real-time.
- Recognize when the aircraft trajectory could penetrate a prohibited volume.
- Provide warnings to the pilot, giving enough time to change the current course of action.

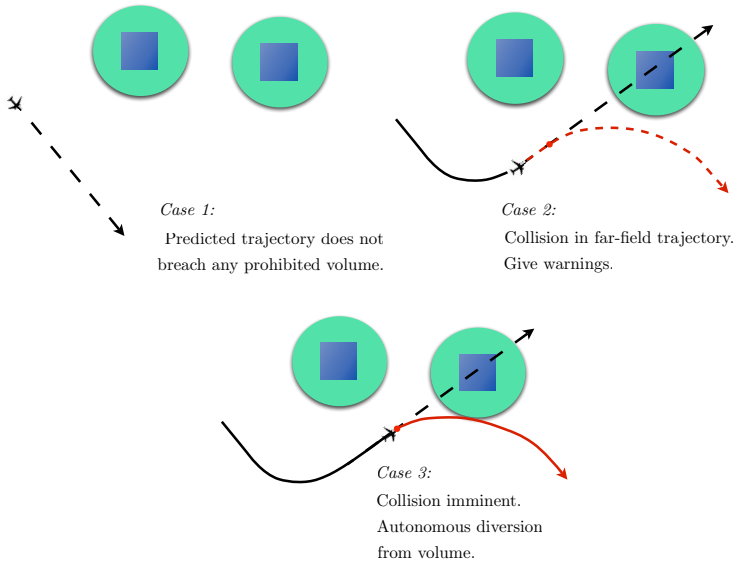


Figure 1.1: PV Avoidance System.

- Generate safe avoidance trajectories that are within the normal manoeuvring envelope of the aircraft.
- Remove control authority from the pilot when the potential for prohibited volume penetration is high, and automatically fly an avoidance trajectory.
- Return control authority to the pilot once the danger of prohibited volume penetration has passed.

Figure 1.2 shows the system flow diagram of the proposed method. Two control loops are shown. The outer loop generates avoidance trajectories, whilst the inner control loop follows the generated trajectory.

The system must be adaptable for different aircraft types. It must be fully contained onboard the aircraft in an inaccessible location with no disabling mechanism. A database of prohibited volumes will be constructed off-line with the input of governments and regulatory authorities to determine their size, shape, location and number. The system will have a selection of PV shapes that can be used, with a suitable PV or combination of PVs selected depending on the shape of the structure to be protected. For the purposes of the proof-of-concept system the ground is assumed to be flat and the PVs are placed at ground level. Terrain is an important concern and must be added as a factor for consideration. It is ignored in this work for ease of simulation and to ensure completion within the time constraints of the project. The methods developed in the thesis however are compatible with variable height terrain and this is discussed in the Future Work section of Chapter 8.

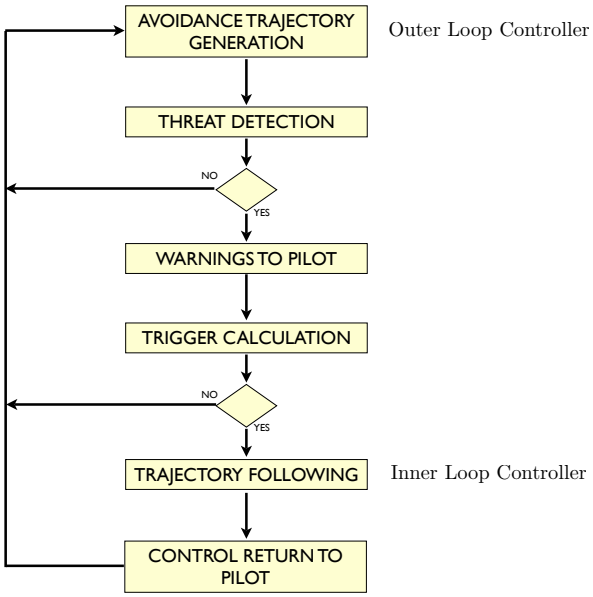


Figure 1.2: System flow diagram.

1.3 PV Avoidance System Control Structure

A control structure schematic is shown in Figure 1.3. The key elements are trajectory generation, the trigger mechanism and trajectory following. The trajectory generation scheme is always active (or can be in a stand-by mode at cruise altitudes), but an avoidance trajectory will only be used once the trigger mechanism is activated. The trigger mechanism activates the override when the potential for PV penetration is found to be too high, after which the trajectory following controllers fly an avoidance trajectory. Figure 1.3 is referred to throughout the thesis to highlight the particular system module under discussion.

1.4 Considerations for the PV Avoidance System

The technical challenges in designing algorithms to the specifications of Section 1.2 make up the majority of the work in this thesis. Aside from these there are substantial challenges involved in deploying a PV Avoidance System from the regulatory, political and cultural viewpoint. Pilots themselves might be uncomfortable to fly with a system which cannot be disabled, and potentially removes control authority. Significant regulatory hurdles need to be overcome to implement a PV Avoidance System. Since a pilot override of the nature considered in this thesis has not been implemented previously, regulations do not currently exist dictating how such a system should operate. However, current autopilot systems provide automatic guidance (albeit managed by the pilot), that is covered by the Federal Aviation Regulations (FAR 25.1329). Along with regulations that state quick disengagement controls are required, FAR 25.1329(g) and FAR 25.1329(h) stipulate that the guidance system should not exert excessive loads on the

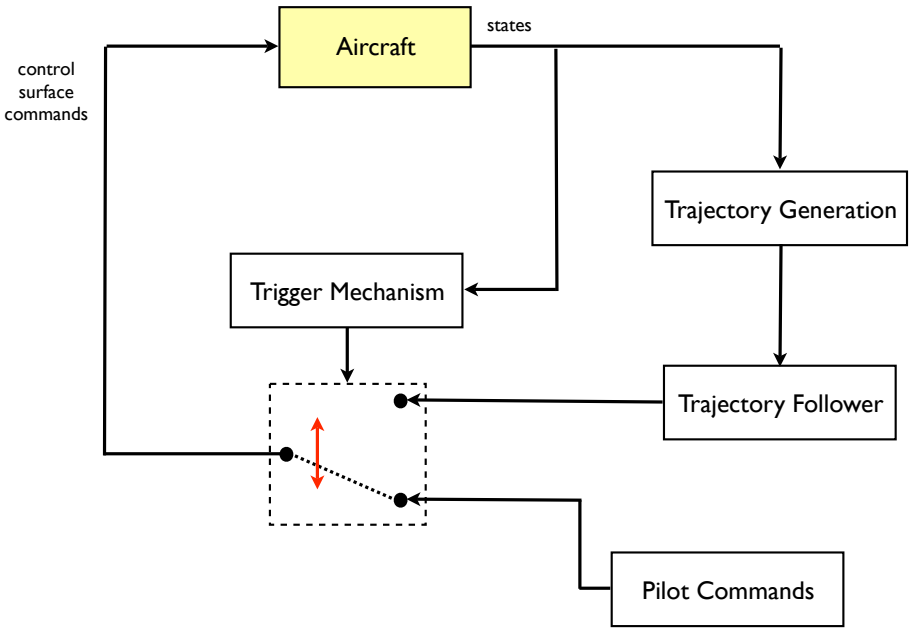


Figure 1.3: Schematic of control system design.

aircraft and should ensure the speed range is kept within the normal flight envelope. These considerations are also relevant for the PV Avoidance System.

To truly mitigate the threat of another 9/11 style attack, deployment of the system on all operational civilian aircraft above a certain size would be required. The threshold size is determined by the fuel carrying capacity of the aircraft since fuel was considered to be the major contribution to the fire that caused the collapse of the Twin Towers in the 9/11 case [89].

The core contribution of this thesis is the development of a proof-of-concept system to demonstrate feasibility of a PV Avoidance System. If further work is pursued towards a commercial solution, these additional issues would become extremely important and it is recommended that regulatory and political input be sought at an early stage of that process. They are not considered further in this thesis.

1.5 Application to Unmanned Aircraft Systems

The research presented in the thesis has implications for unmanned aircraft systems. Unmanned aircraft are seeing increased use in both military and civilian settings. Collision avoidance for these aircraft will most likely be necessary if they are to meet the safety requirements of operation in non-segregated airspace. Methods viable for real-time systems such as that proposed in this research could well become an integral part of an unmanned aircraft safety architecture. More generally, the increased processor speed of computers has made real-

time optimal control of reasonably complex systems feasible, and techniques developed in this research can be applied to a number of different industrial problems.

1.6 Organisation and Highlights

The major contribution of this thesis is directed to the trajectory generation element (outer loop controller) of the complete PV Avoidance System. This is the most technically challenging aspect, and of research interest to a wide audience.

Chapter 2:
In this chapter the basic aircraft flight control systems are described. An analysis of where the PV Avoidance System could exist in the avionics architecture is presented, and the typical design process for a new flight control system is outlined. Finally, a review of currently operational conflict detection systems is given.

Chapter 3:
This chapter starts with a review of the literature on current path planning methods for differentially constrained systems (e.g. aircraft), with a particular emphasis on model predictive control methods. Aspects of convex analysis and optimization are addressed, and the final part of the chapter presents the system dynamics for the aircraft.

Chapter 4:
This chapter describes the application of the direct multiple shooting control strategy for avoidance trajectory generation. The problem statement is described formally in mathematical terms, and a novel method is presented for remodeling non-differentiable obstacle constraint sets (e.g. polytopes) to preserve continuity using duality. This allows the use of a standard gradient based optimization algorithm. The dualization method is further extended to construct positively invariant target sets that ensure the problem remains feasible at the terminal state of each of the finite-horizon avoidance trajectories. These conditions, when combined with a warm-start method based on shift-initialization of prior solutions, ensure the optimization is initialized close to a feasible point for the non-convex avoidance trajectory generation problem.

Chapter 5:
In the first part of the chapter global error bounds are derived for different trajectory integration schemes, and some numerical experiments are presented showing the actual error in various simulation scenarios. An overview of the software implementation is given and results are presented for the trajectory generation scheme using a representative transport aircraft model.

The final part of the chapter describes how the trajectory generation techniques can be used in unmanned aircraft systems in a number of different scenarios.

Chapter 6:
In this chapter three methods for triggering the pilot override are investigated. The first uses the avoidance trajectory optimization algorithm, and examines feasibility of the computation at some point in the future. The second uses an exact penalty function to modify the optimization algorithm to include additional soft constraints that heavily penalize a deviation of control inputs at the start of the trajectory. The final method examines whether there exists a control sequence to cause penetration of a no-go zone. The three trigger mechanisms are compared for effectiveness.

Chapter 7:
The first part of this chapter describes two methods that are investigated for following the generated trajectories. The first is proportional-integral-derivative (PID) control of the inputs generated from the trajectory generation algorithm. A second method based on the work in [68], uses a nonlinear control law similar to proportional navigation to follow the Cartesian trajectory in 3-dimensional space. The second method in particular shows good performance, even in the presence of disturbances such as wind. The second part of this chapter describes testing of the complete PV Avoidance System in the full motion simulator at Imperial College London using a number of volunteers with flying experience, acting as hostile pilots and attempting to penetrate PVs.

Chapter 8:
This chapter presents conclusions from the overall work carried out on the PV Avoidance System and describes future work required to move the project forward.

1.7 Related Publications

Chapters 4, 5 and 6 of this work have been published in the following papers:

1. R.B. Patel, P.J. Goulart and V. Serghides. *'Real-Time Trajectory Generation for Aircraft Avoidance Maneuvers'.* Proceedings of the AIAA Guidance, Navigation and Control Conference, Chicago, Illinois, 2009.

2. R.B. Patel, P.J. Goulart. *'Trajectory Generation for Aircraft Avoidance Maneuvers using Online Optimization'.* AIAA Journal of Guidance, Control and Dynamics, Vol 34:1, 2011

3. R.B. Patel, P.J. Goulart. *'The Design of Trigger Mechanisms for Aircraft Collision Avoidance Maneuvers'.* Proceedings of the AIAA Guidance, Navigation and Control Conference, Toronto, Canada, 2010.

The direct multiple shooting method applied to avoidance trajectory generation as described in Chapter 4, and some numerical simulation results from Chapter 5 are presented in *1.* The dualization work of Chapter 4 and further results are presented in *2.* The different trigger mechanism strategies from Chapter 6 are presented in *3.*

Chapter 2

Background, Part I: Aircraft Systems

2.1 Aircraft Avionics Structure

In this section the flight control system of a typical civilian airliner and positioning of the PV Avoidance System in the top level systems architecture is described. Since each aircraft make and type have unique system architectures, the description in this section is kept as general as possible.

2.1.1 Flight Control Systems

A generic flight control system for medium to large transport aircraft is shown in Figure 2.1. Although the number of control surfaces and flight control computer implementation vary from aircraft to aircraft, all modern fly-by-wire (FBW) systems have this generic form. Fly-by-wire aircraft are not a prerequisite for use of this system, and in theory any aircraft where the pitch, roll and yaw can be commanded via an electrical signal can adopt the system. For each individual aircraft type, the aircraft's particular system architecture would determine how the PV Avoidance System is incorporated.

The key control functions for a civilian aircraft are as follows [64]:

- Pitch control via flight control actuators powering the elevator surfaces.
- Pitch trim via actuators which move the horizontal tailplane surface, stabilizer or trim tabs.
- Roll control via the left and right ailerons, augmented as required by the extension of a number of spoilers on the inboard wing sections.

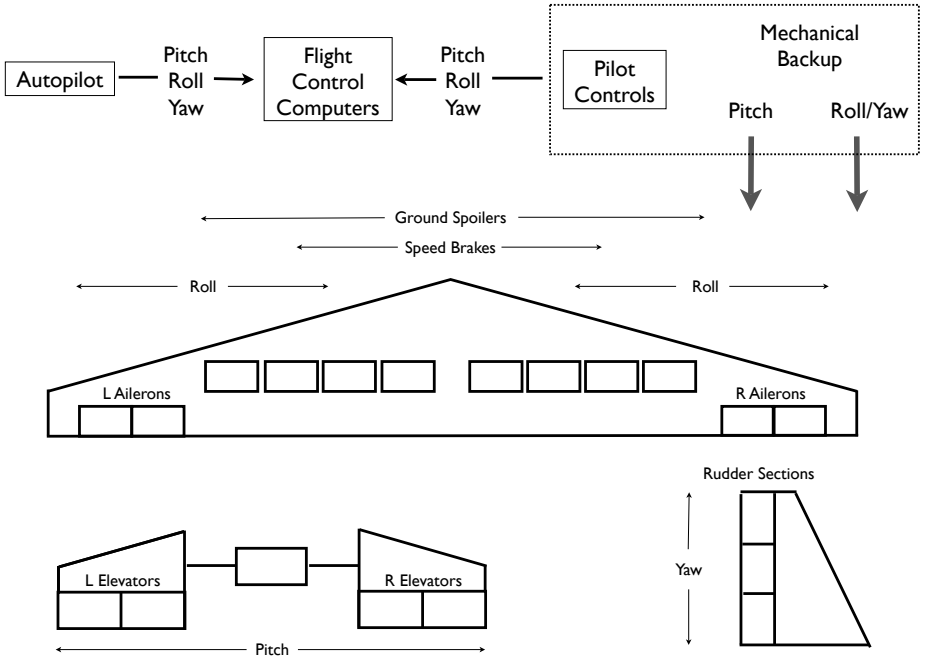


Figure 2.1: Generic flight control system for a modern transport aircraft. Note the number of sections and positioning of control surfaces is dependent on aircraft type. Based on [64].

- Yaw control via rudder sections.
- Both wing sets of spoilers may be extended together to perform the following functions:
  - Inboard spoiler sections provide a speed brake function in flight for rapid deceleration.
  - Use of all spoilers provide a lift dump function.

Direct inputs from the pilot's controls or inputs from an autopilot feed the necessary guidance signals into a number of flight control computers depending on the system architecture. These computers modify the flight control demands according to a number of aerodynamic and other parameters such that effective and harmonized handling characteristics are achieved.

Many modern civil aircraft using FBW systems employ some form of mechanical and/or hydraulic back-up system to ensure basic functionality in the event of a complete power failure [64]:

- In the case of Airbus, the mechanical trim wheel can alter the position of the tailplane surface for pitch control. Inputs from rudder pedals can alter the inputs to the three rudder actuators.
- In the Boeing system, alternate pitch trim levers are mechanically connected to the horizontal stabilizer. A direct mechanical link from the rudder to one pair of spoilers allows roll control to be maintained in a standby mode.

Other aircraft types have comparable back-up modes.

2.1.2 PV Avoidance System, Inputs and Outputs

The inputs and outputs for the proposed PV Avoidance System are shown in Figure 2.2. Air data and inertial reference data is supplied by an Air Data and Inertial Reference Unit or some variation of the two depending on aircraft type. This ensures that GPS is not the only source of navigational data. There is often a back-up unit providing this data in case there is degradation in the primary source. For the PV Avoidance System, input signals are required giving the current aircraft state. These signals affect the model parameters used in computations within the system. An independent power source is required for the system that cannot be disabled (most electronics can be disabled in case of fire).

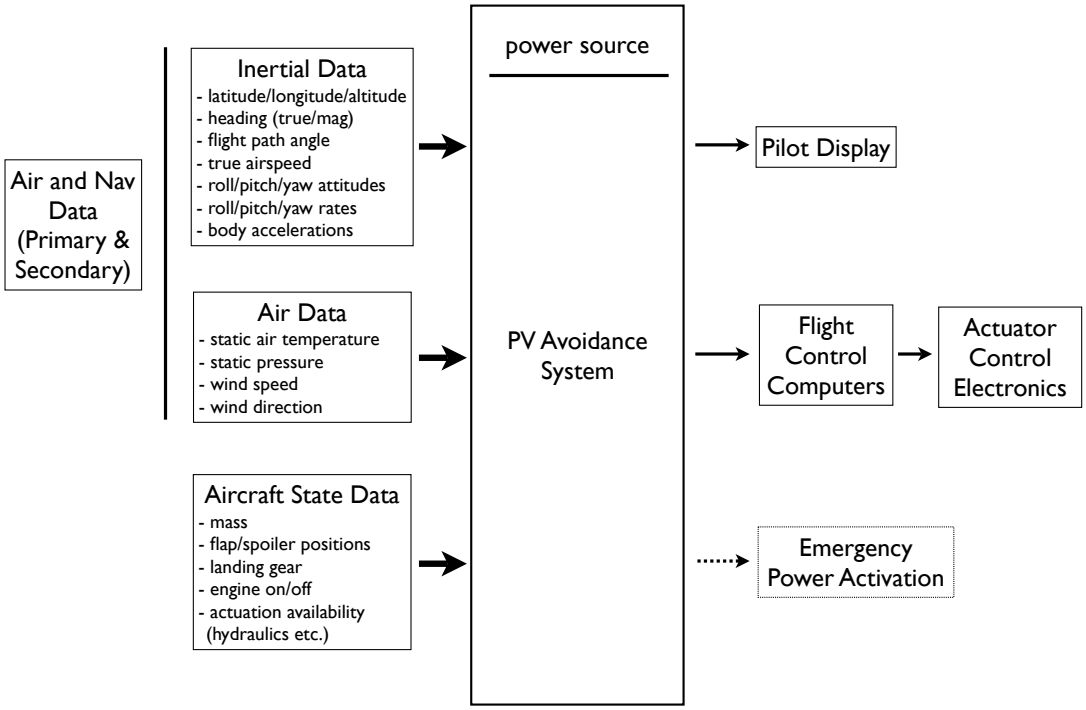


Figure 2.2: Input and output diagram from PV Avoidance System.

Given that in a modern aircraft a pilot can intentionally disable the engine and hydraulics for control surface actuation, the PV Avoidance System must be capable of activating the supply of emergency electrical and hydraulic power. The three standard methods of providing back-up power on civil aircraft are [64]:

- Ram Air Turbine (RAT)
- Back-up converters

- Permanent Magnet Generators (PMGs)

It would be essential for any onboard counter-hijack system to be able to deploy independently a means of providing enough back-up power to move the aircraft control surfaces. It will also need to ensure operability of other systems so that the required navigational data and air data are available, or else have an independent means of obtaining this data.

2.2 Design Considerations for Flight Control Systems

For the purposes of this thesis the main focus is on the algorithmic design of the proposed system. Although a full implementation of the design procedures for aircraft flight control systems is outside the scope of this proof-of-concept work, it is worth briefly reviewing the control system development process. Figure 2.3 shows the flight control law design process.

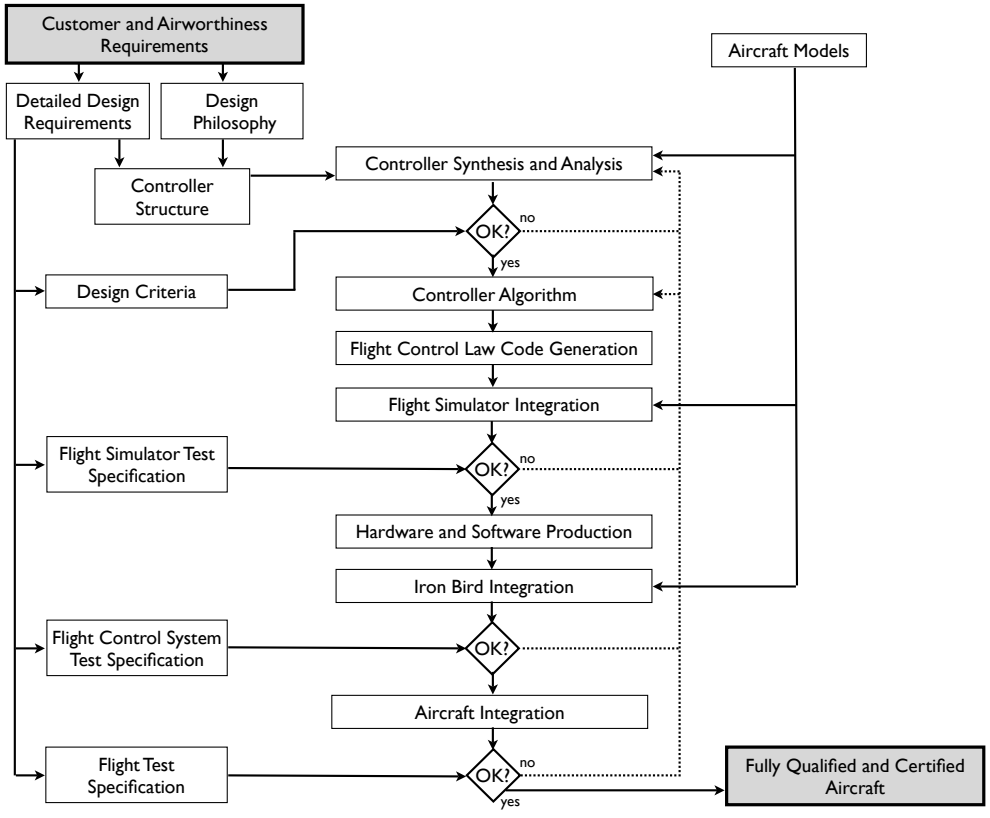


Figure 2.3: Flight control law design process. Based on [71].

The system specifications are outlined in Section 1.2. In this thesis, the technical feasibility of the proposed solution is investigated up to flight simulator integration and the approach broadly follows that shown in the diagram. Beyond that, significant resources are required to further develop the solution. 'Iron bird' integration involves a ground test rig where mechanical elements of the control system are integrated with simulation, and successful completion of this

type of testing results in aircraft integration and flight testing.

In the following section, systems for resolving air-to-air or air-to-ground collisions are reviewed.

2.3 Aircraft Conflict Detection and Resolution Systems

There are currently no operational systems that are specifically intended to detect or prevent an aircraft collision with a building. Terrain following radar could, in some circumstances, induce a combat jet to pull up enough to miss large hangar sized buildings [85]. Stall barriers or stick pushers are present to prevent pilots from stalling aircraft. Warning systems exist that inform the pilot of proximity to the terrain and to other aircraft. The Ground Proximity Warning System (GPWS) and Traffic Collision Avoidance System (TCAS) are the most relevant for this research and will be described in further detail. A comprehensive review of these and other proposed systems can be found in [55].

2.3.1 Ground Proximity Warning System

The leading cause of worldwide aviation fatalities comes from inadvertently flying a fully functioning aircraft into the ground or water. This type of accident is referred to as Controlled Flight Into Terrain (CFIT). Since the International Civil Aviation Organisation recommended the fitting of GPWS systems on all passenger and freight aircraft in 1979, the number of CFIT accidents has dropped dramatically. Despite this, from 1985 to 1995 the number of airline fatalities due to CFIT was recorded as 2219 worldwide, still representing 49% of the total [13].

The GPWS system, developed in 1967, uses a radar altimeter to monitor the aircraft's height above the ground. Warnings are given to the pilot through visual and audio messages when any of the five following modes of undesirable aircraft behaviour occur - 1) Excessive barometric sink rate with respect to terrain clearance; 2) Excessive rate of terrain closure with respect to terrain clearance; 3) Excessive altitude loss after take off; 4) Unsafe terrain clearance with respect to phase of flight; 5) Excessive descent below the Instrument Landing System glide slope. A flaw of the traditional GPWS system is that it only gives data from directly below the aircraft, therefore an upcoming dramatic change in terrain elevation cannot be detected. In 2002 the Enhanced Ground Proximity Warning System (EGPWS) was released which combined the worldwide digital terrain database with Global Positioning System (GPS) technology to allow the system to predict where future problems might arise due to rapidly changing elevation (e.g. cliffs or mountains).

2.3.2 Traffic Collision Avoidance System

The Traffic Collision Avoidance System (TCAS), which began to be used prominently in the 1990s, is designed to prevent mid-air collisions between aircraft [64]. TCAS equipped aircraft carry an interrogator which 'interrogates' all other aircraft within a specified range about their position, with each of the aircraft responding using a transponder. The interrogation and response occurs several times per second. If a collision risk is detected a series of graded 'traffic advisories' would be issued. If the collision risk persists a 'resolution advisory' to descend or climb is given to the pilot. The time horizon for TCAS advisories is 25s. TCAS warnings take priority over Air Traffic Control directions. GPWS warnings take priority over both.

Any system developed through this research will be required to work with these existing warning systems onboard the aircraft as well as the flight management and autopilot systems in current use.

2.3.3 Review of Conflict Detection and Resolution Methods

Assuming a non-hostile pilot, Kuchar and Yang produced a comprehensive review of current methods for conflict detection and resolution (CDR) systems [55]. A CDR system should predict when a conflict will occur in the future, communicate the detected conflict to a human operator and, in some cases, assist in the resolution of the conflict situation. A number of the models reviewed by Kuchar use optimization approaches which combine kinematic aircraft models with a set of cost functionals chosen to induce some favourable system behavior. An optimal resolution strategy is then determined by solving for the trajectories with the lowest cost. TCAS, for example, searches through a set of potential climb or descent manoeuvres and selects the least aggressive maneuver that still provides adequate protection [64]. For CDR systems where the pilot is expected to carry out the corrective action, economic costs or operator workload are often incorporated as design considerations.

2.3.4 H-metaphor and Parallels with the PV Avoidance System

The H-metaphor is a concept that is currently being explored at the NASA Langley Research Center [35, 36, 40] for future aircraft (and other vehicle) control systems. It is slightly outside the domain of the CDR methods described above, and is best understood via analogy to riding a horse. In an H-metaphor control system the pilot is connected to the aircraft through haptic (manipulative touch, tactile and kinesthetic cues) which draw an equivalent to the *seat of the pants* and *reins* for a horse rider. The idea is that a pilot with a heavy workload can 'point' the aircraft in the desired direction in a rough sense and the aircraft will fly in the desired direction avoiding hazards automatically along the way - much in the same way a horse would. The autonomous decision making is relayed to the pilot though the movement of the aircraft

(acceleration cues) and haptic feedback through the pilot controls. This concept is further broken down into *loose rein* and *tight rein* modes. In loose rein mode the aircraft recognizes that the pilot is relatively disengaged and flies autonomously, but with the pilot still in-the-loop through designating longer term tasks. In tight reins mode the pilot applies harder forces on the joystick to overcome the autonomous control and directly manoeuvre the aircraft. The concept leaves open the possibility of a complete override of the pilot commands (through insurmountable feedback forces on the pilot controls) if the risk of catastrophic accident is high enough.

Although the aim of the H-metaphor concept is to increase safety and reduce operator work load for single pilot operations, there are some parallels with the work in this thesis, particularly in the hazard avoidance area. In preliminary simulation-based testing the authors use simple potential field methods to avoid obstacles. These are discussed further in Section 4.2.4. The trajectory generation concepts explored in Chapter 4 provide an alternative method of generating trajectories, which can overcome some of the disadvantages of potential field methods (in particular conservatism, incorporating vehicle constraints and inability to detect trap situations).

Chapter 3

Background, Part II: Control and Modeling Concepts

This chapter starts with a review of path planning methods for vehicle guidance and control. The mathematical definitions relevant to the PV avoidance problem formulation are introduced, the primal-dual interior-point optimization method is described and approaches to solving optimal control problems are briefly summarized. Finally the aircraft system dynamics used for trajectory generation are presented.

3.1 Path Planning

An integral part of the complete PV Avoidance System is the outer loop controller for generating avoidance trajectories (see Figure 1.2). This type of problem is encompassed within the overall category of *path planning* problems. Path planning is encountered in a number of fields including robotics, artificial intelligence, computer graphics and control theory.

3.1.1 Definitions and Terminology

A *path* is a curve traced by the vehicle in the state space and a *trajectory* is a path including the time along the path [58]. A motion planning algorithm is considered to be *complete* if it returns a path when one exists. It is *optimal* if it returns the optimal path with respect to some criterion and *feasible* if it returns a valid solution, without consideration to optimality. A *sound* planner will generate a path that guarantees no obstacle will be hit despite uncertainty in sensing and control [33]. Throughout this work the terms *prohibited volume* and *obstacle* are freely interchangable.

The path planning problem formulated in this thesis relates to a differentially constrained (non-holonomic) system where the equations of motion act as constraints on the path (for

differentially unconstrained problems the vehicle may use infinite accelerations to achieve a path). This class of problems is generally more difficult to solve due to the coupling of time and state space through the differential constraints.

3.1.2 Solution Categories

There are numerous methods that have been investigated in path planning for non-holonomic systems. General categories are given below, though there is some crossover between the methods.

- *Sampling-based trajectory planning.* Involving a search over the entire state space, these methods include gradient descent over a navigation function [60] and rapidly-expanding random tree algorithms [75] [57] [59] [31]. They have been demonstrated as effective and computationally efficient for online motion planning. However, in certain cases convergence can be slow and the trajectories generated far from optimal [33].

- *Decoupled trajectory planning.* This method is based on a two stage approach. First a discrete path of waypoints is generated in the state space. This path is then used as the basis for a trajectory for the non-holonomic system. Well-known algorithms such as A* or the Voronoi approach can be used in the first stage [58] [98]. In [49] a Voronoi based planner is used in conjunction with spline smoothing for producing trajectories. Yakimenko [97] optimizes over a family of polynomials to approximate the two-point boundary value problem. This approach is further explored by Cowling et. al. [17], where the differential flatness property [67] of the equations of motion are used to express the state and control vectors as a function of the output vector. Trajectories are parameterized using polynomials and the inverse dynamics used to determine states and controls at each discretized time point. Optimization then takes place over the output space allowing constraints such as obstacles to be dealt with more readily, but in turn, this results in more difficulty incorporating input constraints.

- *Finite state motion model (Manoeuvre Automaton).* Finite state motion models are used to reduce the optimization from an infinite-dimensional space to a finite problem. In the context of vehicle trajectory planning, this model is called a manoeuvre automaton (MA). Frazzoli et. al. used trim states and a set of trajectory primitives to perform aggressive manoeuvres for an autonomous helicopter [30]. Dever et. al. extended the framework to allow interpolation within the manoeuvre boundary conditions in a class of manoeuvre [18]. Although model constraints are incorporated, direct inclusion of obstacles was not considered.

- *Model Predictive Control.* In model predictive control (MPC) or receding horizon control (RHC) a constrained optimal control problem is solved over a finite time horizon. An

open-loop control policy is implemented over the first time step and the optimization re-calculated over the shifted horizon. Re-optimization in an MPC type framework can be applied to any of the previous solution categories and the following references consider differing planning methods that use MPC. Keviczky et. al. compared MPC techniques for single axis control and found scheduled MPC provided an attractive alternative to full nonlinear control [52]. Kuwata et. al. investigated robust MPC using constraint tightening to design trajectories for an aerial vehicle operating in an environment with disturbances and constraints such as no-fly zones [56]. The computations demonstrated feasibility, but only for a simplified vehicle model operating in a 2D environment. A closed form solution of the nonlinear aircraft model using Taylor series expansions was investigated by Slegers et. al. but the inclusion of constraints using this method would prove difficult [84]. More recent work has shown a hierarchical structure can be implemented with an MPC guidance controller manoeuvring the aircraft between waypoints generated by a higher level path planner [90]. Further work in this area on path planning, specifically with obstacle constraints, is reviewed in Section 4.2.4.

The literature demonstrates that efficient algorithms exist that are well characterized for simpler sub-problems of the aircraft path planning problem. However they are frequently too computationally expensive to be used in real-time applications, and when they are tractable, are not optimal [33]. In this thesis some of these issues are addressed for a short horizon planner.

3.2 Convex Analysis and Optimization

Convex optimization is an important tool for solving constrained optimization problems of the form,

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{subject to} \quad & g_i(x) = 0, \quad \forall i \in \mathcal{E} \\ & h_i(x) \leq 0, \quad \forall i \in \mathcal{I} \end{aligned}$$

where x is a vector of decision variables, and the functions f, h_i and g_i are the cost, inequality and equality functions respectively. In general this problem is hard to solve as there could be many local optima, the feasible set might be empty and stopping criteria are often arbitrary [12]. If $f(\cdot)$ is convex, h_i are all convex and the g_i are affine, then the problem is convex and these issues become manageable in most cases of practical interest. A local optimum for the convex problem is a global optimum, feasibility can be established unambiguously and stopping criteria for algorithms are available through duality. In a number of problems that are convex however, there are no guarantees of convergence in polynomial time. Nesterov showed that for interior

point methods (so called 'path following' algorithms with a penalty function) polynomiality can be achieved if the penalty function satisfies certain conditions in its higher order differentials. The reader is referred to [66] for a detailed explanation.

As Chapter 4 will show, the optimization problem addressed in this work is non-convex. However solutions are obtained by applying methods commonly used (and developed for) convex problems. This section introduces the definitions and notation used in convex analysis. These and related concepts can be found in [78] and [12].

3.2.1 Convex Sets and Convex Functions

Definition 1. *(Convex Functions) Let C be convex with $C \subseteq \mathbb{R}^n$. A function $f : C \to \mathbb{R}$ is* convex *if*

$$f(\alpha x + (1 + \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y), \quad \forall (x, y) \in C, \quad \forall \alpha \in [0, 1].$$

Definition 2. *(Convex Sets) A set $C \subseteq \mathbb{R}^n$ is* convex *if*

$$\alpha x + (1 - \alpha)y \in C, \quad \forall (x, y) \in C, \quad \forall \alpha \in [0, 1],$$

i.e. the the line segment joining two points $x \in C$ and $y \in C$ is contained in C (Figure 3.1). The closure and interior of a convex set are convex. The empty set is considered by definition to be convex.

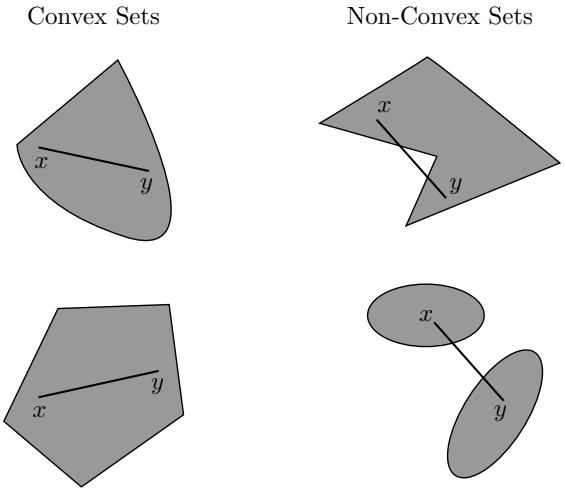


Figure 3.1: Examples of convex and non-convex sets.

Definition 3. *(Affine Functions) A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is* affine *if it has the form*

$$f(x) = Ax + b$$

If F is a matrix valued function, $F : \mathbb{R}^n \to \mathbb{R}^{p\times q}$, then F is affine *if it has the form*

$$F(x) = A_0 + x_1 A_1 + \ldots + x_n A_n$$

where $A_i \in \mathbb{R}^{p\times q}$.

Definition 4. *(Hyperplanes and Halfspaces) A* hyperplane *is a set of the form*

$$\{x \mid a^T x = b\}$$

where $a \in \mathbb{R}^n, a \neq 0$ and $b \in \mathbb{R}$. Hyperplanes are affine sets and are convex. A hyperplane divides $\mathbb{R}^n$ into two halfspaces*. A (closed) halfspace is a set of the form*

$$\{x \mid a^T x \leq b\}$$

where $a \neq 0$. Halfspaces are convex sets. See Figure 3.2.

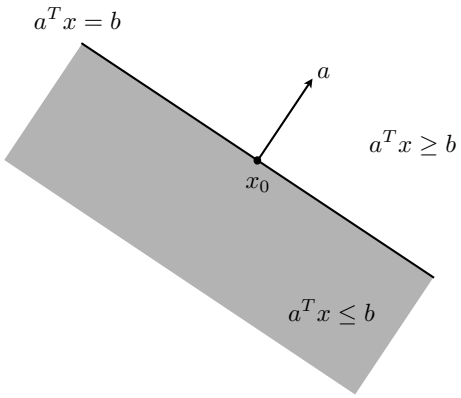


Figure 3.2: A hyperplane defined by $a^T x = b$ in $\mathbb{R}^2$ determines two halfspaces.

Definition 5. *(Polyhedra and Polytopes) A* polyhedron *is the intersection of a finite number of halfspaces*

$$\begin{aligned}\mathcal{P} &= \{x \mid a_i^T x \leq b_i, \quad \forall i \in \{1,\ldots,k\}\} \\ &= \{x \mid Ax \leq b\}.\end{aligned}$$

A bounded polyhedron is called a polytope*.*

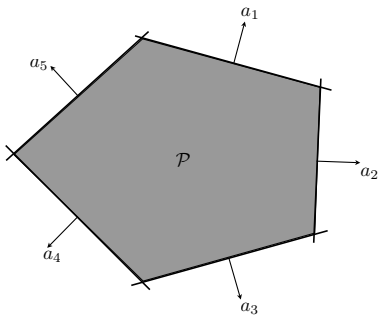


Figure 3.3: Polytope.

Definition 6. *(P-norm) The* p-norm *on* $\mathbb{R}^n$ *is*

$$\|x\|_p = \begin{cases} (\sum_i |x_i|^p)^{1/p} & \text{for } p \geq 1, \\ \max_i |x_i| & \text{for } p = \infty. \end{cases}$$

The p-norm is a convex set. See Figure 3.4.

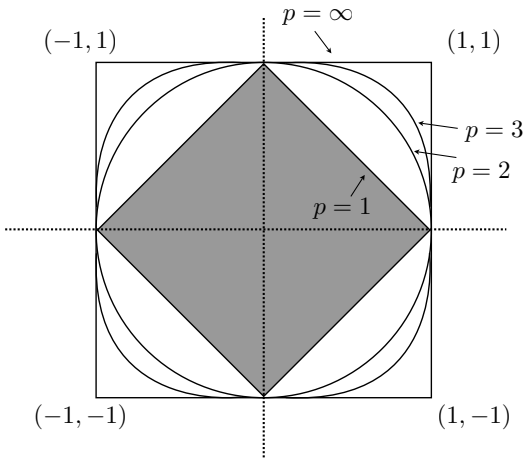


Figure 3.4: Norm balls in $\mathbb{R}^2$.

Definition 7. *(Ellipsoid) The* ellipsoid

$$\mathcal{E} = \left\{ x \;\middle|\; (x - x_c)^T A^{-1} (x - x_c) \leq 1 \right\}$$

with the centre $x_c \in \mathbb{R}^n$ *and* $A = A^T \succ 0$ *is the image of the Eucilidean ball* $\{x \mid \|x\|_2 \leq 1\}$ *under the affine mapping* $f(x) = A^{1/2}x + x_c$. *See Figure 3.5, where* λ_i *are the eigenvalues of* A.

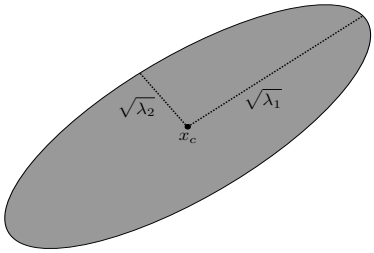


Figure 3.5: Ellipsoid in $\mathbb{R}^2$.

3.2.2 Polar, Support and Gauge Functions

Definition 8. *(Polar Sets) Given a set $C \subseteq \mathbb{R}^n$ with $0 \in C$, the* polar *of C is defined as:*

$$C^\circ \triangleq \{v \mid \langle v, x\rangle \leq 1,\ \forall x \in C\}.$$

See Figure 3.6.

Lemma 1. *(Properties of Polar Sets) The following properties hold [79]:*

i. If C is closed and convex with $0 \in \operatorname{int} C$, then C° is compact and convex with $0 \in \operatorname{int} C^\circ$.

ii. If the set C is polyhedral with

$$C = \{x \mid Hx \leq 1\},$$

then

$$C^\circ = \left\{y \;\middle|\; \exists z, y = H^T z, z \geq 0, z^T \mathbf{1} = 1\right\}.$$

iii. If C is a unit ball for some norm $\|\cdot\|$, i.e. $C = \{x \mid \|x\| \leq 1\}$, the polar set is $C^\circ = \{x \mid \|x\|_* \leq 1\}$ where $\|\cdot\|_*$ is the associated dual norm. In particular, for the p–norm unit ball $\mathcal{B}_p \triangleq \left\{x \;\middle|\; \|x\|_p \leq 1\right\}$,

$$(\mathcal{B}_p)^\circ = \mathcal{B}_q, \quad 1 < p < \infty,\ p^{-1} + q^{-1} = 1$$
$$(\mathcal{B}_1)^\circ = \mathcal{B}_\infty, \quad (\mathcal{B}_\infty)^\circ = \mathcal{B}_1.$$

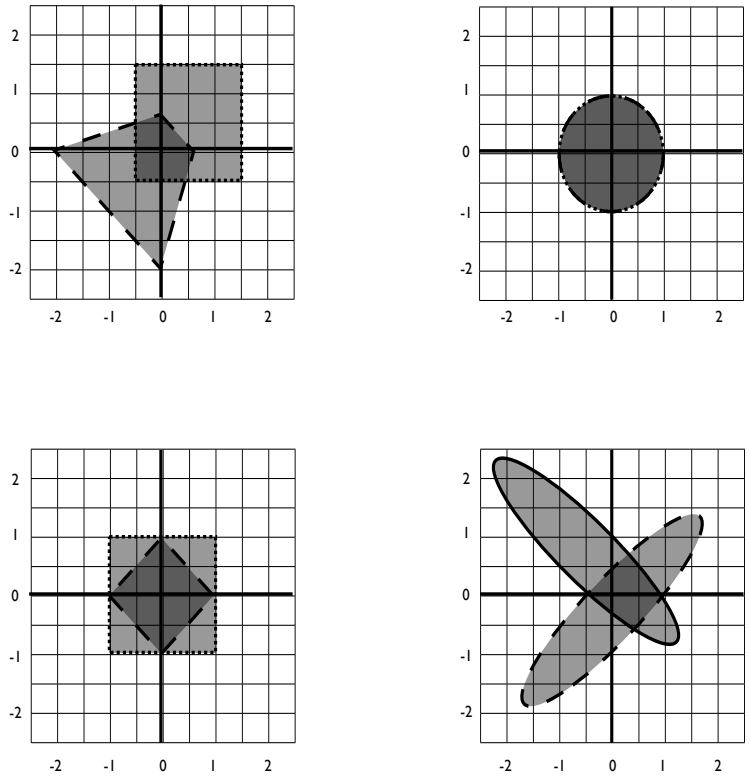


Figure 3.6: Examples of polar sets. The sets shown are polars of each other.

iv. If $C = \{x \mid \|Hx\| \leq 1\}$ *for some matrix* H*, then*

$$C^\circ = \left\{y \mid \exists z,\ y = H^T z,\ \|z\|_* \leq 1\right\}.$$

Definition 9. *(Support and Gauge Functions) Given a convex set* $C \subseteq \mathbb{R}^n$*, the* support function *of* C*,* $\sigma_C : \mathbb{R}^n \to \bar{\mathbb{R}}$ *(where* $\bar{\mathbb{R}}$ *is an upper bounded subset of* $\mathbb{R}$*) is defined as:*

$$\sigma_C(x) \triangleq \sup_{y \in C}(x^T y).$$

If $0 \in C$*, the* gauge function *of* C*,* $\gamma_C : \mathbb{R}^n \to \bar{\mathbb{R}}$ *is defined as:*

$$\gamma_C(x) \triangleq \inf\{\lambda \geq 0 \mid x \in \lambda C\}.$$

Remark 1. *The support function defines a plane tangent* $\left\{y \mid x^T y = \sigma_C(x)\right\}$ *to* C *with normal vector* x*, while the value of the gauge function* $\lambda = \gamma_C(x)$ *is the least amount by which* C *can be scaled while guaranteeing that* $x \in \lambda C$ *(Figure 3.7) [37].*

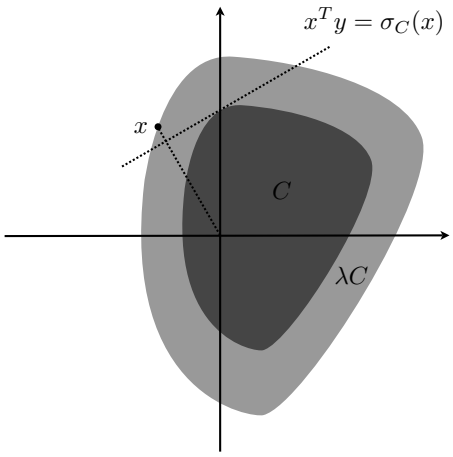


Figure 3.7: Support and gauge Functions.

The support and gauge functions of a set C have several properties that will be useful in subsequent sections; principal among these properties is their relation to one another through the polar set C°:

Lemma 2 (Properties of Support and Gauge Functions)**.** *If C is a closed and convex set with $0 \in C$, then the following properties hold:*

i. $\sigma_C(\cdot) = \gamma_{C^\circ}(\cdot)$ *and* $\gamma_C(\cdot) = \sigma_{C^\circ}(\cdot)$ *[79].*

ii. The Fenchel conjugate *of the gauge function [92],*

$$\gamma_C^*(y) \triangleq \sup_x \left\{ x^T y - \gamma_C(x) \right\}$$

is

$$\gamma_C^*(y) = \begin{cases} 0 & \text{if } y \in C^\circ \\ \infty & \text{otherwise,} \end{cases}$$

i.e. the conjugate of the gauge function is the indicator function of the associated polar set.

In general, it is possible to solve a large class of convex optimization problems and methods for doing so can be found in [12]. In problems which are non-convex, similar methods can be applied but there are no longer guarantees of global optimality, and the algorithm will typically find a local minimum. The nonlinear programming (NLP) optimization method adopted for this research is the well known primal-dual interior point method [95]. The open source IPOPT solver is chosen to perform optimizations and the reader is referred to [91] for a full description of its implementation.

3.3 Optimal Control

This section presents a summary of the methods available for solution of an optimal control problem in ordinary differential equations (ODE), since the trajectory generation PV avoidance problem of Chapter 4 is formulated as an optimal control problem. The ideas described in Sections 3.2 can then be used to solve the resulting NLP problem for the direct methods that are presented.

Consider a general optimal control problem of the form:

$$\begin{aligned}
&\min_{x(\cdot),u(\cdot)} \; F(x(T)) + \int_0^T \ell(x,u)dt \\
\text{subject to:} \quad & x(0) - x_0 = 0 && && \text{(fixed initial state)} \\
& \dot{x}(t) - f(x(t),u(t)) = 0, && \forall t \in [0,T] && \text{(ODE system model)} \\
& h(x(t),u(t)) \geq 0, && \forall t \in [0,T] && \text{(path constraints),}
\end{aligned}$$

where a cost function (consisting of a terminal cost $F(\cdot)$ and stage cost $\ell(\cdot,\cdot)$) is minimized from an initial state x_0 subject to ODE constraints representing the system dynamics and some inequality path constraints. In general, three approaches exist to solve optimal control problems of this type: *dynamic programming*, *indirect* and *direct* approaches. The reader is refered to [19] and [10] for a more extensive description.

In *dynamic programming* the principle of optimality is applied to compute recursively a feedback control for all times t and all x_0. This leads to the Hamilton-Jacobi-Bellman (HJB) equation in continuous time which is a partial differential equation (PDE) in the state space. The approach is restricted to small state dimensions due to the computational cost.

For *indirect methods* (including the calculus of variations and the Pontryagin Maximum Principle) the necessary conditions for optimality are used to derive a boundary value problem (BVP) which is then solved numerically (i.e. optimization is done first followed by discretization). The numerical solution of the BVP can be performed by direct methods (shooting or collocation). However it requires the explicit computation of adjoint and control equations (together these are the Euler-Lagrange equations) and reasonable initial guesses for the constrained arc and adjoint variables.

Direct methods convert the original infinite dimensional optimal control problem into a finite dimensional nonlinear programming problem, amenable to the use of state-of-the-art NLP solvers (therefore discretization is done first, followed by optimization). Direct methods have the advantage of being able to easily treat inequality path constraints [19]. These methods can be further categorized into single shooting, collocation and multiple shooting methods. The three methods differ in the way the state trajectory is handled. In single shooting, only the inputs $u(t)$ are discretized, whereas for multiple shooting and collocation both the inputs $u(t)$

and states $x(t)$ are discretized. In collocation, state continuity is implicit in the constraints via the choice of integration scheme. For multiple shooting an explicit constraint is imposed such that the state at the end of each discretization element is equal to the state decision variable at that node. The multiple shooting strategy is adopted for the PV avoidance trajectory generation problem and is explained in further detail in Section 4.2.

Remark 2. *For problems which are not necessarily formulated in the form given, e.g. trajectory generation for differentially flat systems, the resulting problem is an NLP (albeit in a different space) and can be solved by the methods described.*

3.4 System Dynamics: Aircraft Equations of Motion

In order to determine the future behaviour of an aircraft given its current state, a mathematical representation of the motion of an aircraft is required. This model is used in the trajectory generation component of the PV Avoidance System as described in Chapter 6. The modeling of both translational and rotational dynamics gives a six degree-of-freedom (6DOF) model consisting of 12 nonlinear differential equations. The construction of a 6DOF model is considered extensively in [87]. The state variables of this model are the position and altitude of the aircraft in a three-dimensional inertial frame, the velocity vector, the Euler angles determining the attitude of the aircraft and also the angular rates of pitch, yaw and roll motions about the centre of mass of the aircraft. The standard control variables for the 6DOF model are associated with elevator, aileron and rudder deflections that control the aerodynamic forces and moments on the aircraft. Tangential acceleration is controlled with the throttle setting.

A more simplified three degree-of-freedom (3DOF) model can be produced if the rotational dynamics of the aircraft are ignored [63]. It consists of six nonlinear differential equations describing the motion of an aircraft as a point mass in three dimensional space. The state variables of the point mass model represent the position, altitude and speed of the aircraft as well as the the flight path angle and the heading angle. These angles determine the direction of the velocity vector. The normal acceleration of the aircraft is controlled with the load factor (the ratio of lift to weight) or the angle of attack, and the tangential acceleration with the throttle setting.

The 3DOF and 6DOF equations of motion are generic to all aircraft. However there will be a different set of constraints to the state and control variables set by the pilot or aircraft type. Each aircraft type has a unique set of geometric parameters that define the lift, drag, moment and stability coefficients. The thrust is determined by the engine type. The forces and moments acting on the model are functions of the air density and Mach number of the aircraft. Hence, the properties of the atmosphere must also be taken into account in the aircraft models.

The use of the 3DOF model may lead to unrealistic prediction of aircraft motion especially

in situations where the aircraft moves excessively, as there are no limits on angular velocities. However numerical solutions of the 3DOF model requires less computational effort than the 6DOF model. In this work a 3DOF model with rate constraints is used for the following reasons:

- The 3DOF requires less computational effort. In the NLP formulation for trajectory generation, the computation of Jacobian matrices is required in the optimization. This would be substantially slower for the 6DOF system and prohibit its use in a real-time system with the current computational resources available.
- There are also large uncertainties in 6DOF modeling related to the estimation of aerodynamic data [43].
- The lower order 3DOF point mass model is used to generate avoidance trajectories flyable via coordinated (i.e. no sideslip) turns. Conservative rate constraints are included in the model (through the control strategy) to ensure it manoeuvres within the envelope of a real aircraft. Therefore the 3DOF model allows a subset of the manoeuvres available to the 6DOF model. This approach has previously been shown to be successful in generating terrain following trajectories for a 3DOF model when compared to a 6DOF model for a helicopter [16]. Although some flyable trajectories will be prohibited by the lower fidelity model, this is acceptable as long as the generated trajectories are flyable by the aircraft.
- The time-scale for PV avoidance trajectory generation is in the order of tens of seconds. Therefore transient motions that occur over short time scales which might be captured by the 6DOF model are not necessary for the types of trajectory generated by the PV avoidance system.

3.4.1 3DOF Dynamics

For the purposes of trajectory generation the aircraft dynamics are modeled using a nonlinear 3DOF model (Figure 3.8) which excludes side slip. The states of the model are the horizontal position x_c and y_c and altitude z_c of the aircraft, the true airspeed v, the flight path angle γ, and the heading angle χ. The control inputs to the model are the engine thrust T, the angle of attack α and the roll angle μ. The state vector is defined as $x \triangleq (x_c, y_c, z_c, \gamma, \chi, v)$ and the input vector as $u \triangleq (\alpha, \mu, T)$. In the model described here, thrust is a direct input. However a more realistic approach is to use a fractional throttle as an input and derive a suitable throttle to thrust relationship $T(t_p, z, v)$ where $t_p \in [0, 1]$ is the throttle setting. A description of how this is achieved for the example aircraft in simulations is given in Chapter 7.

Using simple Newtonian dynamics, and assuming there are no disturbances due to wind,

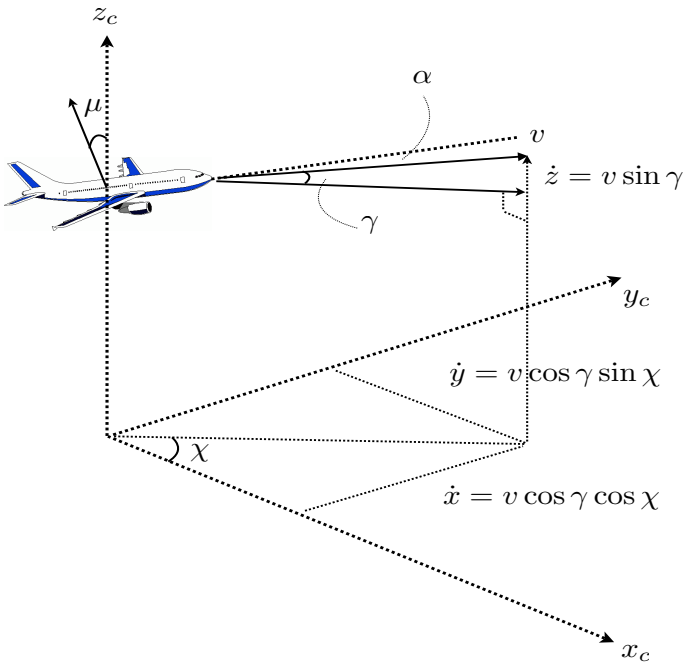


Figure 3.8: Coordinate system for the 3DOF aircraft model.

the equations of motion are:

$$\dot{x_c} = v \cos\gamma \cos\chi \tag{3.1a}$$

$$\dot{y_c} = v \cos\gamma \sin\chi \tag{3.1b}$$

$$\dot{z_c} = v \sin\gamma \tag{3.1c}$$

$$\dot{\gamma} = \frac{1}{mV}[(L + T\sin\alpha)\cos\mu - mg\cos\gamma] \tag{3.1d}$$

$$\dot{\chi} = \frac{1}{mv\cos\gamma}[(L + T\sin\alpha)\sin\mu] \tag{3.1e}$$

$$\dot{v} = \frac{1}{m}[(T\cos\alpha - D) - mg\sin\gamma]. \tag{3.1f}$$

The aerodynamic forces of lift and drag are denoted L and D respectively, and the mass of the aircraft (assumed constant over the duration of the trajectory) by m. The aircraft moment of inertia and rotational dynamics are not included in (3.1) since it is a point mass model. For the purposes of this model the aircraft can be controlled directly via changes to the angle of attack and the roll angle. Sideslip is assumed to be negligible and is therefore omitted. Due to the relatively short distances and time intervals considered in the PV avoidance problem, the earth is regarded as ideally flat and non-rotating.

Remark 3. *If required, wind can be included in* (3.1) *by adding the relative motion between*

the body of air and the ground into the $\dot{x_c}, \dot{y_c}$, and $\dot{z_c}$ state equations:

$$\dot{x_c} = v\cos\gamma\cos\chi + w_x$$
$$\dot{y_c} = v\cos\gamma\sin\chi + w_y$$
$$\dot{z_c} = v\sin\gamma + w_z$$

where w_x, w_y and w_z are the components of wind in the x, y and z directions respectively.

The aircraft model is constrained to incorporate the performance and structural limitations that are present on a real aircraft. These constraints are not intrinsic to the model, but are applied as part of the control strategy. Any avoidance trajectories that are generated can also include additional constraints for the safety of passengers onboard. These input constraints are modelled as lower and upper bounds:

$$\alpha \in [\alpha_{min}, \alpha_{max}] \tag{3.2a}$$
$$\mu \in [\mu_{min}, \mu_{max}] \tag{3.2b}$$
$$T \in [0, T_{max}]. \tag{3.2c}$$

The limits on the angle of attack are chosen to remain within the vehicle stall limits, the roll is bounded by a value suitable to maintain structural integrity and the time varying maximum thrust is determined by the engine specifications at a given altitude and Mach number. The state constraints are likewise modelled as

$$z_c \in [z_{min}, z_{max}] \tag{3.3a}$$
$$v \in [v_{min}, v_{max}] \tag{3.3b}$$
$$\gamma \in [\gamma_{min}, \gamma_{max}]. \tag{3.3c}$$

The maximum altitude achievable is set in (3.3a) as the performance ceiling of the aircraft (although a real implementation of the PV Avoidance System would only be operational at lower altitudes where PVs are present). The minimum altitude is the local altitude above mean sea-level. For the proof-of-concept system the ground is assumed to be flat at this level. The velocity bounds can be a (time-varying) function of the aircraft state, with the stall velocity and maximum operating velocity (v_{MO}) as the bounding limits. Alternatively, the most conservative values for the given aircraft can be used as fixed constraints. The flight path angle is bounded by a suitable value for both passenger safety and aircraft structural integrity. Rate constraints on the inputs ($\dot{\alpha}, \dot{\mu}, \dot{T}$) and also on the flight path angle ($\dot{\gamma}$) are incorporated into the model through the control strategy (described in Section 4.2.2). These prevent unrealistic jumps in states of the actual aircraft in angle of attack, roll angle or thrust, which would otherwise be

allowed by the 3DOF model where the rates are unconstrained.

Chapter 4

Trajectory Generation Scheme

Having introduced the mathematical conventions for convex analysis and established a model of the system, the trajectory generation problem of the PV Avoidance System (see Figure 1.3) can be stated in a formal manner. Initially the modeling of prohibited volume (state exclusion) regions is described leading to a continuous-time optimal control problem. Next, a direct multiple shooting strategy for generating avoidance trajectories is proposed, with a novel method for remodelling non-differentiable constraint sets using dualization. The motivation for doing so is to allow the use of a standard gradient based optimization method. Finally, terminal invariance is addressed as a means of ensuring an aircraft is not on a direct course to penetrate a PV at the end of the avoidance trajectory. The terminal invariance condition is combined with a shift-intialization warm-start procedure in order to initialize the optimization close to a feasible point.

4.1 Modelling PV Regions

Assume that there exist a collection of regions in space (*prohibited volumes*) from which the vehicle trajectory is to remain excluded. Therefore, a collection of M convex sets $\mathcal{C}_j \subseteq \mathbb{R}^3$ are defined with associated reference points r_j, such that the aircraft trajectory is to be excluded from each of the sets $\mathcal{C}_j \oplus r_j$, where the $\oplus$ operator denotes translation of the set $\mathcal{C}_j$ by r_j. Each of the sets $\mathcal{C}_j$ is assumed to be closed and convex with non-empty interior. Formally, the following set exclusion condition is imposed:

$$p(x) \notin \bigcup_{j=1}^{M} \left(\operatorname{int} \mathcal{C}_j \oplus r_j\right), \tag{4.1}$$

where $\operatorname{int} \mathcal{C}_j$ denotes the interior of the set $\mathcal{C}_j$, and $p(\cdot)$ is a function extracting the position vector from the full state vector ($x \in \mathbb{R}^6$, $p(x) \in \mathbb{R}^3$). For notational convenience $p(x)$ will be

simplified to p herein. The interior of the set is used since the analysis of Section 4.2.4 provides a certificate that ensures the aircraft is outside the interior of the set, but allows for points on the boundary. It is possible to model a very wide variety of constraint sets in this way, including non-convex regions of space, due to the presence of the union operation. The constraint (4.1) dictates that the system state remain *outside* of a (collection of) convex set(s), and is therefore non-convex. In many cases the set C_j can be defined in terms of inequality constraints,

$$C_j \triangleq \{x \mid h_i(x) \leq 0, \quad \forall i \in \{1, \dots l\}\}. \tag{4.2}$$

Initially the PV inequality constraints are assumed to be differentiable. Later in the work (Section 4.2.4) this assumption is relaxed and a dualization strategy for dealing with non-differentiable state exclusion regions is described.

Example 1. *(Ellipsoid and Infinite-height Cylinder PVs) Ellipsoidal and infinite-height cylinder state exclusion regions (PVs) can be written as*

$$C = \left\{p \;\middle|\; (p-r)^T V (p-r) - 1 \leq 0\right\},$$

where r defines the centre of the PV and $V (\succeq 0) \in \mathbb{R}^{3\times 3}$ determines the type, size and scaling of the PV in Cartesian space. If V is a diagonal matrix with elements $(1/a^2, 1/b^2, 1/c^2)$ along the leading diagonal, then

i. For the ellipsoidal case $a > 0$, $b > 0$ and $c > 0$ represent the semi-lengths in the x_c, y_c and z_c directions respectively.

ii. For the infinite-height cylinder case $a > 0$, $b > 0$ represent the semi-lengths in the x_c and y_c directions. The third dimension is unconstrained, i.e. $1/c^2 = 0$.

The constraint to be outside $\operatorname{int} C$ *is implemented as*

$$(p-r)^T V (p-r) - 1 \geq 0.$$

Example 2. *(Polytopic PVs) Polytopic state exclusion regions can be written directly from the definition of a polytope (Definiton 5):*

$$C = \{p \mid A(p-r) - b \leq 0\},$$

where r defines the centre of the PV and $A \in \mathbb{R}^{l\times 3}, b \in \mathbb{R}^l$ determine the type, size and scaling of the polytopic PV in Cartesian space. The dimension l determines the number of faces for the polytope. Since only one halfspace constraint needs to be satisfied for a point to be outside

$\operatorname{int} C$, *the constraint can be written as*

$$\max_i \{a_i^T (p - r) - b_i\} \geq 0, \qquad \forall i \in \{1 \dots l\}.$$

4.2 Optimal Control Problem and Solution Strategy

The aim is to formulate a problem which can be solved to produce an aircraft trajectory that avoids all the prohibited volumes in the local area of the aircraft. In order to do so, the continuous-time problem is outlined, followed by a description of the standard direct multiple shooting method, applied to the aircraft specific problem.

The continuous-time infinite dimensional trajectory generation problem can be stated as:

$$\min_{x(\cdot),u(\cdot)} \; J(x(\cdot), u(\cdot)))$$

subject to:

$$\begin{aligned}
&\dot{x}(t) = f(x(t), u(t)), && \forall t \in [0, t_f) \\
&x(t) \in X, \quad u(t) \in U, \quad \dot{x}(t) \in X_{rate}, \quad \dot{u}(t) \in U_{rate}, && \forall t \in [0, t_f) \\
&p(t) \notin \bigcup_{j=1}^{M} \left(\operatorname{int} \mathcal{C}_j \oplus r_j \right), && \forall t \in [0, t_f).
\end{aligned}$$

where $\dot{x}(t) = f(x(t), u(t))$ represents the system dynamics from (3.1) and the sets $X \in \mathbb{R}^6$, $U \in \mathbb{R}^3$, $X_{rate} \in \mathbb{R}^6$ and $U_{rate} \in \mathbb{R}^3$ are the manoeuvring bounds of the aircraft. The time horizon over which the problem is solved is from $t = 0$ to $t = t_f$. The objective function J is assumed to be twice continuously differentiable and is selected to induce a desirable trajectory for the aircraft (e.g. minimum control action or minimum deviation from original aircraft path). Figure 4.1 shows an avoidance trajectory with two PVs in the original flight path of the aircraft.

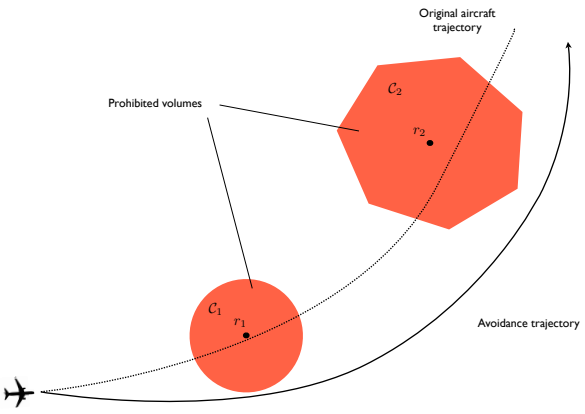


Figure 4.1: Avoidance trajectory.

There are a number of strategies for solving nonlinear optimal control problems depending on the class of problem (cf. Section 3.3) [32]. Most practical problems are solved using nonlinear programming methods based on some discretization of the original continuous control problem. The controls u and states x can be discretized in time to form a finite-dimensional approximation to the original infinite dimensional, continuous time problem. This approximated problem can then be solved using one of many numerical optimization methods available.

Multiple shooting is one such method used frequently in the chemical engineering industry, where large scale control problems often arise [20]. It was first introduced by Plitt and Bock [69], and its use in aerospace applications suggested by Betts [9]. In multiple shooting the control horizon is divided into N intervals, with states variables introduced at the end of each interval as additional decision parameters. This is generally preferable to single shooting (e.g. optimization over the control inputs only, without modelling intermediate states) due to improved convergence and enhanced robustness to poor initial data, discretization or round-off errors [69]. Figure 4.2 shows a graphical interpretation of the multiple shooting strategy. The inputs are held constant over each interval and the states are decoupled.

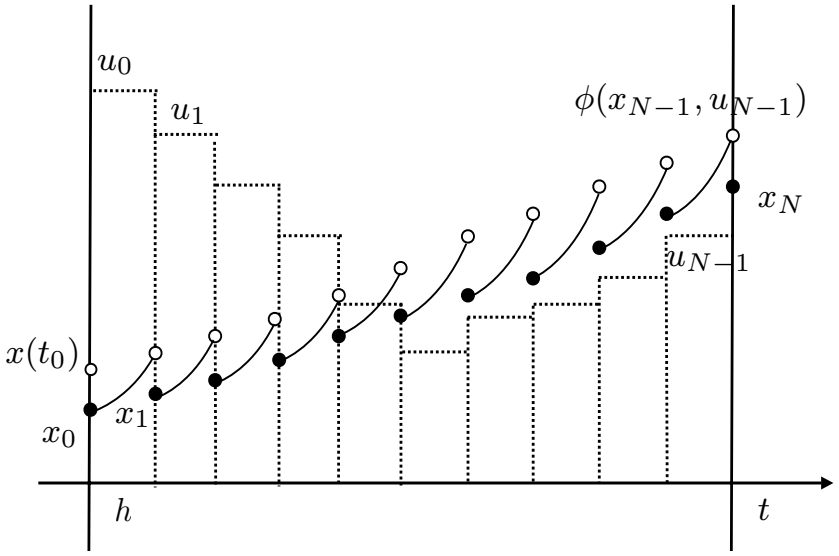


Figure 4.2: Multiple shooting

Sections 4.2.1 and 4.2.3 describe how the continuous time aircraft trajectory generation problem is solved in the multiple shooting framework. It is important to note that multiple shooting itself is a well established strategy. The research novelty lies in the representation of the prohibited volumes (particularly polytopic PVs), and the subsequent construction of a positively invariant terminal set which induces the optimization into converging on a favourable local optimum.

4.2.1 Discretizing the Optimal Control Problem

In order to find a constraint-admissible trajectory for the aircraft trajectory generation problem, the aircraft dynamics (3.1) are incorporated into a single expression

$$\dot{x} = f(x, u), \tag{4.3}$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ are the combined state and input vectors respectively. Since the interior-point optimization method requires a finite-dimensional decision space, the dynamics are then discretized by integration over fixed intervals h using an ordinary differential equation (ODE) solver. The state at the end of a multiple shooting segment is

$$\phi(x_i, u_i) \triangleq x_i + \int_t^{t+h} f(x(\tau), u(\tau))d\tau. \tag{4.4}$$

Alternatively, a simpler integration method (e.g. a first order Euler method) can be adopted if the time step is small enough to ensure an acceptable level of accuracy (see Section 5.1). In either case, the function $\phi : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ represents the dynamics of the discretized system, with $x_i \in \mathbb{R}^n$ and $u_i \in \mathbb{R}^m$, representing the system states and inputs at discrete time instants, treated as decision variables. This method produces discrete state points at each of the multiple shooting nodes. Jumps in states and inputs which cannot be realised by the actual aircraft are prohibited by the rate constraints of the following section.

4.2.2 State Continuity and Rate Constraints

Continuity of the system state trajectory is ensured via introduction of equality constraints in the form

$$x_0 - x(t_0) = 0 \tag{4.5a}$$

$$\phi(x_i, u_i) - x_{i+1} = 0, \quad \forall i \in \{0, \ldots, N-1\}, \tag{4.5b}$$

where the initial state $x(t_0)$ is assumed to be known. These state continuity equality constraints can be bundled into a single expression $g_{cont}(\mathbf{x}) = 0$. For this system, a constraint-admissible state and input sequence is determined to minimize

$$J_N(\mathbf{x}, \mathbf{u}) = F(x_N) + \sum_{i=0}^{N-1} \ell_i(x_i, u_i) \tag{4.6}$$

where $\mathbf{u} \triangleq (u_0, \ldots, u_{N-1})$ and $\mathbf{x} \triangleq (x_0, \ldots, x_N)$. The stage costs $\ell_i : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_+$ and terminal cost $F : \mathbb{R}^n \to \mathbb{R}_+$ are assumed to be twice continuously differentiable. Additional

rate constraints on the states and inputs can be posed in the form

$$\dot{u}_{min} \leq \frac{u_0 - u(t_0)}{h} \leq \dot{u}_{max} \tag{4.7a}$$

$$\dot{u}_{min} \leq \frac{u_{i+1} - u_i}{h} \leq \dot{u}_{max}, \quad \forall i \in \{0, \dots, N-2\} \tag{4.7b}$$

$$\dot{x}_{min} \leq \frac{x_{i+1} - x_i}{h} \leq \dot{x}_{max}, \quad \forall i \in \{0, \dots, N-1\} \tag{4.7c}$$

where the condition (4.7a) ensures that the first input of the control sequence u_0 can feasibly be reached from the initial condition $u(t_0)$. In practical terms this ensures that the generated control sequence starts close to the pilot's current selection. The rate constraints can be bundled into the single expression $h_{rate}(\mathbf{x}, \mathbf{u}) \leq 0$.

4.2.3 Problem Formulation

The optimal control problem, denoted $\mathcal{P}(x(t_0), u(t_0))$, can be written

$$\begin{aligned} & \min_{\mathbf{x},\mathbf{u}} \; J_N(\mathbf{x}, \mathbf{u}) \\ \text{subject to:} \quad & g_{cont}(\mathbf{x}) = 0 \\ & h_{rate}(\mathbf{x}, \mathbf{u}) \leq 0 \\ & \mathbf{x} \in X, \quad \mathbf{u} \in U \\ & p_i \notin \bigcup_{j=1}^{M} \left(\operatorname{int} \mathcal{C}_j \oplus r_j\right), \qquad \forall i \in \{0, \dots N\}, \forall j \in \{1, \dots, M\}. \end{aligned} \tag{OCP}$$

where $g_{cont}(\mathbf{z}) = 0$ and $h_{rate}(\mathbf{z}) \leq 0$ encompass the sate continuity (4.5) and rate constraints (4.7) respectively. Note that the problem formulation still allows for states between the node points to be inside the PV. Therefore the multiple shooting time interval should be small enough such that linear interpolation between nodes will not result in PV penetration. If the sets $\mathcal{C}_j$ can be characterized in the form (4.2), then the state exclusion constraint can be rewritten to obtain

$$\begin{aligned} & \min_{\mathbf{x},\mathbf{u}} \; J_N(\mathbf{x}, \mathbf{u}) \\ \text{subject to:} \quad & g_{cont}(\mathbf{x}) = 0 \\ & h_{rate}(\mathbf{x}, \mathbf{u}) \leq 0 \\ & \mathbf{x} \in X, \quad \mathbf{u} \in U \\ & h_{PV}(\mathbf{x}) \leq 0, \end{aligned}$$

where the differentiable PV constraints are written as an inequality constraint h_{PV}. In order to solve the direct multiple shooting problem via the primal-dual interior-point method. The problem elements in the following paragraph are defined.

The decision variable over which the optimization takes place includes both the inputs and

states interleaved,

$$\mathbf{z} \triangleq (x_0; u_0; x_1; u_1; \ldots; u_{N-1}; x_N). \tag{4.8}$$

The cost function for the problem is defined as

$$J_N(\mathbf{x}, \mathbf{u}) = \|(x_N - x_N^r)\|_P^2 + \sum_{i=0}^{N-1} \left(\|x_i - x_i^r\|_Q^2 + \|u_i - u_i^r\|_R^2 \right) \tag{4.9}$$

where $P \in \mathbb{R}^{6\times6}$, $Q \in \mathbb{R}^{6\times6}$ and $R \in \mathbb{R}^{3\times3}$ are positive semi-definite, symmetric matrices adjusted to penalize the states or inputs as required. The calculation is initialized with a reference trajectory, $\{x_i^r\}$ and $\{u_i^r\}$. The reference trajectory is generated by time integration of the model from the initial state x_0, with the inputs fixed as $u(t_0)$ over the time horizon (i.e. the trajectory of the aircraft with the current pilot's input held fixed). Deviations from this trajectory on some or all of the states and inputs can then be penalized within (4.9).
The gradient of the cost function is

$$\nabla J(\mathbf{z}) = \begin{bmatrix} (x_0 - x_0^r)'Q \\ (u_0 - u_0^r)'R \\ \vdots \\ (x_{N-1} - x_{N-1}^r)'Q \\ (u_{N-1} - u_{N-1}^r)'R \\ (x_N - x_N^r)'P \end{bmatrix}. \tag{4.10}$$

The vector of constraints is

$$c(\mathbf{z}) = \left[\begin{array}{c} x(t_0) - x_0 \\ \phi_0 - x_1 \\ \vdots \\ \phi_{N-1} - x_N \\ \hline h_{rate}(\mathbf{z}) \\ h_{PV}(\mathbf{z}) \end{array}\right]. \tag{4.11}$$

where h_{rate} encompasses the equality constraints of (4.7) and $\phi_i = \phi(x_i, u_i)$. The Jacobian of

the constraint function is

$$\nabla c(\mathbf{z}) = \left[\begin{array}{ccccccccc} -I & & & & & & & & \\ \frac{\partial \phi_0}{\partial x_0} & \frac{\partial \phi_0}{\partial u_0} & -I & & & & & & \\ & & \frac{\partial \phi_1}{\partial x_1} & \frac{\partial \phi_1}{\partial u_1} & -I & & & & \\ & & & & \ddots & \ddots & & & \\ & & & & & & -I & & \\ & & & & & & \frac{\partial \phi_{N-1}}{\partial x_{N-1}} & \frac{\partial \phi_{N-1}}{\partial u_{N-1}} & -I \\ \hline \multicolumn{9}{c}{\nabla h_{rate}(\mathbf{z})} \\ \multicolumn{9}{c}{\nabla h_{PV}(\mathbf{z})} \end{array}\right], \tag{4.12}$$

where $\frac{\partial \phi_i}{\partial x_i}$ is the sensitivity at the end of the interval $[0, \Delta T)$ to the initial condition x_i. Similarly, $\frac{\partial \phi_i}{\partial u_i}$ is the sensitivity at the end of the interval to the control input u_i. Note that the matrix blocks $\nabla h_{rate}(\mathbf{z})$ and $\nabla h_{PV}(\mathbf{z})$ also have block diagonal structures. Discretization of the system dynamics into intervals, for which the initial states x_i are treated as decision variables, leads to a sparse banded Jacobian structure which is advantageous during matrix factorization within the optimization. The problem structure is also amenable to parallel computation since integration over multiple shooting intervals are completely decoupled which is a consequence of the use of the state as a decision variable.

Remark 4. *The multiple shooting problem is presented with nonlinear system dynamics (3.1). From a linearization of the 3DOF model the maximum errors obtainable over a typical trajectory are larger than acceptable. Therefore, the nonlinear model (instead of a linearization of the 3DOF model) is used throughout the work presented in the thesis. Results of an analysis of the linearized model are presented in Appendix A.*

4.2.4 Non-differentiable State Exclusion Regions

In Section 4.1 the prohibited volumes are assumed to be differentiable. This assumption is now relaxed and, using results from Section 3.2, a more general problem formulation is derived, which allows in particular the use of polytopic PVs.

A difficulty in solving OCP is the presence of the non-convex and non-differentiable state exclusion regions. State exclusion conditions (obstacle constraints) of this type often appear in path planning problems. Earlier work considered the obstacle avoidance problem in a continuous time framework using dynamic optimization [88, 83, 39]. The set of states that can be steered to a target set, whilst satisfying bound constraints and avoiding obstacles, is a level set of the value function of the dynamic optimization problem, obtained by solving a Hamilton-Jacobi-Bellman equation. Another approach is to include an additional potential function for avoiding obstacles in the cost function, thus converting the collision avoidance problem into

an unconstrained optimization allowing the use of gradient based solvers [53]. However the construction of potential functions in a general framework is difficult. Polytopic constraints specifically, have been dealt with via the use of additional binary variables in mixed integer optimization. Each halfspace representing a polytope face is a linear constraint, and satisfaction of at least one constraint ensures avoidance [77, 76, 72, 11]. A new approach to modeling polytopic constraints is developed in this work, that avoids the use of integer variables by preserving continuity and is amenable for use with a gradient based optimization strategy.

The non-differentiability of a convex set is not a well defined concept and it is explained in context here. In order to use a gradient based optimization such as the interior-point algorithm, the solver requires the Jacobian of the obstacle inequality constraints ∇h_{PV} as shown in (4.12). The problem arises when for a convex function $f(x) : \mathbb{R}^n \to \mathbb{R}$, its gradient $\nabla f(x)$ is not a continuous function of x.

Definition 10. *(Subgradients and Subdifferentials [8]) Given a convex function $f(x) : \mathbb{R}^n \to \mathbb{R}$, then $d \in \mathbb{R}^n$ is a* subgradient *of f at point $x \in \mathbb{R}^n$ if*

$$f(z) \geq f(x) + (z - x)^T d, \qquad \forall z \in \mathbb{R}^n \tag{4.13}$$

The set of all subgradients of a convex function at $x \in \mathbb{R}^n$ is called the subdifferential *of f at x, and is denoted $\partial f(x)$ (itself a nonempty, convex and compact set). See the examples in Figure 4.3.*

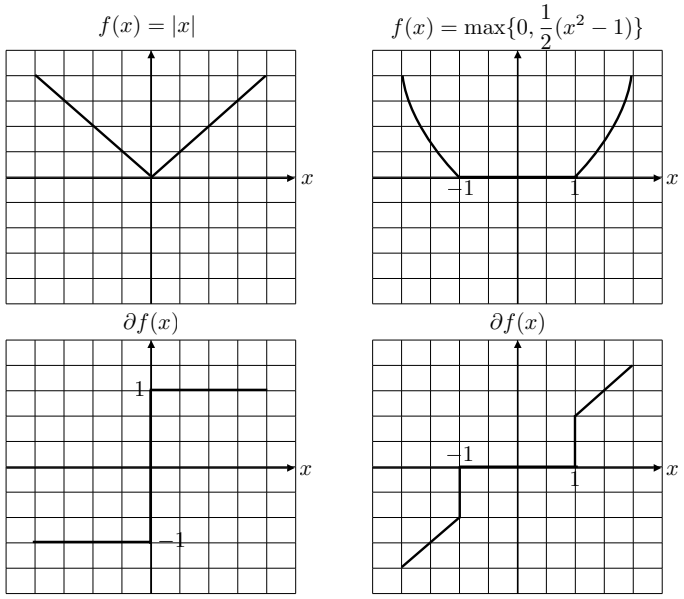


Figure 4.3: The subdifferential of some scalar convex functions as a function of the argument x. Based on [8].

For a convex set C which can be locally defined by a convex function f, there is no unique

subgradient $\nabla f(x)$ at its geometric corner x. Therefore it is not possible to define a unique search direction as required in the optimization.

The issue of non-unique subgradients is overcome by dualization as described here. Since the gauge function is the minimum scaling required for a convex set $\mathcal{C}_j$ centered at r_j to include the point $p_i - r_j$, the set exclusion condition (4.1) can be remodeled as

$$\gamma_{C_j}(p_i - r_j) \geq 1, \quad \forall i \in \{0, \ldots, N\}, \forall j \in \{1, \ldots, M\}.$$

Using Lemma 2.i, this can be rewritten as

$$\sigma_{C_j{}^\circ}(p_i - r_j) \geq 1.$$

From Definition 9 for the support function, this in turn can be written as

$$\sup_{y \in C_j{}^\circ} ((p_i - r_j)^T y) \geq 1$$

which is equivalent to

$$p_i - r_j \in \left\{ \tilde{p} \;\middle|\; \exists y,\ \tilde{p}^T y \geq 1,\ y \in C_j^\circ \right\}, \quad \forall i \in \{0, \ldots, N\}, \forall j \in \{1, \ldots, M\}. \tag{4.14}$$

Remark 5. *Note that any vector y satisfying the conditions of* (4.14) *provides a certificate verifying the condition $p_i - r_j \notin \operatorname{int} \mathcal{C}_j$, since the hyperplane $\left\{ p \;\middle|\; p^T y = 1 \right\}$ contains the point p_i and is disjoint from the set $(\operatorname{int} \mathcal{C}_j \oplus r_j)$.*

The remodeled optimal control problem can now be written as

$$\begin{aligned}
&\min_{\mathbf{x},\mathbf{u},y_{i,j}} && J_N(\mathbf{x}, \mathbf{u}) \\
&\text{subject to:} && g_{cont}(\mathbf{x}) = 0 \\
&&& h_{rate}(\mathbf{x}, \mathbf{u}) \leq 0 \\
&&& \mathbf{x} \in X, \quad \mathbf{u} \in U \\
&&& (p_i - r_j)^T y_{i,j} \geq 1 \\
&&& y_{i,j} \in C_j^\circ
\end{aligned} \Bigg\} \quad \forall i \in \{0, \ldots, N\}, \forall j \in \{1, \ldots, M\}, \tag{OCP2}$$

recalling that $g_{cont}(\mathbf{x}) = 0$ and $h_{rate}(\mathbf{x}, \mathbf{u}) \leq 0$ are the state continuity constraints (4.5) and rate constraints (4.7) respectively, and $y_{i,j}$ are auxiliary decision variables in the problem. The

set of feasible initial conditions for the problem $\mathcal{P}(x(t_0), u(t_0))$ of OCP2 is

$$\mathcal{X}_{\mathcal{N}} \triangleq \{(x,u) \in \mathbb{R}^n \times \mathbb{R}^m \mid \text{OCP2 is feasible with } x = x(t_0),\ u = u(t_0) \text{ and horizon } N\}. \tag{4.15}$$

Geometric Interpretation and Implementation of Problem OCP

The geometric interpretation of the remodeling of problem OCP into the form OCP2 via introduction of polar sets and dual variables is straightforward. The polar sets C_j° are 'large' in the directions for which the original constraint sets C_j are 'small', and vice-versa. If a multiplier $y_{i,j} \in C_j^\circ$ can be found such that the constraint $(p_i - r_j)^T y_{i,j} \geq 1$ in OCP2 is satisfied, then that multiplier must be aligned sufficiently well with the relative aircraft position vector $(p_i - r_j)$ to ensure that the aircraft position at time i lies outside of the original constraint set $C_j \oplus r_j$.

This remodeling ensures that the problem constraints (and in particular polytopic constraints) are representable in terms of continuously differentiable functions, rather than requiring the use of integer variables. This has considerable practical advantages for numerical implementation, since it allows the problem OCP to be solved using an existing gradient-based solver suitable for continuous optimization problems. This is achieved at the cost of increased problem dimensionality due to the appearance of the new dual variables $y_{i,j}$ at each time step. The non-convexity of the control problem is unchanged by this remodeling.

For the most common constraint types in practical applications (e.g. polytopes, scaled norm balls for different norms), the polar sets C_j° are not inherently more complex to represent than the original sets C_j. From Lemma 1, polar sets of polytopes are themselves polytopes (with face normals exchanged with vertices), and scaled balls in any norm are replaced with scaled balls in the associated dual norm. The original non-differentiable polytopic constraint

$$\max_i \{a_i^T (p - r) - b_i\} \geq 0 \qquad \forall i \in \{1 \dots l\}.$$

is replaced with multiple independent differentiable constraints by the dualization. In practice this approach is only required for state exclusion constraints with geometric corners, since norm ball constraints are differentiable in all cases except the infinity norm, which is dealt with in Remark 6. In Section 4.3 however, extension of the problem to the infinite horizon is addressed by employing the same dualization methods. In this case dualization is required for both the norm-ball and polytopic constraints.

Remark 6. *Note that the constraints of the polar set C° (Lemma 1.iv) are differentiable for the 2-norm case. The infinity norm case $C = \{x \mid \|Hx\|_\infty \leq 1\}$ requires an extra step to preserve*

continuity. Since the norm $\|\cdot\|_\infty$ *has the dual norm* $\|\cdot\|_1$*, then*

$$C^\circ = \left\{ x \;\middle|\; \exists z,\ x = H^T z,\ \|z\|_1 \leq 1 \right\}. \tag{4.16}$$

The variable z *can be further partitioned into positive and negative components respectively to give*

$$C^\circ = \left\{ x \;\middle|\; \exists (z^+, z^-) \geq 0,\ x = H^T(z^+ - z^-),\ \mathbf{1}^T(z^+ + z^-) \leq 1 \right\}, \tag{4.17}$$

where the constraints are now differentiable at the cost of additional variables (z^+, z^-)*.*

4.2.5 The Relationship Between Mixed Integer Linear Programming and Dualization methods for Disjunctive Programming

As stated in the previous section mixed integer optimization has previously been used for vehicle path planning applications where polytopic obstacles are present. In this section the relationship between the dualization method and the MILP form of the obstacle avoidance problem is derived, and the dualization method is shown to be a relaxation of the MILP formulation.

A general disjunctive program can be defined as

$$\begin{aligned} &\min_x f(x) \\ \text{subject to} \quad &\bigwedge_{i=1,\ldots,p} \Big(\bigvee_{j=1,\ldots,q_i} c_{i,j}(x) \leq 0 \Big), \end{aligned} \tag{4.18}$$

where $f(\cdot)$ is a function to be minimized over the decision variable $x \in \mathbb{R}^n$, subject to a conjunction of p clauses, with each clause being a disjunction of q_i inequalities.

In vehicle path planning applications, polytopic obstacles for which the vehicle has to remain outside an object can be expressed as the disjunctive constraints. Without loss of generality, assume there is a single polytopic obstacle constraint $\mathcal{C}$ with $0 \in \operatorname{int} \mathcal{C}$. The constraint can be expressed as

$$\bigvee_{i=1,\ldots,q} a_i^T x - 1 \geq 0, \tag{4.19}$$

where the vectors $a_1, \ldots, a_q$ represent the outward normals of N halfspaces that define the obstacle. The magnitude $\|a_i\|_2$ defines the distance of the halfspace from the origin. The position state of the vehicle at a single point in time is denoted x. The big-M method states

that (4.19) can be reformulated into mixed integer linear constraints

$$a_1^T x - 1 \geq M(1 - b_1) \tag{4.20}$$

$$a_2^T x - 1 \geq M(1 - b_2) \tag{4.21}$$

$$\vdots$$

$$a_q^T x - 1 \geq M(1 - b_q) \tag{4.22}$$

$$\sum_{i=1}^{q} b_i = 1 \tag{4.23}$$

where $b_i \in \{0, 1\}$ represent auxiliary binary variables in the problem and M is some sufficiently large number. The tightest value of M can be calculated from $M = \max_i\{a_i x - 1\}$ with $x_l \leq x \leq x_u$. If any of the binary variables equal 1, the corresponding original constraint is satisfied (for $b_i = 0$ the constraint is always satisfied). The disjunction is imposed by the constraint of (4.23) since this ensures that *only one* of the inequalities is satisfied. The big-M formulation can be expressed in matrix form as

$$Ax - \mathbf{1} \geq M(\mathbf{1} - b) \tag{4.24}$$

$$\mathbf{1}^T b = 1, \tag{4.25}$$

where A is a $q \times n$ matrix with a_i as its ith row vector and $b \in \{0, 1\}^q$ is a vector of binary variables $(b_1, \ldots, b_q)$.

As shown in the previous section, dualization can also be used to express disjunctive constraints of the type considered here. In the dualization method the constraint is expressed as

$$x^T y - 1 \geq 0 \tag{4.26}$$

$$y \in \mathcal{C}^\circ, \tag{4.27}$$

where $\mathcal{C}^\circ$ is the polar of $\mathcal{C}$. Since the polar of a polytope $\mathcal{C} = \{x | Ax \leq 1\}$ is defined as (Lemma 1.ii)

$$\mathcal{C}^\circ = \{y \mid \exists z, y = A^T z, z \geq 0, \mathbf{1}^T z = 1\}, \tag{4.28}$$

the constraints (4.26) and (4.27) become

$$x^T A^T z - 1 \geq 0 \tag{4.29}$$

$$z \geq 0 \tag{4.30}$$

$$\mathbf{1}^T z = 1 \tag{4.31}$$

where $z \in \mathbb{R}^q$ is a *continuous* auxiliary variable. Since both the big-M and dualization formulations characterize the same problem there is a relationship between the two auxiliary variables z and b. Taking (4.24) and multiplying by z gives

$$x^T A^T z - \mathbf{1}^T z \geq M(1-b)^T z. \tag{4.32}$$

Comparing (4.32) and (4.29) and noting $\mathbf{1}^T z = 1$ gives

$$M(1-b)^T z = 0. \tag{4.33}$$

The equation shows the relationship between the binary and continuous auxiliary variables in the two formulations and is only satisfied for a large M when $z = b$ (recalling that $\mathcal{B} \subseteq \mathcal{Z}$, with $\mathcal{B} := \{0,1\}^q$ and $\mathcal{Z} := \mathbb{R}^q$). In geometric terms, $z = b$ corresponds to the vertices of the polar set $\mathcal{C}^\circ$ (see Example 3).

The dualization method therefore demonstrates a relaxation of the big-M formulation through the use of continuous auxiliary variables instead of binary variables. The big-M method for disjunctive programming suffers computationally since MILPs are inefficient when the number of binary variables is large. The dualization method allows the use of a gradient based solver for computing (4.18), and is an efficient formulation for disjunctive programs when constraints are of the form shown in (4.19).

Example 3. *(Unit Square)*
The set $\mathcal{C} \subset \mathbb{R}^2$ representing a unit square ($Ax \leq 1$) can be written as

$$\begin{pmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{pmatrix} x \leq \mathbf{1}. \tag{4.34}$$

The polar set $\mathcal{C}^\circ \subset \mathbb{R}^2$ is then defined as

$$\left\{ y \;\middle|\; \exists y, y = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix} z, z \geq 0, \mathbf{1}^T z = 1 \right\} \tag{4.35}$$

which can be rewritten as

$$\left\{ y \;\middle|\; \exists y, y_1 = z_1 - z_2, y_2 = z_3 - z_4, z \geq 0, \mathbf{1}^T z = 1 \right\}. \tag{4.36}$$

Figure 4.4 shows the unit square and its polar set. The vertices of the polar set correspond to the binary variables of the big-M formulation.

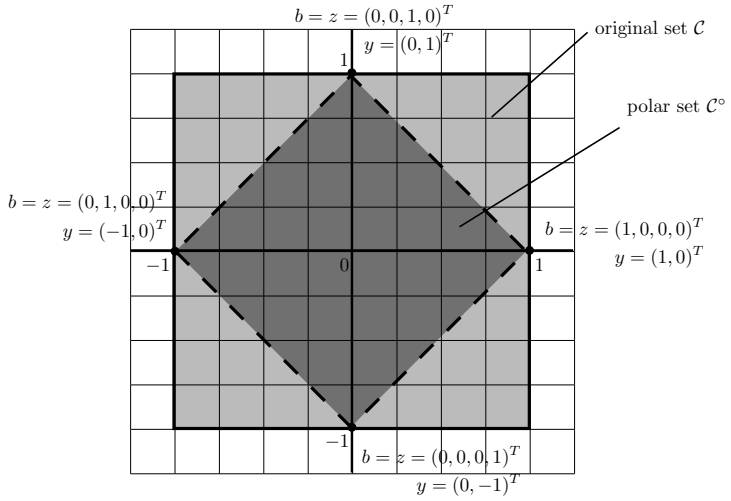


Figure 4.4: The unit square and its polar. The vertices correspond to the binary variables in the big-M formulation.

4.2.6 Trajectory Generation Outer Loop Controller

Given the multiple shooting optimal control problem $\mathcal{P}(x(t_0), u(t_0))$, a control sequence $\mathbf{u}^*$ is calculated that would produce a state sequence $\mathbf{x}^*$ assuming the system dynamics $f(x, u)$ are a perfect match to the real aircraft. In reality, model errors and disturbances would produce a state sequence $\mathbf{x} + \Delta\mathbf{x}$ for the outer loop controller. To mitigate this state error in the trajectory generation part of the scheme, a suitably robust inner loop controller is required for trajectory following. For the purposes of design of the outer loop controller it is assumed that the inner loop controller allows accurate tracking of $\mathbf{x}^*$. The inner loop trajectory following control strategy is discussed in Chapter 7.

The following section is concerned with finding a solution to $\mathcal{P}(x(t_0), u(t_0))$ such that a solution to $\mathcal{P}(x(t_f), u(t_f))$ exits, and more generally $\mathcal{P}(x(t), u(t))$ exists where $\forall t \in [t_0, \infty]$.

4.3 Extension of OCP2 to the Infinite Horizon

An obvious difficulty with the problem formulation OCP2 is that it generates a trajectory to avoid prohibited volumes over a finite time interval only, and does not therefore provide any assurance of obstacle avoidance over an infinite horizon [62]. The motivation for doing so is to ensure that a trajectory cannot end with an obstacle directly in the subsequent flight path. This effect is achieved by constructing a positively invariant target set, a standard method in predictive control, by employing dualization techniques similar to those of Section 4.2.4. The dualization approach here is required for both differentiable and non-differentiable obstacle

constraints as it is otherwise difficult to achieve feasibility over an infinite horizon. The construction of the invariant set also delivers an initial iterate for the next optimization via shift-initialization, therefore ensuring that the optimization starts near a feasible point.

It is important to note that whilst the terminal invariance technique of predictive control is being employed, the problem is not guaranteed to be recursively feasible. Since the control input is not actually applied to the aircraft whilst the pilot is still in command, the pilot can continue to fly until inside a PV. At some point close to the PV the problem becomes infeasible, i.e. no control action is available to avoid the PV. The aim of the PV Avoidance System is to remove control authority from the pilot *before* this situation can occur. The methods for doing so are presented in Chapter 6 .

4.3.1 Terminal Invariance Condition

The following terminal conditions are selected to extend the finite horizon solution to achieve safety over the infinite horizon:

- ensure the aircraft is flying level or in a shallow climb at the end of the trajectory.
- ensure the straight line trajectory of the aircraft at the end of the avoidance trajectory does not intersect any PVs.

The core difficulty with the current formulation of OCP2 is that the feasible set (4.15) does not satisfy the nesting condition

$$\cdots \subseteq \mathcal{X}_{i-1} \subseteq \mathcal{X}_i \subseteq \mathcal{X}_{i+1} \subseteq \cdots ,$$

i.e. the set of feasible initial conditions for OCP2 is not guaranteed to be monotonically non-decreasing with increasing horizon length. Such a guarantee can be provided by augmenting OCP2 with a suitable target/terminal set $\mathcal{X}_f \subseteq \mathbb{R}^n$, along with an appropriate terminal control law $\kappa_f : \mathbb{R}^n \to \mathbb{R}^m$. For simplicity the terminal control law is assumed to be constant, i.e. $\kappa_f(x) = u_f$ for all x. The following requirements are imposed on the terminal conditions:

Assumption 1. *The set $\mathcal{X}_f \subseteq \mathbb{R}^n$ and constant control law $u_f = \kappa_f(x)$ satisfy the following conditions:*

i) The constraints $x \in X$ are satisfied for all $x \in \mathcal{X}_f$.

ii) The control input $u_f = \kappa_f(x)$ satisfies $u_f \in U$ for all $x \in \mathcal{X}_f$.

iii) The set $\mathcal{X}_f$ is positively invariant under the control law κ_f, i.e. $\phi(x, \kappa_f(x)) \in \mathcal{X}_f$ for all $x \in \mathcal{X}_f$.

iv) The position components of $\mathcal{X}_f$ do not intersect any prohibited volume, i.e.

$$\mathbb{P}_3\mathcal{X}_f \bigcap (\operatorname{int} \mathcal{C}_j \oplus r_j) = \emptyset, \quad \forall j \in \{1, \ldots, M\}, \tag{4.37}$$

where $\mathbb{P}_3\mathcal{X}_f$ is the projection of the set $\mathcal{X}_f$ onto the position states.

A set satisfying the above assumptions can be constructed by imposing conditions to ensure that the terminal set $\mathcal{X}_f$ and input u_f represent a level flight condition or steady climb, and that the straight-line extrapolation of the aircraft terminal state does not intersect any PVs.

The first of these conditions can be satisfied by ensuring that the state x and terminal control input u_f satisfy

$$Gf(x, u_f) = 0 \tag{4.38}$$

for $x \in \mathcal{X}_f$, where $G \in \mathbb{R}^{6\times 6}$ is a diagonal matrix selecting the three states $\dot{\gamma}$, $\dot{\chi}$ and $\dot{v}$. Denoting the vehicle velocity vector at some position state p as v, and defining the straight line projection of a given state as

$$R(p, v) \triangleq \{p + \beta v \mid \beta \geq 0\},$$

the condition (4.37) can be satisfied by ensuring that the ray $R(p, v)$ does not intersect any prohibited volume for any $x \in \mathcal{X}_f$. A set $\mathcal{X}_f$ can now be constructed satisfying all of the conditions of Assumption 1, by defining it as

$$\mathcal{X}_f \triangleq \left\{ x \;\middle|\; \begin{array}{c} \exists u,\ Gf(x, u) = 0, \\ (x, u) \text{ satisfy constraints } x \in X,\ u \in U, \\ R(p, v) \bigcap \operatorname{int} \mathcal{C}_j \oplus r_j = \emptyset, \\ \forall j \in \{1, \ldots, M\} \end{array} \right\},$$

The following analysis provides a result that will allow us to treat the set exclusion condition of Assumption 1.iv entirely in terms of continuously differentiable functions. This is achieved by once again exploiting the results of Section 3.2.2 to remodel the problem in terms of the polar sets of $\mathcal{C}_j$.

Proposition 1. *Suppose that the vectors $(p, v) \in \mathbb{R}^3$ parameterize a ray*

$$R(p, v) \triangleq \{p + \beta v \mid \beta \geq 0\},$$

and C is a closed and convex set containing the origin in its interior, and $r \in \mathbb{R}^3$. Then the condition

$$R(p, v) \bigcap (\operatorname{int} C \oplus r) = \emptyset$$

is satisfied if and only if

$$\left\{ y \;\middle|\; (p-r)^T y \geq 1,\ v^T y \geq 0,\ y \in C^\circ \right\} \neq \emptyset. \tag{4.39}$$

Proof. Rewriting in terms of the gauge function of C, the first statement is equivalent to

$$\inf_z \left\{ \gamma_C(z) \mid \beta \geq 0,\ z = (p-r) + \beta v \right\} \geq 1. \tag{4.40}$$

The Lagrange dual function for this optimization problem is

$$g(\lambda, y) = \inf_{\beta, z} \left(\gamma_C(z) + \left((p-r) + \beta v - z\right)^T y - \beta\lambda \right)$$

with multipliers (λ, y). This can be rewritten as

$$\begin{aligned} g(\lambda, y) &= \inf_z \big(\gamma_C(z) - z^T y\big) + \inf_\beta \big(\beta(v^T y - \lambda)\big) + (p-r)^T y \\ &= -\sup_z \big(z^T y - \gamma_C(z)\big) + \inf_\beta \big(\beta(v^T y - \lambda)\big) + (p-r)^T y \\ &= -\gamma_C^*(y) + \inf_\beta \big(\beta(v^T y - \lambda)\big) + (p-r)^T y. \end{aligned}$$

Recalling from Lemma 2.ii that the gauge function's conjugate γ_C^* is equivalent to the indicator function of the polar set C°, the above simplifies to

$$g(\lambda, y) = \begin{cases} (p-r)^T y & \text{if } y \in C^\circ \text{ and } v^T y = \lambda \\ -\infty & \text{otherwise.} \end{cases}$$

Since the minimization problem (4.40) is convex and contains only linear constraints, Slater's condition is satisfied and its dual maximization problem

$$\sup_{(y, \lambda \geq 0)} g(\lambda, y) = \sup \left\{ (p-r)^T y \;\middle|\; v^T y \geq 0\ , y \in C^\circ \right\}$$

obtains the minimum value of (4.40). Since the polar set C° is compact (Lemma 1.i), a maximizer for this problem always exists. Therefore the set (4.39) is guaranteed to be nonempty if and only if (4.40) holds. □

The generalized condition (4.39) for a closed and convex set C can be used to derive constraints for polyhedral and p-norm ball prohibited volumes using Lemma 1 (see Figure 4.5).

Corollary 1. *(polyhedral prohibited volume) The condition*

$$R(p, v) \bigcap (\operatorname{int} C \oplus r) = \emptyset$$

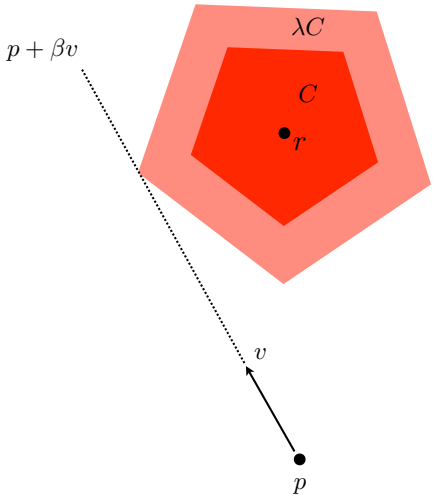


Figure 4.5: Polyhedral PV and scaling via the gauge function. If $\lambda > 1$ the aircraft will remain outside the PV.

with

$$C = \{p \mid Hp \leq 1\}$$

is satisfied if and only if

$$\left\{\eta \;\middle|\; (p-r)^T H^T \eta \geq 1,\ v^T H^T \eta \geq 0,\ \eta^T \mathbf{1} = 1,\ \eta \geq 0\right\} \neq \emptyset. \tag{4.41}$$

Proof. From Lemma 1.ii, the condition $y \in C^\circ$ is equivalent to

$$C^\circ = \left\{y \;\middle|\; \exists \eta,\ y = H^T \eta,\ \eta \geq 0,\ \eta^T \mathbf{1} = 1\right\}.$$

Substituting into (4.39) obtains the condition (4.41) above. □

Corollary 2. *(p-norm ball prohibited volume) The condition*

$$R(p, v) \bigcap (\operatorname{int} C \oplus r) = \emptyset$$

with

$$C = \{p \mid \|Hp\| \leq 1\}$$

is satisfied if and only if

$$\left\{\eta \;\middle|\; (p-r)^T H^T \eta \geq 1,\ v^T H^T \eta \geq 0,\ \|\eta\|_* \leq 1\right\} \neq \emptyset. \tag{4.42}$$

Proof. From Lemma 1.iv, the condition $y \in C^\circ$ is equivalent to

$$C^\circ = \left\{y \;\middle|\; \exists \eta,\ y = H^T \eta,\ \|\eta\|_* \leq 1\right\}.$$

Substituting into (4.39) obtains the condition (4.42) above. □

In summary, using the duality property of the Lagrange dual function [12], the original inner minimization of (4.40) is converted into a maximization where only the existence of a solution has to be proved for the constraint to be satisfied. The constraints can be interpreted geometrically in two dimensions as shown in Figures 4.6 and 4.7. In these diagrams the PV is shown centered at the origin ($r = 0$) and the shaded halfspaces are regions where η is excluded. As well as defining the bounding region shown in the diagrams (offset from the center), the constraint $p^T H^T \eta \geq 1$ also ensures that the starting point is not already inside the PV. The permissible region for the η vector narrows as the projected velocity passes closer to the PV. If the projected velocity passes through the PV, the set (4.42) is empty.

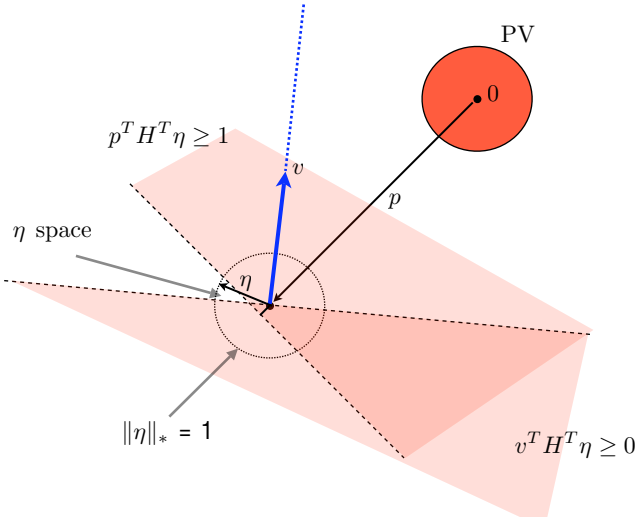


Figure 4.6: Projected velocity far from PV.

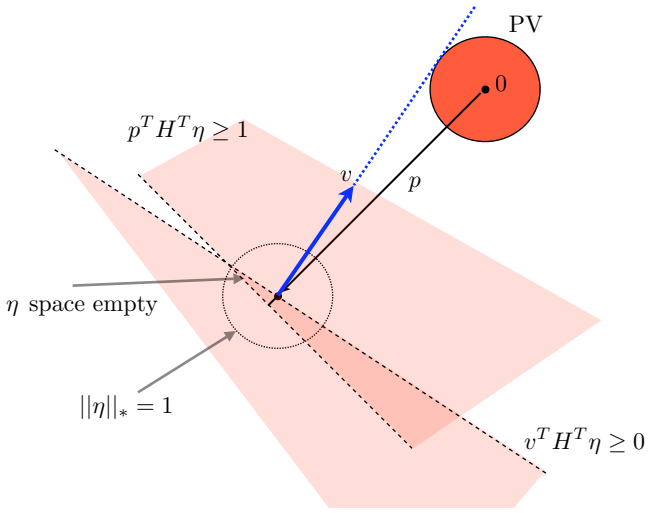


Figure 4.7: Projected velocity touching PV.

The additional constraints are easy to incorporate into the problem structure of (OCP2):

$$
\begin{aligned}
\min_{\mathbf{x},\mathbf{u},y_{i,j},\eta_j} \quad & J_N(\mathbf{x},\mathbf{u}) \\
\text{subject to:} \quad & g_{cont}(\mathbf{x}) = 0 \\
& h_{rate}(\mathbf{x},\mathbf{u}) \leq 0 \\
& \mathbf{x} \in X, \quad \mathbf{u} \in U \\
& \left.\begin{aligned}
& (p_i - r_j)^T y_{i,j} \geq 1, \quad \forall i \in \{0,\dots,N\} \\
& y_{i,j} \in C_j^\circ, \qquad\qquad \forall i \in \{0,\dots,N\} \\
& (p_N - r_j)^T \eta_j \geq 1 \\
& v_N^T \eta_j \geq 0 \\
& \eta_j \in C^\circ
\end{aligned}\right\} \quad \forall j \in \{1,\dots,M\}
\end{aligned}
$$

where $y_{i,j}$ and η_j are auxiliary variables in the problem. Note that the constraints of this problem are differentiable (cf. Remark 6) and, in the case of polytopic obstacles, can be solved without the introduction of integer variables. This preserves the overall continuity of the problem, ensuring that its solution is amenable to the use of standard solvers for continuous nonlinear optimization.

4.3.2 Warm Starting

The proposed scheme has similarities with a traditional MPC formulation. However, two key differences exist. Firstly, in normal operation before the override is triggered, the control generated by the system is not applied because the pilot retains control of the aircraft. Therefore it is possible for the problem to become infeasible close to a PV. This is mitigated by ensuring that an override is triggered before this can happen. Secondly, when the override has been triggered the control input generated by the optimization is not explicitly applied as there is no obvious mapping of inputs from the 3DOF model to the real aircraft. Instead, an inner-loop controller is used to track the trajectory. Once the override is activated and the aircraft is being flown by the controller, the optimization is not recalculated at each time step as is required for traditional MPC. Instead, the last avoidance trajectory generated is flown to completion via the inner-loop controllers (see the Future Work section of Chapter 5.7 for a discussion on implementing the override in a receding horizon fashion).

There is a choice available in how to initialize the optimization, both for the decision variables and the multipliers. Warm starting is commonly used to decrease the computation time for the optimization but can have an affect on an interior-point algorithm's ability to find a solution as shown in Chapter 5. In this work only warm starting of the the primal (decision) variables in the optimization are investigated (note that auxiliary variables in the problem are always

initialized to zero) as the primary intention of warm-starting is to ensure the optimization is initialized close to a feasible solution in the primal space. Assuming an optimal solution $(x_0^*; u_0^*; x_1^*; u_1^*; \ldots; u_{N-1}^*; x_N^*)$ to the original problem (OCP2) has been computed, in the finite horizon case the next calculation at future time t_1 can be initialized with $(x_1^*; u_1^*; \ldots; u_{N-1}^*; x_N^*)$. For the moving horizon case the new variables at the end of the time horizon u_N, x_{N+1}, also need to be initialized. This is often done using 'shift initialization' where the state x_{N+1} is obtained by forward simulation holding the final input constant, i.e. $u_N := u_{N+1}$ [21]. The positive invariance aspect of the formulation serves two purposes. Given the assumption of perfect tracking of $\mathbf{x}$, a solution to $\mathcal{P}(x(t_f), u(t_f))$ is known to exist and therefore the aircraft is not left in a trapped situation at the end of an avoidance trajectory. The solution of $\mathcal{P}(x(t_0), u(t_0))$ can also be used before the override is triggered to initialize the problem for the next time step using shift initialization, as opposed to initialization using the reference trajectory (recall that the reference trajectory is generated by holding the input constant at $u(t_0)$ and integrating forwards from the initial state $x(t_0)$). This prevents the decision variable becoming 'stuck' in an infeasible region and ensures that a solution can be found when one is available, as demonstrated in the results of the following chapter.

Chapter 5

Implementation and Results for Trajectory Generation

5.1 Choice of Integration Scheme and Error Bounds

In the multiple shooting formulation of OCP2, the system state is propagated across a time step h via integration:

$$\phi(x_i, u_i) \triangleq x_i + \int_t^{t+h} f(x(\tau), u(\tau))d\tau. \tag{5.1}$$

Conventionally, integration in the multiple shooting method would be performed by a state-of-the-art adaptive variable step, variable order ODE solver (usually based on the Adam-Moulton formula), with quadratures calculated to provide sensitivity information for the Jacobian matrix. Collocation methods using polynomial approximations to the integration across a fixed grid are also available. These typically result in larger problems as intermediate nodes are required in the segment which appear explicitly in the equality constraints for the problem. However, collocation can result in a more sparsely banded Jacobian structure. The advantage of the traditional multiple shooting method is that an adaptive grid is 'inbuilt' via the use of an adaptive ODE solver and hence the integration error is managed within the optimization process. For collocation this is not the case since the grid size is fixed, hence optimization with a refined grid is required if the integration error is found to be excessive.

Referring back to the constraint vector in multiple shooting given by (4.11), for collocation the constraints are implicit. For example, if a 2nd order Runge-Kutta integration scheme is

adopted the constraint vector would become:

$$c(\mathbf{z}) = \left[\begin{array}{c} x(t_0) - x_0 \\ x_0 + \frac{h}{2}(f_1 + f_0) - x_1 \\ \vdots \\ x_{N-1} + \frac{h}{2}(f_N + f_{N-1}) - x_N \\ \hline h_{rate}(\mathbf{z}) \\ h_{PV}(\mathbf{z}) \end{array}\right] \tag{5.2}$$

where $f_k := f(x_k, u_k)$. This is the equivalent of fitting a 2nd order polynomial which matches the function at the beginning of the interval and slope at times t_k and t_{k+1}. Theoretically a multiple shooting scheme using an explicit Euler or Runge-Kutta integration method is equivalent to a collocation scheme with the corresponding integration method where no intermediate (state) node points are used. However, from an implementation stand-point the two are different since in the multiple shooting case an external function is called to evaluate the constraints and Jacobian of constraints. Therefore, there is a computational overhead associated with implementing a fixed order scheme for multiple shooting. However, the advantage of doing so is that different integration schemes can be more easily implemented and compared since the constraint vector does not need to be modified if an external function is called.

In this section the use of fixed order explicit schemes such as Euler and Runge-Kutta is investigated and the performance compared to a higher order solver. The motivation for doing so is to ascertain whether a lower order solver, which is computationally less expensive, can still provide sufficient performance for the trajectory generation problem.

The first order Euler method can be written as

$$\phi(x_i, u_i) = x_i + f(x_i, u_i)h, \tag{5.3}$$

where h is the step size. The gradients, required to construct the Jacobian of the multiple shooting strategy, are given by

$$\frac{\partial \phi}{\partial x} = I + \frac{\partial f}{\partial x}h \tag{5.4}$$

$$\frac{\partial \phi}{\partial u} = \frac{\partial f}{\partial u}h. \tag{5.5}$$

The gradients $\partial f/\partial x$ and $\partial f/\partial u$ are evaluated by analytical differentiation of the aircraft model (3.1).

The integration error is an important consideration since a cumulative error over the time horizon for the trajectory indicates the possible set of states of the aircraft in relation to the final state (x_N). An integration scheme cannot be used if it demonstrates a larger than acceptable

error. Ideally the integration error should be small compared to model errors which also affect the accuracy of the trajectory generation scheme.

Errors due to numerical integration schemes can be classified as *local* or *global*. Local errors include input errors (errors in the input data for the problem), discretization error and truncation (round-off) error. The global error is the error in the final numerical result. The input data for the multiple shooting problem is assumed to be correct. Given the machine precision for representation of floating point numbers, the discretization error dominates the truncation error. The following result is a generalization of results in [6] to vector functions:

Proposition 2. *The global error bound estimate for the Euler first-order time integration of* $\dot{x} = f(x, u)$ *with piecewise constant* u *and time step* h *is given by*

$$E \approx \frac{h}{2}(t_f - t_0) \sup_{x,u} \left\| \frac{\partial f}{\partial x} f \right\|_\infty, \qquad x \in X, u \in U. \tag{5.6}$$

Proof. First take a Taylor series expansion, noting the control u is assumed constant over the multiple shooting interval, so that

$$x(t + h) = x(t) + hf + \frac{h^2}{2}\frac{df}{dt} + O(h^3). \tag{5.7}$$

For small h, the higher order errors $O(h^3)$ are dominated by the second order term. Therefore the maximum local discretization error over one time step is

$$\epsilon = \sup_{x,u} \left\| \int_t^{t+h} f(x, u)dt - \phi(x, u) \right\|_\infty \approx \frac{h^2}{2} \sup_{x,u} \left\| \frac{df}{dt} \right\|_\infty, \qquad x \in X, u \in U. \tag{5.8}$$

The global error $E = n\epsilon$ where n is the number of multiple shooting steps. As $n = (t_f - t_0)/h$ the global error is

$$E = \frac{h}{2}(t_f - t_0) \sup_{x,u} \left\| \frac{df}{dt} \right\|_\infty. \tag{5.9}$$

Noting that the ordinary differential can be defined in terms of partials,

$$\frac{df}{dt} = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial x}\frac{dx}{dt} = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial x} f, \tag{5.10}$$

and $\partial f/\partial t = 0$ since t does not explicitly appear in the function $f(x, u)$, the global error bound can be written as (5.6). □

This demonstrates that the order of error for the Euler method is $O(h)$. Although easy to implement, the Euler scheme generally exhibits poor convergence to the actual solution with decreasing h. Therefore higher order schemes are usually preferred.

One such scheme is the explicit 2nd order Runge-Kutta discretization. The general second

order Runge-Kutta scheme can be written as

$$\phi(x_i, u_i) = x_i + \left[w_1 f(x_i, u_i) + w_2 f(k, u_i)\right] h, \tag{5.11}$$

$$k = x_i + \beta f(x_i, u_i) h, \tag{5.12}$$

with gradients

$$\frac{\partial \phi}{\partial x} = I + \left[w_1 \frac{\partial f}{\partial x} + w_2 \frac{\partial f}{\partial k}\frac{\partial k}{\partial x}\right] h, \tag{5.13}$$

$$\frac{\partial k}{\partial x} = I + \beta \frac{\partial f}{\partial x} h, \tag{5.14}$$

$$\frac{\partial \phi}{\partial u} = \left[w_1 \frac{\partial f}{\partial u} + w_2 \frac{\partial f}{\partial k}\frac{\partial k}{\partial u}\right] h, \tag{5.15}$$

$$\frac{\partial k}{\partial u} = \beta \frac{\partial f}{\partial u} h. \tag{5.16}$$

Proposition 3. *A minimum error bound is obtained for an explicit second order Runge-Kutta time integration of $\dot{x} = f(x, u)$ with piecewise constant u and time step h when the coefficients $w_1 = 1/4$, $w_2 = 3/4$ and $\beta = 2/3$.*

Proof. The method of proof follows [54]. Consider the two vectors

$$k_1 = hf(x, u) \tag{5.17}$$

$$k_2 = hf(x + \beta k_1, u), \tag{5.18}$$

which form the weighted mean $\bar{k} = w_1 k_1 + w_2 k_2$, with w_1 and w_2 as weighting coefficients. The aim is to determine the constants β, w_1, w_2 such that $\bar{k}$ agrees with the Taylor series expansion of $x(t + h) - x(t)$ to within $O(h^3)$. The Taylor series (5.7) can be written as

$$x(t + h) - x(t) = hf + \frac{h^2}{2}\frac{\partial f}{\partial x} f + \frac{h^3}{6}\left(\frac{\partial^2 f}{\partial x^2} f^2 + \left(\frac{\partial f}{\partial x}\right)^2 f\right) + O(h^4), \tag{5.19}$$

where $\partial^2 f / \partial x^2$ is a tensor and f^2 is a scalar (the product is a vector). Taking a Taylor expansion of k_2,

$$k_2 = h\left(f + h\beta \frac{\partial f}{\partial x} f + \frac{h^2}{2}\beta^2 \frac{\partial^2 f}{\partial x^2} f^2 + O(h^3)\right) \tag{5.20}$$

and substituting into $\bar{k}$ obtains,

$$\bar{k} = w_1 hf + w_2 h\left(f + h\beta \frac{\partial f}{\partial x} f + \frac{h^2}{2}\beta^2 \frac{\partial^2 f}{\partial x^2} f^2\right) \tag{5.21a}$$

when higher order terms in k_2 are dropped. Comparing (5.7) and (5.21a) for terms in h, h^2

and h^3, the following relations are found:

$$w_1 + w_2 = 1 \tag{5.22a}$$

$$w_2\beta\frac{\partial f}{\partial x}f = \frac{1}{2}\frac{\partial f}{\partial x}f \tag{5.22b}$$

$$\frac{1}{2}w_2\beta^2\frac{\partial^2 f}{\partial x^2}f^2 = \frac{1}{6}\left(\frac{\partial^2 f}{\partial x^2}f^2 + \left(\frac{\partial f}{\partial x}\right)^2 f\right). \tag{5.22c}$$

From (5.22b), $w_2\beta = 1/2$. The minimum error is then obtained by eliminating the $\partial^2 f/\partial x^2$ terms in (5.22c) which gives $\beta = 2/3$ and $w_2 = 3/4$. The final constant is $w_1 = 1/4$ from (5.22a). □

Corollary 3. *A global error bound estimate for the 2nd order Runge-Kutta time integration of $\dot{x} = f(x, u)$ with piecewise constant u and time step h, using the form*

$$\phi(x_i, u_i) = x_i + \left[\frac{1}{4}f(x_i, u_i) + \frac{3}{4}f(k, u_i)\right]h, \tag{5.23}$$

$$k = x_i + \frac{2}{3}f(x_i, u_i)h, \tag{5.24}$$

is given by

$$E = \frac{h^2}{6}(t_f - t_0)\sup_{x,u}\left\|\left(\frac{\partial f}{\partial x}\right)^2 f\right\|_\infty, \qquad x \in X, u \in U. \tag{5.25}$$

Proof. From Proposition 3 the local error bound in h^3 is given by

$$\epsilon = \sup_{x,u}\left\|\int_t^{t+h} f(x, u)dt - \phi(x, u)\right\|_\infty \approx \frac{h^3}{6}\sup_{x,u}\left\|\left(\frac{\partial f}{\partial x}\right)^2 f\right\|_\infty. \tag{5.26}$$

Since $n = (t_f - t_0)/h$ and the global error $E = n\epsilon$, the result (5.25) is obtained. □

The order of error is now $O(h^2)$. Two function evaluations are required per time step increasing the computational cost when compared with the Euler scheme. If a time integration is performed over n steps, the maximum (global) error per function evaluation for the Euler scheme is $n\epsilon_E/n = \epsilon_E$, whilst for the Runge-Kutta scheme it is $n\epsilon_R/2n = \epsilon_R/2$. Therefore the Runge-Kutta scheme has a lower error per function evaluation when $\epsilon_R < \epsilon_E$, i.e. when

$$h\sup_{x,u}\left\|\left(\frac{\partial f}{\partial x}\right)^2 f\right\|_\infty < 6\sup_{x,u}\left\|\frac{\partial f}{\partial x}f\right\|_\infty. \tag{5.27}$$

It might be advantageous to use a more advanced variable order, variable step ODE solver such as CVODES [42] based on the Adam-Moulton formulae. Whilst a higher order scheme allows a larger step size for a fixed number of function evaluations, for a control application a balance is required between error, computational cost and control signal smoothness. Since a zero-order hold is implemented for the control within each multiple shooting segment, large

step sizes would result in less control freedom. Some numerical experiments investigating the integration error for differing discretization schemes are presented in the following section.

5.2 Integration Scheme - Numerical Experiments

In Section 5.1 theoretical error bounds for the Euler and 2nd order Runge-Kutta (RK2) methods were derived. The global error bounds for the aircraft specific problem can now be determined.

To evaluate $(\partial f/\partial x)f$ and $(\partial f/\partial x)^2 f$ for the Euler and RK2 bounds respectively first evaluate the maximum element wise magnitudes of the matrix $(\partial f/\partial x)$:

$$\begin{bmatrix} \max_{x\in X,u\in U}|a_{11}| & \max_{x\in X,u\in U}|a_{12}| & \dots \\ \max_{x\in X,u\in U}|a_{21}| & \max_{x\in X,u\in U}|a_{22}| & \dots \\ \vdots & \vdots & \ddots \end{bmatrix} \tag{5.28}$$

and f:

$$\begin{bmatrix} \max_{x\in X,u\in U}|f_1| \\ \max_{x\in X,u\in U}|f_2| \\ \vdots \end{bmatrix} \tag{5.29}$$

The values are computed using a constrained optimization function in Matlab and the global errors are displayed as a function of step size h in Table 5.1. Using these values the maximum

State	Euler $\left(\times(t_f - t_0)\frac{h}{2}\right)$	RK2 $\left(\times(t_f - t_0)\frac{h^2}{6})\right)$
x (m)	260	68
y (m)	260	68
z (m)	162	66
γ (deg)	11.8	4.4
χ (deg)	3.2	1.3
V (m/s)	10	4

Table 5.1: Integration error bounds for the Euler and RK2 methods.

step size allowable for a given maximum global error E can be calculated. For example if 50m is deemed to be an acceptable error in x for a 20s time horizon, the Euler method would require a step size less than $(50 \times 2)/(260 \times 20) = 0.02$s. The Runge-Kutta method, on the other hand, would require a step size of $\sqrt{(50 \times 6)/(68 \times 20)} = 0.47$s.

Remark 7. *There are less conservative methods for evaluating the vectors $(\partial f/\partial x)f$ and $(\partial f/\partial x)^2 f$, however these require greater computation. Deducing that the error bounds are still acceptable with the more conservative method shown is sufficient.*

Numerical experiments show that the actual errors due to the integration method are much lower than the bounds given here. In the following numerical experiments, five representative

simulations are conducted where avoidance trajectories are generated from different initial conditions for an aircraft heading into a spherical PV. The test cases are shown in Figure 5.1. Since a true solution is unavailable for the integrations, the variable order variable step solver

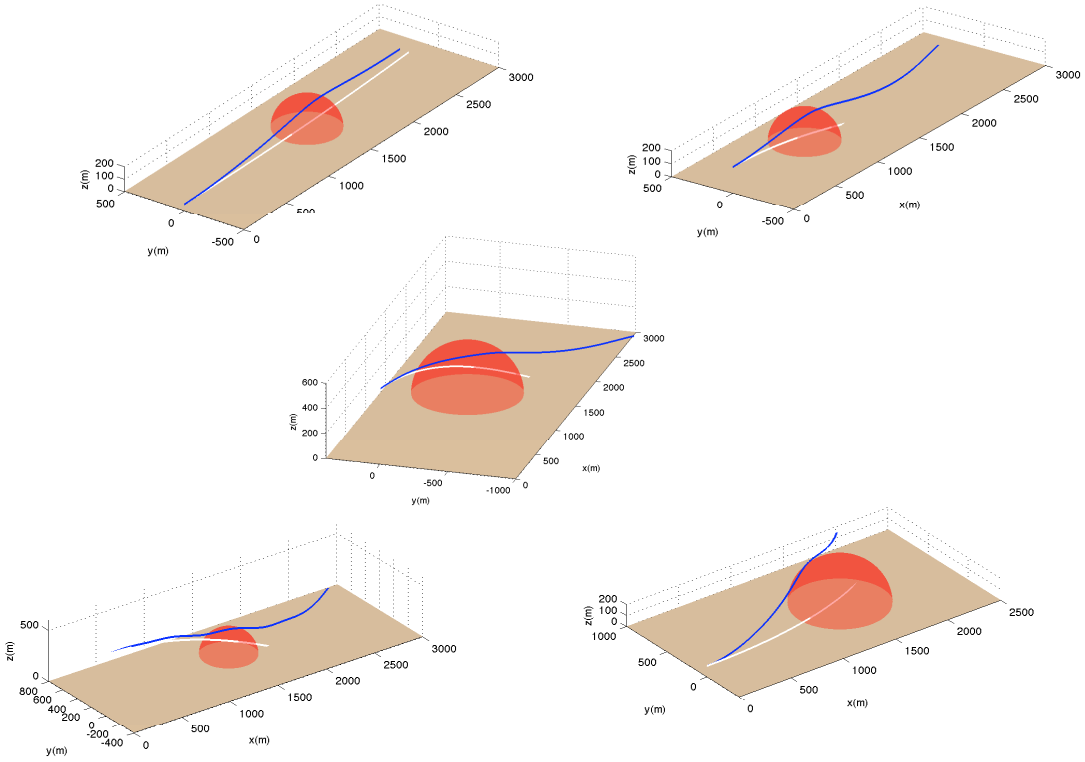


Figure 5.1: Integration test cases.

CVODES (set with an absolute error tolerance of $1e^{-16}$) is used. The errors for the Euler and RK2 methods are calculated relative to the CVODES solution. The maximum global error across the five cases is displayed in Figure 5.2 for the x, y and z states. The graphs show that the actual errors for both integration schemes are significantly lower than the derived conservative bound values. In both cases the modelling errors and disturbances (e.g. wind) will dominate. As expected the RK2 errors are lower than the Euler values, and the maximum errors across all five simulations are 9m and 29m respectively. Table 5.2 shows the bound values and maximum errors obtained for all states. The number of function evaluations $f(x, u)$ required for the Euler and RK2 methods are 50 and 100 respectively (for $N = 50$ integration steps). The higher order solver CVODES with strict tolerances required on average 2307 function evaluations for the simulation cases tested.

As well as analysing the relative errors between the Euler and RK2 methods for a fixed discretization for the integration, the computation time and error can be analysed for different integration discretizations. This is done in Figure 5.3 for a fixed control problem (i.e. a single simulation test case with $N = 50$ multiple shooting nodes). The graph on the left shows the

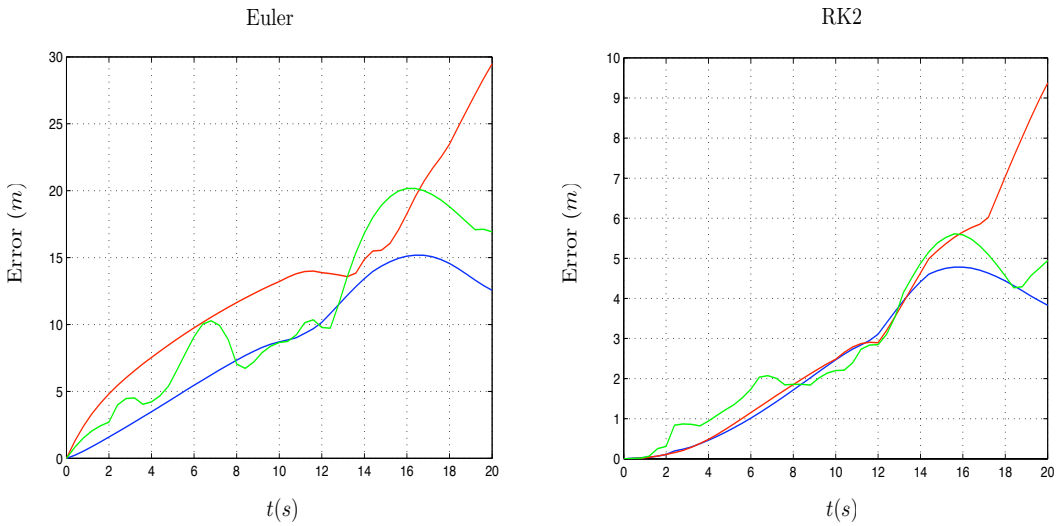


Figure 5.2: The maximum integration error for x (blue), y (red) and z (green) states over five simulations. There are 50 integration steps with a step size of 0.4s.

State	Euler derived bound	Max. Euler error from simulation	RK2 derived bound	Max. RK2 error from simulation
x (m)	1004	15	36.2	5
y (m)	1004	29	36.2	9
z (m)	648	20	35.2	6
γ (deg)	47	0.7	2.4	0.24
χ (deg)	13	0.2	0.7	0.04
V (m/s)	40	1.6	0.2	0.43

Table 5.2: Comparison between conservative bound and actual simulation values of integration error.

computation time for the integration relative to the fastest (Euler with 20 integration steps) and the graph on the right shows the corresponding final error in the position states. The Euler method shows some improvement at finer discretizations but for RK2 the error remains relatively constant.

The conclusion drawn from this analysis is that the 2nd order Runge-Kutta method with a time step of 0.4s is suitable for the trajectory generation application.

5.3 Software Implementation

The software for solving OCP2 was initially prototyped in Matlab, and then implemented in C++ using an object orientated methodology. A basic outline for the software structure is shown in Figure 5.4. There are a number of optimization packages available for solving constrained nonlinear problems. A gradient based interior-point approach is chosen (described in the background section of Chapter 3) since these methods are typically efficient for large scale

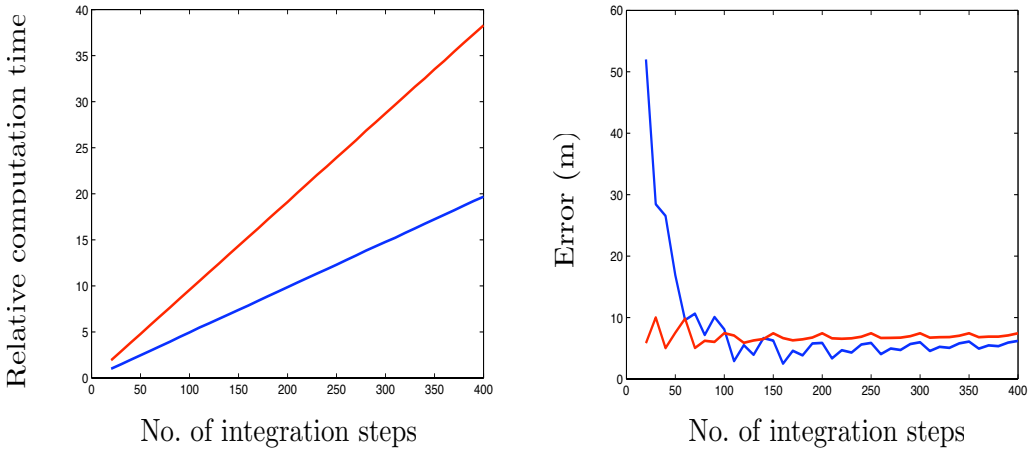


Figure 5.3: Left: relative computation time vs integration discretization. Right: 2-norm error in position vs integration discretization. Euler is shown in blue and RK2 in red.

problems, particularly when there are a number of active constraints [7]. The advantage of using a multiple shooting strategy together with a gradient based optimization is the sparse bounded Jacobian matrix resulting from the problem structure. The algorithm chosen, IPOPT [91], uses a barrier method which prevents iterates becoming infeasible once inside the feasible region. The solver can handle equality and inequality constraints as provided by the user.

The Hessian of the Lagrangian, $\nabla^2\mathcal{L}$, has a block diagonal structure described by Diehl et. al. [20]. In problems with a least squares form of the objective function $J(\cdot)$, Hessian block approximations can be used that do not depend on the values of the Lagrange multipliers. However, since the Hessian is made up of second derivatives of the objective function and constraints, it is not easy to compute analytically. Therefore, it is approximated with a limited memory quasi-Newton method (L-BFGS) in the optimization algorithm [94].

5.4 Trajectory Generation Results

The first part of this section describes some scenarios for which the optimization problem OCP2 (including end constraints) has been run with various PV constraints. The second part looks at the effect of warm starting on computation time and demonstrates how the combination of warm starting and positive invariance provides a means of ensuring that the optimization is initialized close to a feasible point.

The aircraft data for a Boeing 767 is used to conduct the simulations [47]. Details of the flight model development can be found in Appendix C. The implementation contains an additional throttle to thrust model based on the engine type of the aircraft. In this case the Pratt and

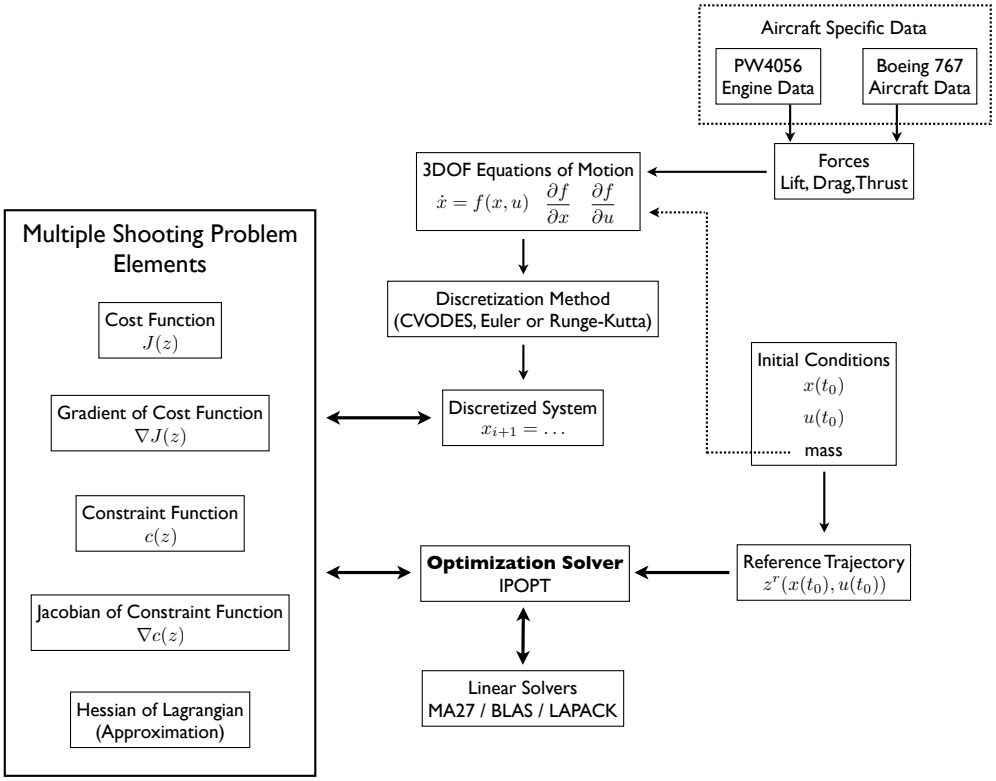


Figure 5.4: Software Structure.

Whitney 4056 installed engine thrust $T(z, v)$ is modeled using data from the design operating points and Mattingly's code is used to deduce the off design data [70]. The relationship between throttle and thrust is then assumed to be linear. This would not be the case in reality. However in an actual implementation the data would be available and an appropriate curve derived. In the optimization the explicit second order Runge-Kutta integration scheme is used to propagate the model dynamics. The number of multiple shooting segments is $N = 50$ and the time horizon $t_f = 20s$. In all the simulations shown the cost matrices $P = Q = I$ and $R = I$, i.e deviations of all the states and inputs away from their reference values are penalized. In the software, angular states are measured in radians and position states in kilometers to avoid problems associated with scaling (in the plots metres and degrees are used).

5.5 Algorithm Performance

Results from a number of tests are shown in Figures 5.5 - 5.8. In the state/input graphs, on the right of each figure, the aircraft states are given in the top two rows and the inputs in the final row of the three-by-three matrix. In the trajectory graphs the reference trajectory (the current predicted trajectory x^r with inputs u^r held fixed) is shown in white, and the problem solution (avoidance trajectory) shown as the darker line. The dashed line appended to the avoidance

trajectory shows the projected final trajectory with the final input u_N held fixed. The PVs are placed with their centres at $z_c = 0$ (ground level) and the ground is assumed to be flat. The simulation is started with the aircraft traveling along the x_c axis. The optimization is initialized with the calculated reference trajectory, and the auxiliary variables $u_N = u_{N-1}$, $\eta_j = 0$ and $y_{i,j} = 0$ for polytopic PV constraints. The number of iterations to convergence and time taken by the algorithm are given in the figure captions. The trajectory generation algorithm of OCP2 is implemented in C++ and simulations performed on a 2.4 GHz Intel iMac.

Conservative bounds on the max/min velocity are calculated from the aircraft's initial state. The aircraft is assumed to be flying in a clean configuration (i.e. without flaps and landing gear), however different models would incorporate these conditions. A ground constraint such that $z > 0$ for all t is imposed so that the aircraft cannot be flown into the ground. As well as the rate constraints on all the inputs and flight path angle $\dot{\gamma}$, bound constraints are applied to the absolute values ($|\mu| \leq 20^\circ$ and $|\gamma| \leq 20^\circ$) to prevent the aircraft manoeuvring excessively. A minimum final height of 50m is imposed to ensure safe separation from the ground at the end of the trajectory. The flight path angle at the end of the trajectory is constrained to be $0^\circ \leq \gamma \leq 5^\circ$ to ensure the aircraft is not pointed nose down or in an excessive climb. The terminal invariance constraint $\dot{\chi} = 0$ ensures that the roll angle μ is zero at the end of the trajectory giving straight flight.

5.5.1 Single Hemispherical PV

The first case shown in Figure 5.5 is a single hemispherical PV with a radius of 250m. The aircraft is started on a banked trajectory shown in white, passing through the side of the PV. The avoidance trajectory climbs above the PV and leaves the aircraft in a steady climb. The α profile shows a drop after 10s as the solution tries to move back to the reference trajectory (recall that deviations from this line are penalized). There is a subsequent increase to ensure the γ end condition is satisfied. The μ profile shows a drop to zero so that the aircraft is wings-level at the end of the trajectory. The solve time for the 13 IPOPT iterations is 0.24s, indicating that the solution method is viable for online trajectory generation in the PV Avoidance System.

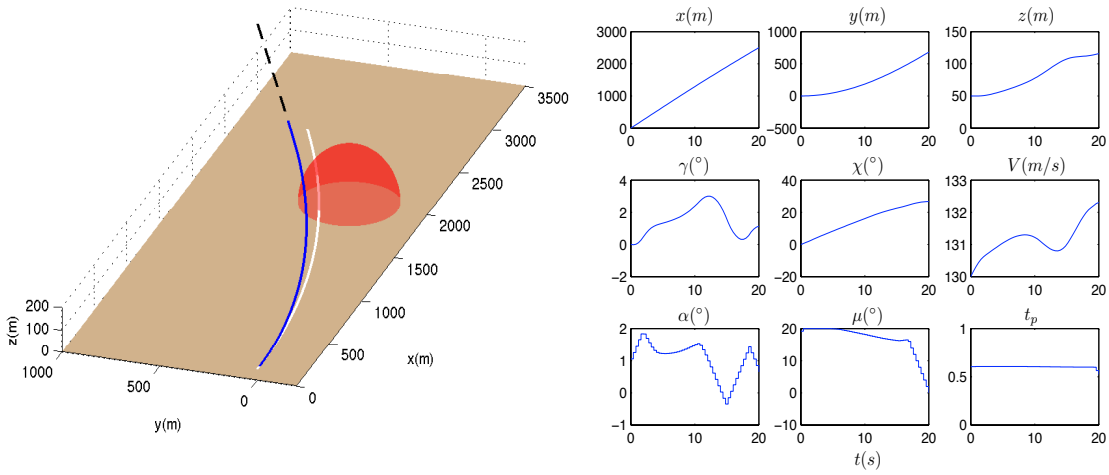


Figure 5.5: Solve time = 0.24s, 13 IPOPT iterations.

5.5.2 Hemispherical and Cylindrical PVs

The same situation as the previous case is considered but with a cylindrical PV of infinite height placed on the projected path of the previous solution. Figure 5.6 shows that the additional end constraints in the optimization have caused a shift in the avoidance trajectory so that the projected path passes to the side of the cylindrical PV. A similar drop in the α profile is encountered as the previous case, again because the solution remains as close to the reference trajectory as possible. The number of iterations required to find the optimal solution in this case has increased resulting in a greater computation time of 0.44s.

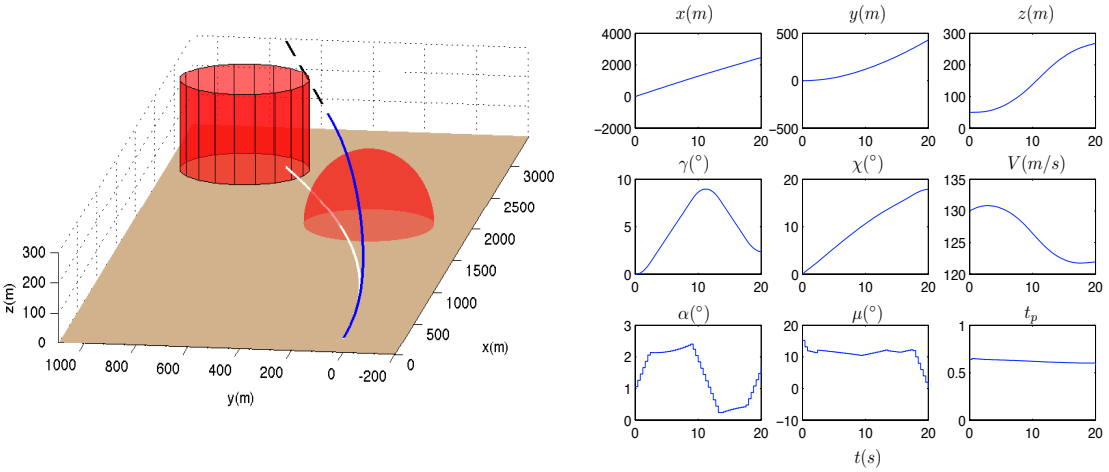


Figure 5.6: Solve time = 0.44s, 21 IPOPT iterations.

5.5.3 Polyhedral PV

Avoidance of a polyhedral PV is shown in Figure 5.7. The additional constraints in the optimization due to the polyhedral obstacle result in a longer computation time per iteration. There are also more iterations before convergence when compared with a hemispherical PV.

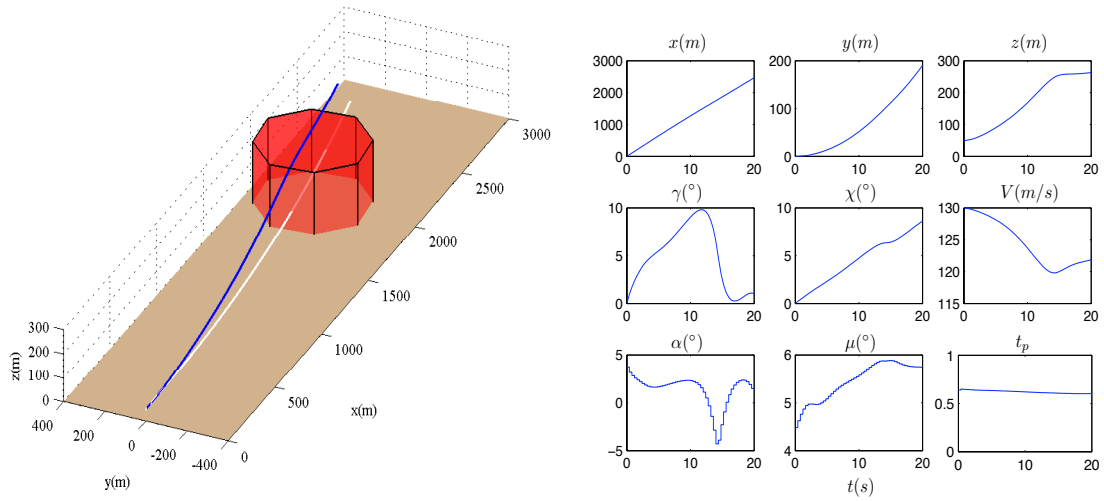


Figure 5.7: Solve time = 0.65s, 27 IPOPT iterations.

5.5.4 Urban Environment Simulation

A number of PVs of varying types are placed in a representative urban environment scenario. Figure 5.8 shows the generated avoidance trajectory which passes above the final cuboidal PV. The algorithm deals with the increased number of constraints and is still able to find a solution within a reasonable time.

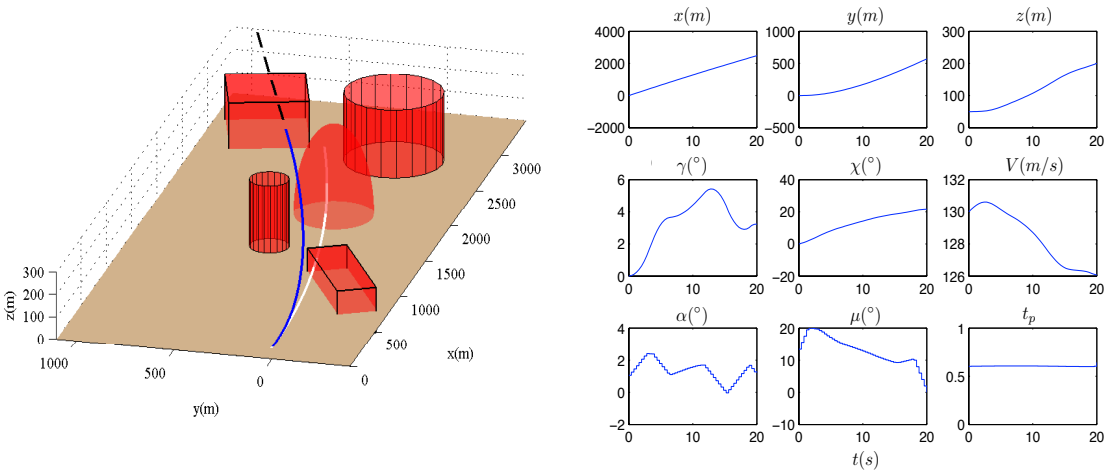


Figure 5.8: Solve time = 0.51s, 22 IPOPT iterations.

5.6 Warm Starting

Warm starting the optimization with different initialization points affects the number of iterations required, and therefore the solution time for producing the avoidance trajectory. In the cases shown previously, the solution was initialized with the reference trajectory and the additional variables $u_N = u_{N-1}$ and $\eta_j = 0$. For the same problem shown in test Figure 5.5, but now zero-initialized, the same solution (within the optimizer tolerance limits) is obtained after 39 iterations with a computation time of 0.68s. This is almost three times the computation time compared to the reference trajectory initialized case and demonstrates the impact the initialization point can have on the efficiency of the optimization.

Assuming the outer loop controller runs once every 1.2s (to allow convergence of the optimization), the affect of initializing with a previous solution on the computation time can be investigated. Figure 5.9 shows a solution at t_0 of $\mathcal{P}(x(t_0), u(t_0))$. A second solution is shown for the forward problem $\mathcal{P}(x(1.2), u(1.2))$ and Table 5.3 shows the computation times for different initialization points. The pre-trigger starting point refers to the case where the override has not been activated and the pilot is assumed to still be following the reference trajectory, i.e. the solution of $\mathcal{P}(x^r(1.2), u^r(1.2))$. The post-trigger case assumes that the avoidance trajectory generated at t_0 is being followed.

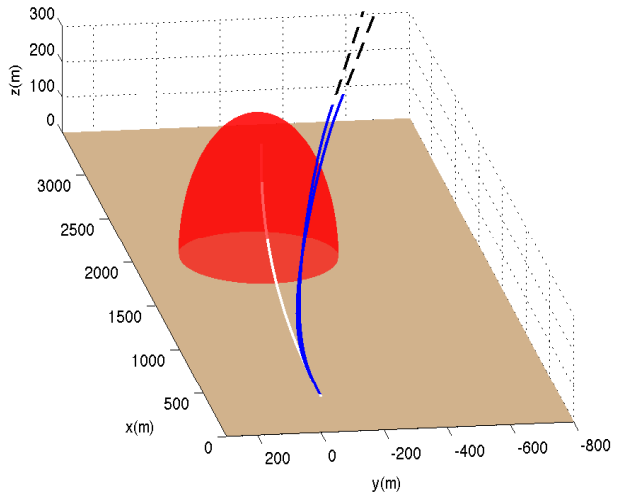


Figure 5.9: Two avoidance trajectories with a 1.2s time interval.

The results show the shift initialization does not necessarily reduce the computation time compared with the reference trajectory initialized case. This is because the optimal point is usually at the boundary of the feasible set, which is not an ideal initialization point for the next problem [34]. Strategies are available for perturbing the initialization point or taking a sub-optimal solution to the original problem as the initializer which is on the interior of the feasible set [96]. Given the computation time is reasonably low in any case, a greater concern

Initialization Point	Starting Point	Solution Time
zero-initalized	pre-trigger	0.78s
reference trajectory initialized	pre-trigger	0.24s
shift-initialized	pre-trigger	0.26s
zero-initialized	post-trigger	0.75s
reference trajectory initialized	post-trigger	0.20s
shift-initialized	post-trigger	0.20s

Table 5.3: Computation times for forward problem.

is the ability of the algorithm to find any feasible solution if one is available. Figure 5.10 shows a scenario where four infinite height cylindrical PVs are present. The gradients of this arrangement cause the reference trajectory initialized problem to fail (not converged after 500 iterations of the solver). In contrast the shift-initialized solution, which has been initialized from previous solutions progressively approaching this initial condition, has solved demonstrating the advantages of shift-initialization for the scheme. In general, cases where the gradients of the PVs produce a local minimum which is still infeasible cannot be dealt with by the trajectory generation scheme alone. The shift-initialization framework provides some guarantee that local infeasible minimums are avoided.

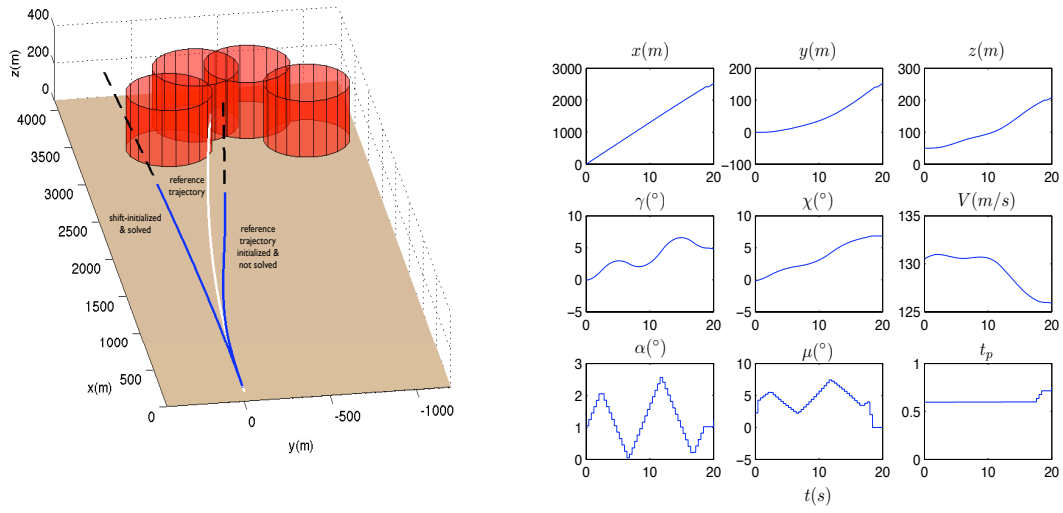


Figure 5.10: Comparison of reference trajectory initialized and shift-initialized problems. The trajectory used for the initialization is shown on the right.

5.7 Extension to Unmanned Aircraft

Given the potential for the trajectory generation strategy to be used in UAV path planning, this class of problems is a natural extension for the work. In broadening the scope to UAV path planning problems, a number of differences to the original aircraft avoidance problem are

encountered:

- In the aircraft PV avoidance problem the final state is free to vary subject to certain constraints. UAV problems are typically defined with a final goal point. The aim is then to reach the goal from a given initial position with some performance criterion e.g. minimizing time or fuel burn, whilst ensuring obstacle constraints are avoided.
- Whilst atmospheric disturbances affect both large aircraft and smaller UAVs, large aircraft are better able to deal with such disturbances within the inner loop controllers available for trajectory following, due to the control actuation available. This is not the case for UAVs, so path planning algorithms need to robustly account for such disturbances.
- UAVs do not necessarily carry the same range of sensors available on civilian aircraft, therefore uncertainty in the vehicle state can arise and needs to be dealt with adequately.
- Avoidance trajectories are currently generated over short time horizons (of the order of 20s). However, a global approach to UAV path planning could require longer horizons to fulfill mission requirements.
- In some UAV missions the environment (position of obstacles, stationary or moving) would not be known a priori. Therefore simultaneous localization and mapping (SLAM) may be required with sensors and perception playing an important role on the behaviour of an autonomous guidance system.

In this chapter the direct multiple shooting trajectory generation scheme is applied to a range of UAV scenarios. It is worth noting that a UAV might not carry the same computational power as a civilian airliner. However, since the multiple shooting strategy exhibits an inherent parallelism, it has a strong case for use in resource constrained platforms. For example, ongoing research is being conducted in the use of field programmable gate arrays (FPGAs) for solving linear systems of equations - the key computationally intensive step in the optimization [81].

5.7.1 UAV Model

Assume that the aircraft state is propagated using the nonlinear Dubins aircraft model [15] which extends the Dubins car [23] to three dimensions. The states of the model are the horizontal position x_c and y_c, altitude z_c and the heading angle χ. The control inputs (u_1, u_2) in the simplified model directly manipulate the vertical velocity $\dot{z}_c$ and the rate of change of heading $\dot{\chi}$. The velocity v is assumed to be constant for simplicity. However, it can be an

additional state if required.

$$\dot{x_c} = v\cos\chi \tag{5.30a}$$

$$\dot{y_c} = v\sin\chi \tag{5.30b}$$

$$\dot{z_c} = u_1 \tag{5.30c}$$

$$\dot{\chi} = u_2. \tag{5.30d}$$

Constraints are applied to the model with $x \in X$ and $u \in U$ to satisfy the manoeuvring envelope of the UAV. The model is lower fidelity than the 3DOF aircraft model used in the PV Avoidance System since autonomous UAVs are commonly commanded using rates of change of height and heading, with in-built inner loop controllers (autopilot) used to achieve the desired rates.

5.7.2 UAV Path Planning Problem Formulation

The discrete time UAV problems can be written and solved in a similar way to the PV avoidance trajectory generation problem of Section 4.2.3. In this section the same notation is employed, with $g_{cont}(\mathbf{x})$ representing the state continuity constraints of the multiple shooting strategy, $\mathcal{C}_j$ representing the obstacle at position r_j, and $p(\cdot)$ a function extracting the position vector from the full state vector. The position vector $p(x)$ is simplified to p as before for notational convenience. The inequality rate constraints $h_{rate}(\mathbf{x}, \mathbf{u})$ are not required for the UAV problem since the controls directly manipulate the rate of change for height and heading ((5.30c), (5.30d)). Therefore bounding $u_i \in U$ is sufficient to constrain these rates.

Fixed Time-to-Target Problem with Static Obstacle Constraints

The fixed time to target control problem can be written as a feasibility problem:

$$\begin{aligned}
&\min_{\mathbf{x},\mathbf{u}} \ 0 \\
\text{subject to: } & g_{cont}(\mathbf{x}) = 0 \\
& \mathbf{x} \in X, \quad \mathbf{u} \in U \\
& p_i \notin \bigcup_{j=1}^{M} \left(\operatorname{int}\mathcal{C}_j \oplus r_j\right), \qquad \forall i \in \{0,\dots,N\}, \forall j \in \{1,\dots,M\} \\
& x_N \in X_N
\end{aligned} \tag{UAV-OCP}$$

where the bound constraints $x_i \in X$ and $u_i \in U$ represent the manoeuvring envelope of the UAV and $x_N \in X_N$ is the desired target set. The system dynamics $\dot{x} = f(x, u)$, which are implicit in the state continuity constraints $g_{cont}(\mathbf{x}) = 0$, come from (5.30). As in Section 4.2.4, the problem UAV-OCP and the subsequent problem formulations can be composed entirely of twice continuously differentiable functions, thereby allowing the use of a smooth optimization

algorithm.

Time-Optimal Problem with Static Obstacle Constraints

The problem can be posed by using slack variables in the following way. Define two additional slack variables $s := (s_0, \ldots, s_N)$ and $t := (t_0, \ldots, t_N)$ corresponding to scalar variables at each multiple shooting interval. Referring to Figure 5.11, the aim of introducing the slack variables is to identify the first node inside the target set $\mathcal{D}$. The constraints are formulated in such a way as to induce the variable s_i to equal one at this node. The second variable t_i is induced to be one for subsequent nodes in the trajectory, and time-optimality can be achieved by maximizing the sum of the elements of t_i. Given a target set,

$$\mathcal{D} = \{x_i \mid h(x_i) \leq 1\}, \tag{5.31}$$

the following set of constraints can be imposed $(\forall i \in \{0, 1, \ldots, N\})$:

$$h(x_i) \leq 1 + K(1 - s_i) \tag{5.32a}$$

$$s_i \geq 0 \tag{5.32b}$$

$$\|s\|_\infty \geq 1 \tag{5.32c}$$

$$0 \leq t_i \leq 1 \tag{5.32d}$$

$$t_i \leq 1 - s_i \tag{5.32e}$$

$$t_i \leq t_{i+1} \tag{5.32f}$$

where K is a large constant (it is selected to be large enough such that if $s_i \neq 1$ (5.32a) is always satisfied). The constraint (5.32a) pushes the variables s_i to zero, due to the large gradient in the negative s direction. However the non-convex constraint (5.32c) ensures that at least one slack variable along the trajectory is equal to one, and therefore x_i satisfies $h(x_i) \leq 1$ and the trajectory passes through the target region. As stated, time-optimality is achieved by introducing the second variable t_i. As Figure 5.11 shows, $t_i = 1$ from the end of the trajectory, working backwards until the first point that enters the target. At this point $t_i = 0$ due to (5.32e) and is subsequently zero up until the start of the trajectory due to (5.32f). The point of entry is forced to be as early in the trajectory as possible by implementing a cost function to maximize the sum of the elements of t_i (see UAV-TOCP below). The time-optimal control

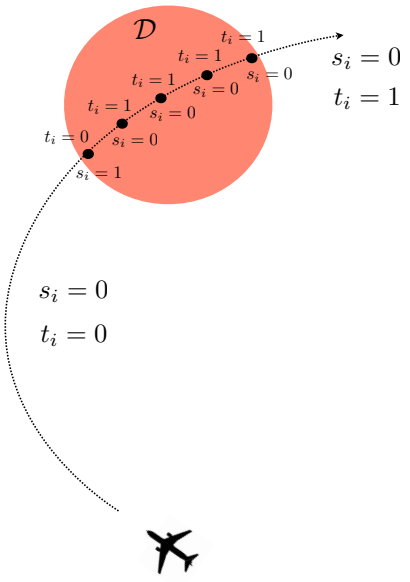


Figure 5.11: Time-optimal problem formulation.

problem is

$$\max_{x,u,t,s} \sum_{i=0}^{N} t_i$$

subject to: (5.32)

$$g_{cont}(\mathbf{x}) = 0 \tag{UAV-TOCP}$$

$$\mathbf{x} \in X, \quad \mathbf{u} \in U$$

$$p_i \notin \bigcup_{j=1}^{M} (\operatorname{int} \mathcal{C}_j \oplus r_j), \qquad \forall i \in \{0, \ldots, N\}, \forall j \in \{1, \ldots, M\}.$$

As described in Remark 6 of Chapter 4, the non-differentiable constraint of (5.32c) can be rewritten in differentiable form at the cost of some extra variables.

Time-Optimal Problem with Moving Obstacle Constraints

The time-optimal problem with moving obstacle constraints is useful for maintaining safe separation for UAVs operating in crowded environments. In this case the trajectory of the obstacle (which could be another UAV) is assumed to be known. The problem formulation follows from the preceding section, but now the position r_j of the obstacle constraints have a

time varying component. The modified time-optimal control problem is

$$\max_{x,u,t,s} \sum_{i=0}^{N} t_i$$

subject to: (5.32)

$$g_{cont}(\mathbf{x}) = 0 \qquad \text{(UAV-TOCP2)}$$

$$\mathbf{x} \in X, \quad \mathbf{u} \in U$$

$$p_i \notin \bigcup_{j=1}^{M} (\operatorname{int} \mathcal{C}_j \oplus r_{i,j}), \qquad \forall i \in \{1, \ldots, N\}, \forall j \in \{1, \ldots, M\}.$$

The time varying obstacles are assumed to be other UAVs with known trajectories. A UAV reference trajectory can be simulated by forward integration of the dynamics (5.30) with a known input sequence $\mathbf{u}$ from an initial state x_0. For a minimum separation distance, the obstacle $\mathcal{C}$ is spherical with a fixed radius. The obstacle can then be modeled as a sphere moving along the predetermined trajectory of the UAV.

5.7.3 Results

The data for the Aerosonde UAV [44] is used to conduct a number of test simulations. A fixed cruising speed v is set at 25 m/s and the maximum turning rate $\dot{\chi}$ and maximum climb rate $\dot{z}_c$ at 6 deg/s and 2 m/s, respectively. The IPOPT algorithm [7] is the interior point solver chosen for the optimization. As with the aircraft avoidance problem, an explicit second order Runge-Kutta integration scheme is used to integrate the system dynamics forward in time. The number of multiple shooting segments is $N = 120$ and the time horizon $t_f = 60s$. Angular states in the model are measured in radians and position states in kilometers. The simulation is started with the UAV at $x_c = 0$ and $y_c = 0$, traveling along the x_c axis. Unless stated, the optimization states and inputs are initialized with the trajectory that the UAV would follow if the initial input $u(t_0)$ is held fixed over the time horizon. This is calculated by integrating forwards in time from the initial condition $x(t_0)$. Where slack variables are present for the time-optimal problems, these are initialized as ones. A ground constraint such that $z > 0, \forall t$ is imposed to prevent the UAV trajectory intersecting the ground. The algorithm is implemented in C++ and simulations performed on a 2.4 GHz Intel iMac with IPOPT as the interior-point solver.

Fixed Time-to-Target Problem with Static Obstacle Constraints

The first problem investigated is that of reaching an end goal when no obstacles are present. Figure 5.12 shows the generated trajectory entering the target set (cuboid). The non-holonomic dynamics can be seen as the trajectory initially turns away from the target before manoeuvring

towards it. The graphs show the four states in the first two rows and the generated input sequence in the final row. It should be noted that the length of the time horizon is fixed and the UAV would enter the target set after 60s. The multiple shooting scheme accomplishes the desired behaviour within a viable solution time for a real-time application. In this example the constraint $x_N \in X_N$ is applied with the heading χ unbounded. However, a desired final orientation can be achieved by constraining χ. The same problem is shown again in Figure 5.13

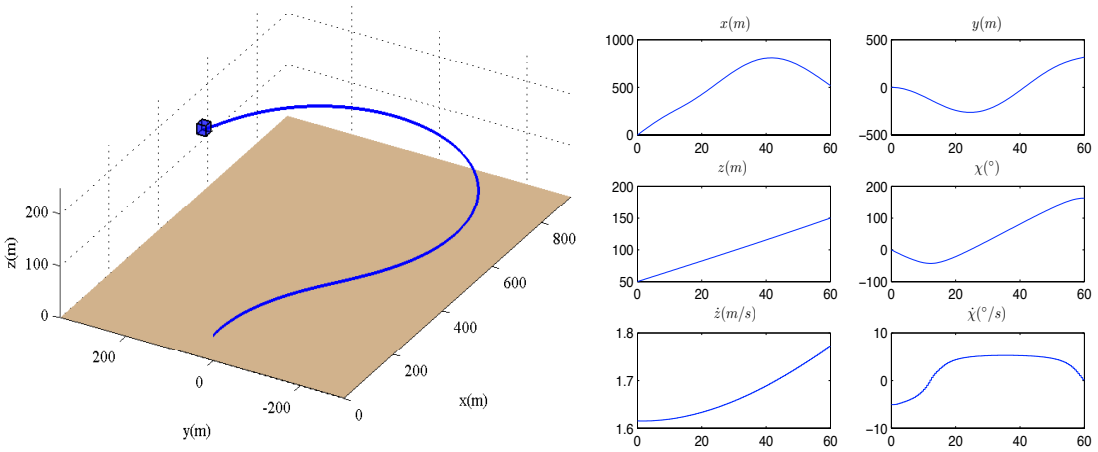


Figure 5.12: Fixed time-to-target problem. Solve time = 0.25s.

with static obstacles placed in the path of the UAV. The scheme generates an adjusted route to compensate for the obstacles. However, the computation time has increased as the size and complexity of the optimization problem increases with the additional constraints.

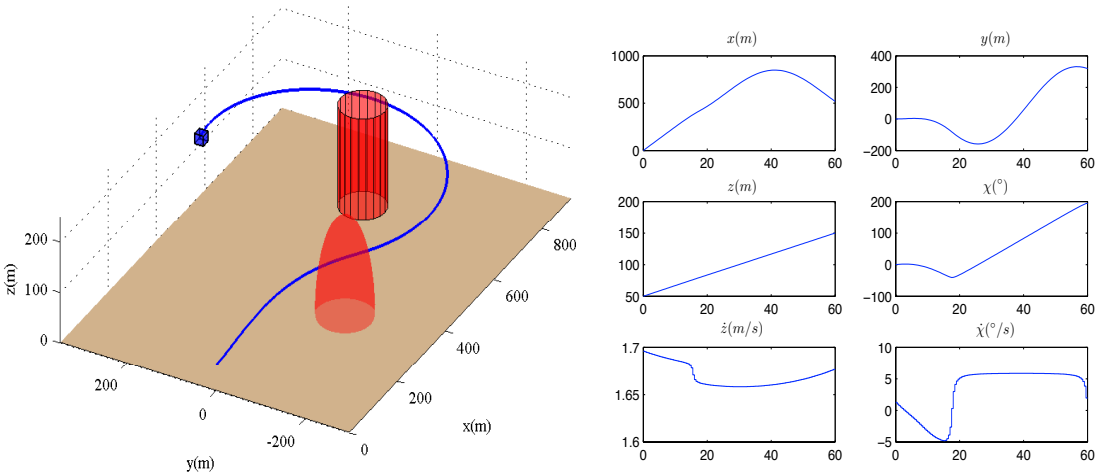


Figure 5.13: Fixed time-to-target problem with obstacle constraints. Solve time = 0.87s.

Time-Optimal Problem with Static Obstacle Constraints

Often a UAV mission profile will require the aircraft to reach a target set in the minimum time required. Again, the case is first simulated with no obstacle constraints as shown in Figure 5.14, with a spherical target set. The time-optimal trajectory is achieved with a turn followed by straight flight, a result that has been proven via the Pontyagin maximum principle in previous work [93]. In practice the final part of the trajectory can be ignored once the UAV has reached the target. The same problem is shown in Figure 5.15 but with one cubic and two ellipsoidal

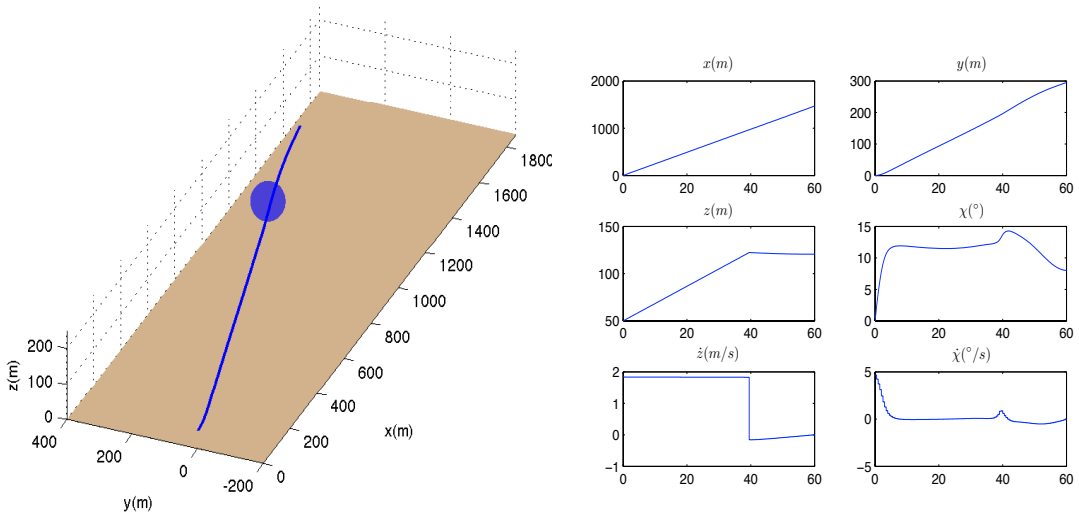


Figure 5.14: Time-optimal problem with no obstacle constraints. Solve time = 0.77s.

obstacles placed between the UAV and the target. The generated trajectory penetrates the target set taking a path around the obstacles. It should be noted that the initialization of the optimization plays an important role in determining whether optimality is achieved or not. In this example, initialization with a straight line trajectory was sufficient. However in different cases locally optimal trajectories might be found due to the inherently non-convex nature of the problem. One strategy can be to run parallel computations with multiple initialization points. Another is to first run the computation with no obstacles giving the most direct target penetration, then initialize with this first trajectory (with slack variables, s_i and t_i, set to one). A set of results demonstrating this can be found in Figure 5.16.

Time-Optimal Problem with Moving Obstacle Constraints

Figures 5.17 and 5.18 display results with moving obstacles. In these simulations the obstacle trajectory is known a priori, and is determined by modeling the obstacle as another UAV with fixed inputs. A separation bubble of fixed radius is then applied.

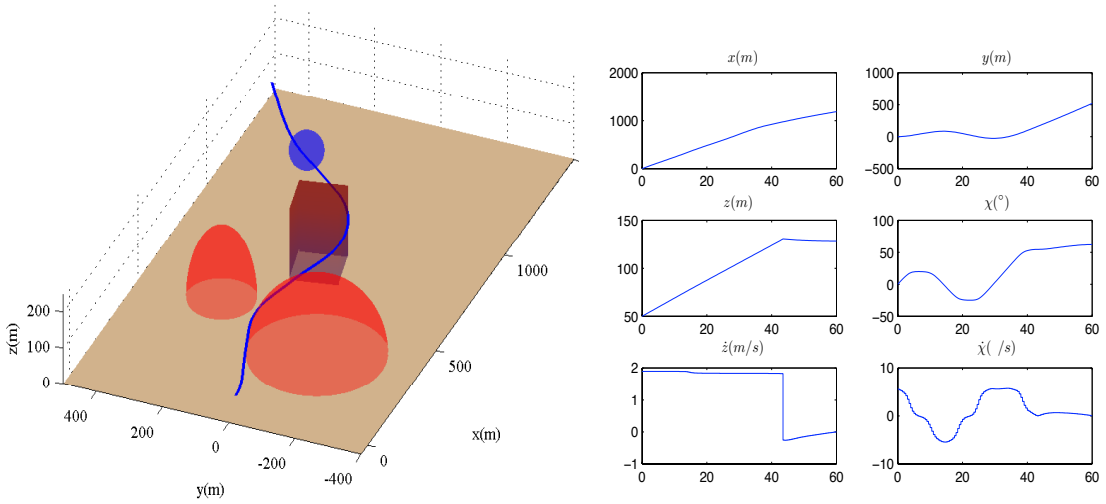


Figure 5.15: Time-optimal problem with obstacle constraints. Solve time = 1.47s.

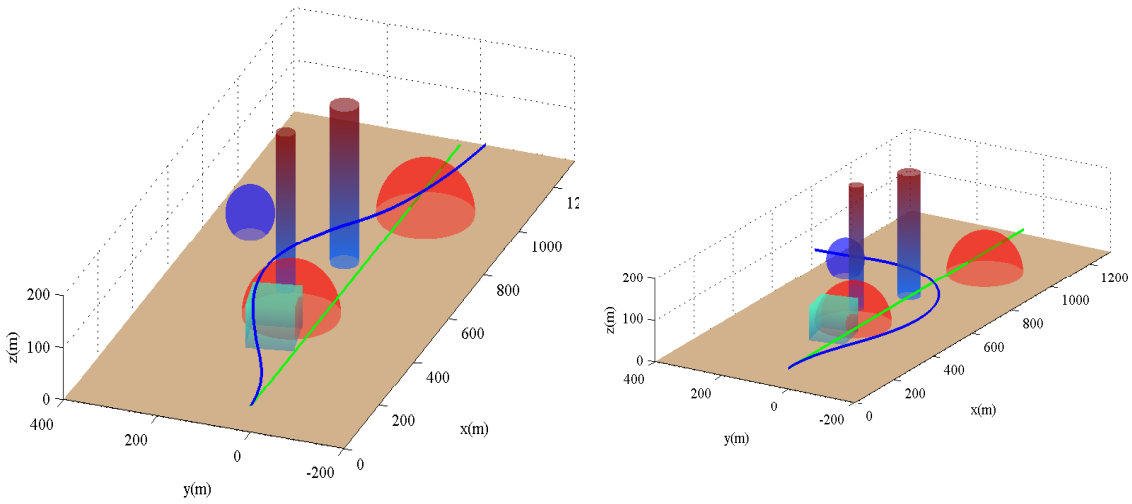


Figure 5.16: Left: initialized with the reference trajectory (shown in green). Right: initialized with the trajectory obtained when the computation is first run with no obstacles.

5.7.4 Conclusions and Future Work in UAV Guidance

The multiple shooting strategy developed for the aircraft avoidance problem has the potential to be used as a short horizon UAV path planner. However, over some threshold time horizon the computation will no longer be viable for real-time applications. A hierarchical controller with multiple levels running at different rates can overcome this problem. The aircraft avoidance controller already contains two levels - the outer loop trajectory generator and the inner loop trajectory following PID controllers. A third deliberate layer can be added above the first two, generating waypoints over long time horizons with the lower layers optimally moving between

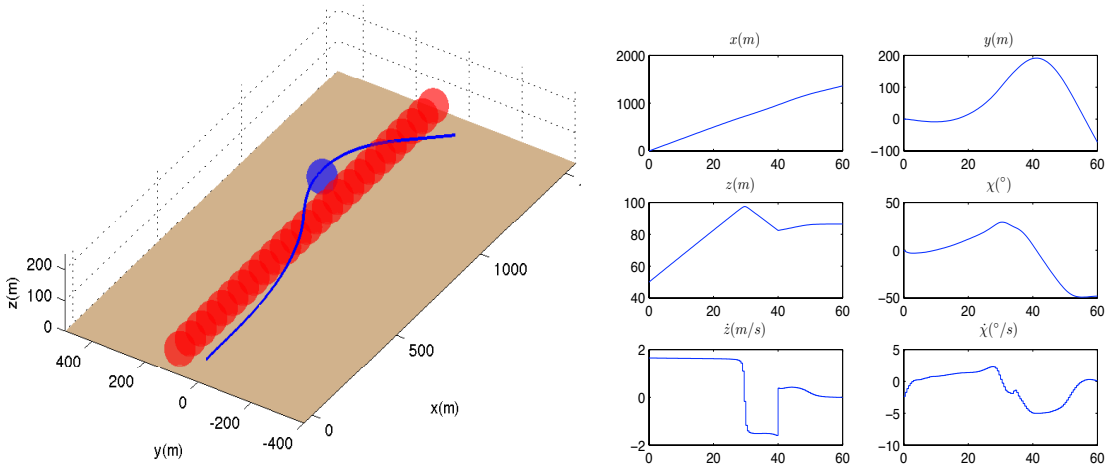


Figure 5.17: Time-optimal problem with a moving obstacle constraint (50m separation). Solve time = 1.51s.

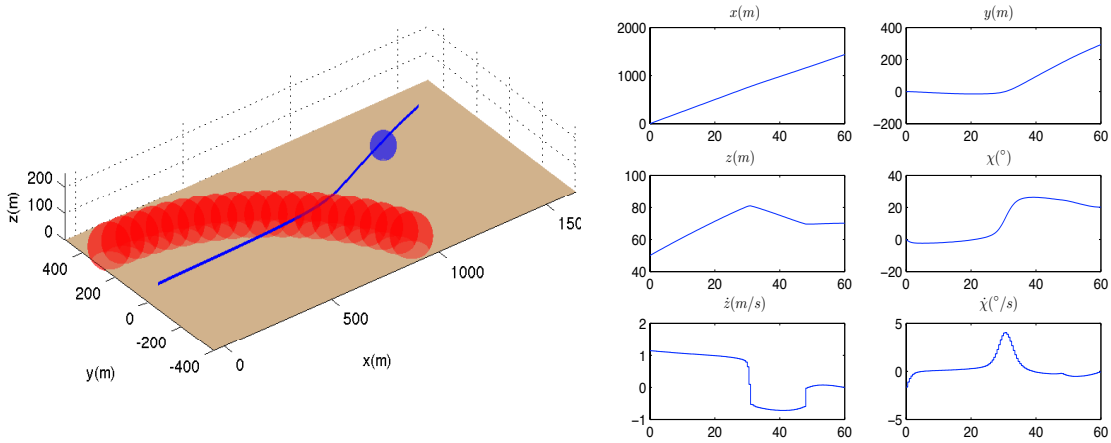


Figure 5.18: Time-optimal problem with a moving obstacle constraint (80m separation). Solve time = 1.45s.

waypoints and avoiding obstacles. A number of schemes are already available for this type of waypoint generation which would typically use a lower fidelity vehicle model, for example RRT (rapidly expanding random tree) methods with a Dijkstra step to improve optimality [58].

A proposed global approach to autonomous control of aircraft is presented in Figure 5.19. This approach allows known obstacles to be dealt with by the long term planner, whilst the higher rate trajectory generation scheme can deal with obstacles that 'appear' in the path plan. Blending the current trajectory generation scheme with a long term planner and demonstrating an effective implementation could form a large body of future work. Communication between the layers of hierarchical controllers of this type has received little attention to date and the

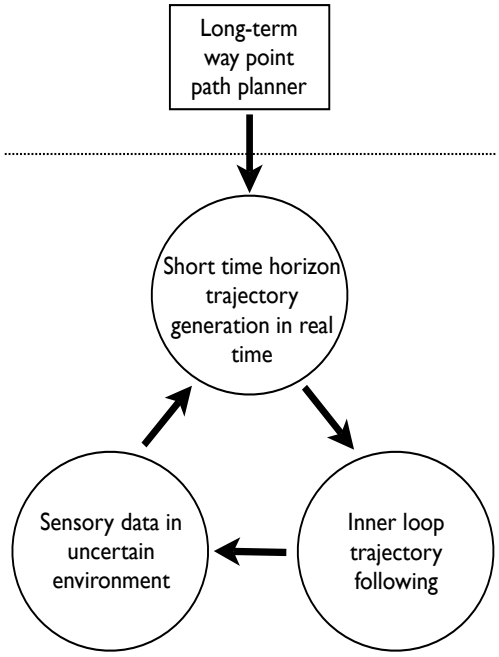


Figure 5.19: Autonomous control scheme for unmanned vehicles.

aim would be to develop a robust algorithm with flexibility for use with differing UAV models.

Work can also be conducted on the application of the control scheme in an uncertain environment. Placing unknown obstacles along the original path of the vehicle, the reactive element of the trajectory generation strategy can be tested. Using the exact penalty function and soft constraints in the work described on trigger mechanisms in the following chapter, for example, the final state (at a fixed waypoint with a given speed and orientation) for the shorter horizon computation can be implemented as a soft constraint in the optimization. This would heavily penalize deviations away from this final state, but still produce a solution if an unknown obstacle appeared close to or on a waypoint, which would otherwise result in infeasibility of the hard constrained problem.

There are situations where a trajectory might not be found via optimization, for example if an object appeared very close to the UAV, hence purely reactive methods (e.g. potential field) can also be incorporated as a fail-safe.

Chapter 6

Override Trigger Mechanism

This chapter focuses on the *trigger of pilot override* part of the system, referred to as the 'trigger mechanism' (see Figure 1.3). In determining a suitable trigger mechanism for an aircraft override system, a balance needs to be found between ensuring a sufficient margin to allow the control system to fly a safe avoidance trajectory and not excessively constricting the airspace in which pilots are able to fly. There are situations, for example the Hudson river landing in January 2009 [4], where in extraordinary circumstances an aircraft might need to fly close to prohibited volumes to perform an emergency landing. In these situations it is desirable to have a trigger mechanism that allows the pilot to retain control until the threat is deemed too serious too ignore. The system works on the principle that once the trigger has been activated, the last avoidance trajectory generated will be flown by the *trajectory-following* inner loop controllers.

6.1 Problem Formulation and Relationship to Trajectory Generation Outer Loop Controller

Recall from Figure 1.3 that the outer loop controller runs continuously in the background as the pilot flies the aircraft. Once the trigger mechanism is activated at time t_0 after a threat of PV violation is detected, control is removed from the pilot and the inner loop controller flies the sequence $\mathbf{x}^*$ from the solution of $\mathcal{P}(x(t_0 - t_d), u(t_0 - t_d))$ where t_d is the time delay between the optimization calculation start time and trigger activation.

In this chapter the aim is to determine when an override parameter τ should be switched from zero to one. If $\tau = 0$ (as default), the pilot remains in control of the aircraft. When $\tau = 1$ the pilot's control input is ignored and the aircraft is manoeuvred via the PV Avoidance System inner loop controllers. To ensure an avoidance trajectory is available for the inner loop controllers to follow, τ needs to be switched to one *before* the trajectory generation optimization problem OCP2 becomes infeasible (i.e. before there is no longer a control sequence available to

avoid the PV), or formally, before

$$\mathcal{X}_N(x, u) = \emptyset, \tag{6.1}$$

where X_N is the feasible set for the trajectory generation problem. The following sections investigate three different methods for achieving this. Numerical results for the trigger mechanisms can be found in Section 6.5.

6.2 Two-Solution Method

Assuming an avoidance trajectory has been generated by the optimization method outlined in Chapter 4 at time t_0, the aircraft state can be predicted at some time t in the future in much the same way as the reference trajectory is generated, by taking the initial state $x(t_0)$ of the aircraft and integrating forward in time assuming the input $u(t_0)$ is held constant. The simplest form of trigger mechanism is derived by attempting to solve the optimization again at a future initial state (see Figure 6.1).

At a given time t_0 the problems $\mathcal{P}(x(t_0), u(t_0))$ and $\mathcal{P}(x_1^r, u_1^r)$ are solved simultaneously, where x_1^r and u_1^r are the predicted states and inputs at a future time t_1. If either

$$\mathcal{X}_N(x(t_0), u(t_0)) = \emptyset \quad \text{or} \tag{6.2}$$

$$\mathcal{X}_N(x_1^r, u_1^r) = \emptyset, \tag{6.3}$$

the override is triggered ($\tau : 0 \rightarrow 1$). Normally the forward problem (i.e. the problem with initial state and input x_1,u_1) is expected to fail first. The lookahead time t_1 determines the 'conservativeness' of the override system since a longer look ahead time will result in earlier override of the pilot controls.

6.3 Exact Penalty Function Method

With this trigger mechanism strategy an additional term is added to the objective function of the trajectory generation problem OCP2 in order to find whether an avoidance trajectory can be generated, even if the pilot input is held at the current value for a set amount of time (i.e. it is not immediately necessary to change the control input to ensure avoidance of a PV). The exact penalty function allows this constraint to be implemented as a 'soft' constraint, meaning the constraint can still be violated if this is required for convergence to a solution (as a 'hard' constraint the problem would become infeasible). Once this violation is detected, the trigger can be activated and there is still a feasible avoidance trajectory available to follow.

More generally, it is possible to implement all the constraints as soft constraints if required, which allows convergence to a solution in all cases.

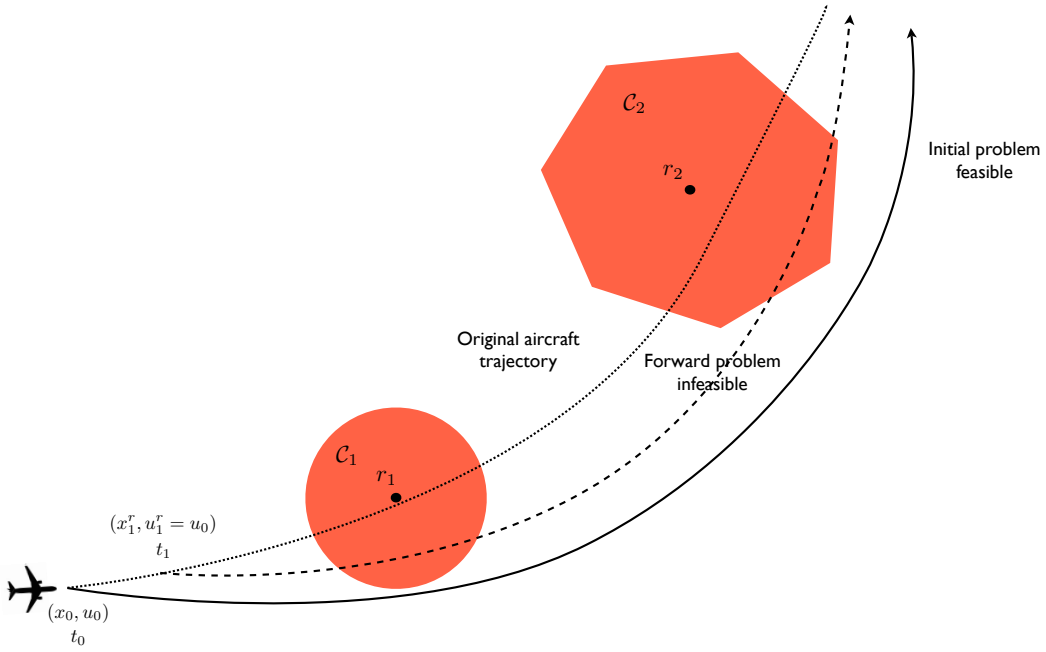


Figure 6.1: Two solution method trigger mechanism.

6.3.1 Exact Penalty for a General Optimization Problem

The exact penalty function was first conceived as a means for converting a constrained optimization into an unconstrained optimization [29] and its use for MPC problems has been investigated in [50]. Consider the general optimization problem

$$\begin{aligned} &\min_z \ J(z) \\ \text{subject to: } &g_i(z) = 0, \quad \forall i \in \mathcal{E} \\ &h_i(z) \leq 0, \quad \forall i \in \mathcal{I}. \end{aligned} \tag{OP}$$

The exact penalty function $\psi : \mathbb{R}^n \to \mathbb{R}$ is defined in terms of the original objective function J, the equality constraints g_i and inequality constraints h_i. Under certain conditions the local minimizer z^* of $\psi(\cdot)$ is the solution to the original constrained problem. An L_1 penalty function is given by

$$\psi(z) = J(z) + \rho\Big(\sum_{i\in\mathcal{E}} |g_i(z)| + \sum_{i\in\mathcal{I}} [h_i(z)]^-\Big), \tag{6.4}$$

where $[h_i(z)]^- \triangleq \max\{0, -h_i(z)\}$. The parameter ρ ($\rho > 0$) provides a means of weighting the relative contribution of $J(z)$ and the penalty terms. It can be shown [29] that if the weighting of the penalty terms is sufficiently large (i.e. $\rho \in [\rho^*, \infty)$ where ρ^* is some threshold value) the penalty incurred by moving into a region where the constraint is no longer satisfied always

dominates the $J(z)$ contribution, creating a local minimum in the modified objective function. Note that the modified objective function ψ for the new problem is non-differentiable. To bypass the discontinuity a modified optimization problem can be formulated:

$$\begin{aligned} \min_{z} \ & \psi(z) = J(z) + \rho\big(||s_1||_1 + ||s_2||_1\big) \\ \text{subject to: } & g_{sc}(z) \leq s_1 - s_2 \\ & s_1 \geq 0 \\ & s_2 \geq 0, \end{aligned} \tag{MOP}$$

where new slack variables s_1 and s_2 have been introduced. The soft constraint $g_{sc}(z)$ is assumed to be an inequality. Note that two variables are used here to ensure differentiability of the objective function. Since it is a 1-norm penalty, if a single variable was used this would require splitting in any case (cf. Remark 6). The constraint here is an inequality, however equality constraints $g_{sc} = s_1 - s_2$ can also be dealt with by the solver. For the problem OP and its reformulation MOP, the optimal value z^* remains the same for both problems whenever OP is feasible. However MOP is still feasible if the constraint $g_{sc}(z) \leq 0$ is violated, which is not the case for OP.

Conservative lower bound for ρ^*

A lower bound on the penalty parameter (which can be conservative) is required to have confidence that the unconstrained minimum of the modified function is equivalent to the constrained minimum. Rosenberg looked at the problem for non-smooth but locally Lipschitz objective functions [82]. The results are presented here in the context of smooth objective functions and a conservative bound is deduced from this result and the specific structure of the multiple shooting problem. First defining the Dini directional derivative $J'(z^*; y)$ of J at the point z^* in the direction y ($\neq 0$) as

$$J'(z^*; y) = \liminf_{\alpha \downarrow 0} \left\{ \frac{J(z^* + \alpha y) - J(z^*)}{\alpha} \right\}, \tag{6.5}$$

with $z^* \in \mathbb{R}^n$, $\alpha > 0$ and $z^* + \alpha y \in \mathbb{R}^n$. The threshold ρ^* is determined by

$$\rho^*(z^*) = \begin{cases} \frac{-\beta(z^*)}{\Gamma(z^*)} & \text{if} \quad J'(z^*; y) \leq 0, \|y\|_2 \leq 1 \\ 0 & \text{otherwise.} \end{cases} \tag{6.6}$$

The threshold condition of (6.6) states that z^* is already a local minimum if the gradients in all directions from z^* are positive, therefore $\rho^* = 0$. In this case no penalty parameter is required as there are no active constraints. The terms $\beta(z^*)$ and $\Gamma(z^*)$ are associated with the gradients

of the objective and constraint functions respectively:

$$\beta(z^*) \triangleq \min_y \{J'(z^*; y) \mid \|y\|_2 = 1\} \tag{6.7}$$

where and

$$\Gamma(z^*) \triangleq \min_y \left\{ \sum_{h_i(z^*)=0} \max(0, h_i'(z^*; y)) + \sum_i |g_i'(z^*; y)| \right\}. \tag{6.8}$$

The result stated here provides a tight lower bound on the threshold value of the penalty parameter. The minimizer z^* is assumed to be known, however in practice this would not be the case.

6.3.2 Exact Penalty for the Aircraft-Specific Problem

The bound on ρ^* must consider all constraints in the optimization, those which are strict hard constraints ($g_i(z)$, $h_i(z)$) in the problem and also the soft constraints ($g_{sc}(z)$).

Proposition 4. *A conservative threshold bound for an exact penalty function in a multiple shooting problem with a quadratic cost function can be computed from*

$$\rho^* = \frac{\bar{\sigma}_Q N\bar{x} + \bar{\sigma}_R N\bar{u} + \bar{\sigma}_P \bar{x}}{\inf_{\|y\|_2 \leq 1} \left\{ \sum_i |g_i'(z^*; y)| \right\}}, \tag{6.9}$$

where $\bar{\sigma}_Q$, $\bar{\sigma}_R$ and $\bar{\sigma}_P$ are the maximum singular values of the Q, R and P cost matrices respectively.

Proof. The objective function for the multiple shooting problem (4.9) can be written

$$J(z) = z^T H z \tag{6.10}$$

with $H \triangleq \text{diag}\{Q_0, R_0, Q_1, R_1, \ldots Q_{N-1}, R_{N-1}, P\}$. The gradient of the objective function in the direction y is

$$J'(z; y) = z^T H y, \tag{6.11}$$

and the maximum gradient can be found by calculating

$$\sup_{z \in Z} \{ \max_{\|y\|_2 \leq 1} J'(z; y) \} \tag{6.12}$$

$$= \sup_{z \in Z} \{ \|z^T H\|_2 \} \tag{6.13}$$

$$\leq \|H\|_2 \sup_{z \in Z} (\|z\|_2). \tag{6.14}$$

Defining $\bar{u} = \sup\{\|u\|_2 \mid u \in U\}$ and $\bar{x} = \sup\{\|x\|_2 \mid x \in X\}$, (6.14) is itself bounded by

$$\bar{\sigma}_Q N\bar{x} + \bar{\sigma}_R N\bar{u} + \bar{\sigma}_P\bar{x}. \tag{6.15}$$

Therefore a conservative upper bound on $-\beta(z^*)$ is

$$-\beta(z^*) \leq (\bar{\sigma}_Q N\bar{x} + \bar{\sigma}_R N\bar{u} + \bar{\sigma}_P\bar{x}). \tag{6.16}$$

A conservative lower bound for the denominator $\Gamma(z^*)$ can now be found. The gradients of the inequality constraints $h_i'(z^*; y)$ only appear in (6.8) when there are active inequality constraints. Dropping the inequality constraint terms in (6.8) gives

$$\Gamma(z^*) \geq \inf_{\|y\|_2 \leq 1} \left\{ \sum_i |g_i'(z^*; y)| \right\}, \tag{6.17}$$

where z^* is the local minimizer. From (6.6), (6.16) and (6.17) the conservative threshold is determined by

$$\rho^* = \frac{\bar{\sigma}_Q N\bar{x} + \bar{\sigma}_R N\bar{u} + \bar{\sigma}_P\bar{x}}{\inf_{\|y\|_2 \leq 1} \left\{ \sum_i |g_i'(z^*; y)| \right\}}. \tag{6.18}$$

□

The denominator of (6.18) is still awkward for the case where it is equal to zero. However, for the trigger mechanism application the value is computable and is found in the next section.

Trigger using soft constraints

In the trajectory generation algorithm presented in Chapter 4, the control input at the first time step u_0 is free to vary during the optimization within the rate constraints defined. However, using the exact penalty function to apply a soft constraint $g_{sc}(z) = u_0 - u(t_0) = 0$, forces the trajectory generation algorithm to select $u_0 = u(t_0)$ i.e. forces the input over the first time step to be the pilot's current selection. Note that all other constraints remain as hard constraints in OCP2.

The bound of (6.17) can be evaluated by only including the new equality constraint, bypassing the more difficult problem of evaluating bounds relating to sensitivity terms in the other state continuity equality constraints (4.5) of OCP2:

$$\Gamma(z*) \geq \min \left\{ \sum_i |g_i'(z^*; y)| \right\} \tag{6.19}$$

$$\geq |g_{sc}'(z^*; y)| = \|Iy\| = 1, \tag{6.20}$$

since $g'_{sc}(z^*; y) = I$ and $\|y\|_2 \leq 1$. Therefore the conservative threshold value is calculated with

$$\rho^* = \bar{\sigma}_Q N\bar{x} + \bar{\sigma}_R N\bar{u} + \bar{\sigma}_P\bar{x}, \tag{6.21}$$

which is computable.

When the algorithm can no longer satisfy $g_{sc}(z)$, because doing so will cause a violation of the other hard constraints, the solution will move away from this constraint and incur a large penalty in the objective function. The soft constraint violation can be detected and used to trigger the override. In formal terms, the modified feasible set is now

$$\begin{aligned}\mathcal{X}_N^{\mathcal{I}} = \{(x,u) \mid &\text{OCP2 is feasible with } x = x(t_0),\\ &u = u(t_0),\, u_0 = u(t_0) \text{ and horizon } N\}.\end{aligned} \tag{6.22}$$

In this case the 'conservativeness' of the trigger can be adjusted by modifying the number of time steps the generated control input should match the reference input $u(t_0)$. A longer matching period will result in earlier override of the pilot controls. This method is comparable to the previous two solution method of Section 6.2, where in effect the pilot input is held constant and feasibility is checked at a future point (horizon lengths differ however). It should be noted that this trigger mechanism only requires the solution of a single optimization as opposed to the previous method which requires two parallel computations. Whilst two parallel computations are feasible, the size of the problem is hardly altered through the addition of the soft constraint. In addition, the trajectory generation and trigger mechanism are bundled into a single computation in this case.

Remark 8. *The analysis described here has produced a conservative threshold value for the exact penalty function. An alternative approach is to determine the threshold heuristically by observing values of the Lagrange multipliers for various problems as the algorithm is run in simulation, and take the overall maximum observed value plus some safety factor as the threshold. If a particular situation results in the threshold value still being too small, the penalty function would no longer ensure the constraint was matched as a movement away from the constraint might produce a lower objective function. In this case the soft constraint is violated earlier than is strictly necessary producing a more conservative trigger mechanism - which can still be assumed to be 'safe'.*

6.4 PV Entry Method

The two previous mechanism proposals will only trigger when the projected trajectory of the aircraft directly penetrates a PV. If this is not the case, the reference trajectory itself is a feasible solution to the trajectory generation optimization. This allows the pilot to fly close

to a PV without the override being triggered, as shown in Figure 6.2. A pilot can therefore follow a path close to the PV and then apply an extreme control input to enter it. Time delays in the system (including the computation time for the avoidance trajectory) might result in the PV being penetrated. Testing of the system with flight simulation software shows this

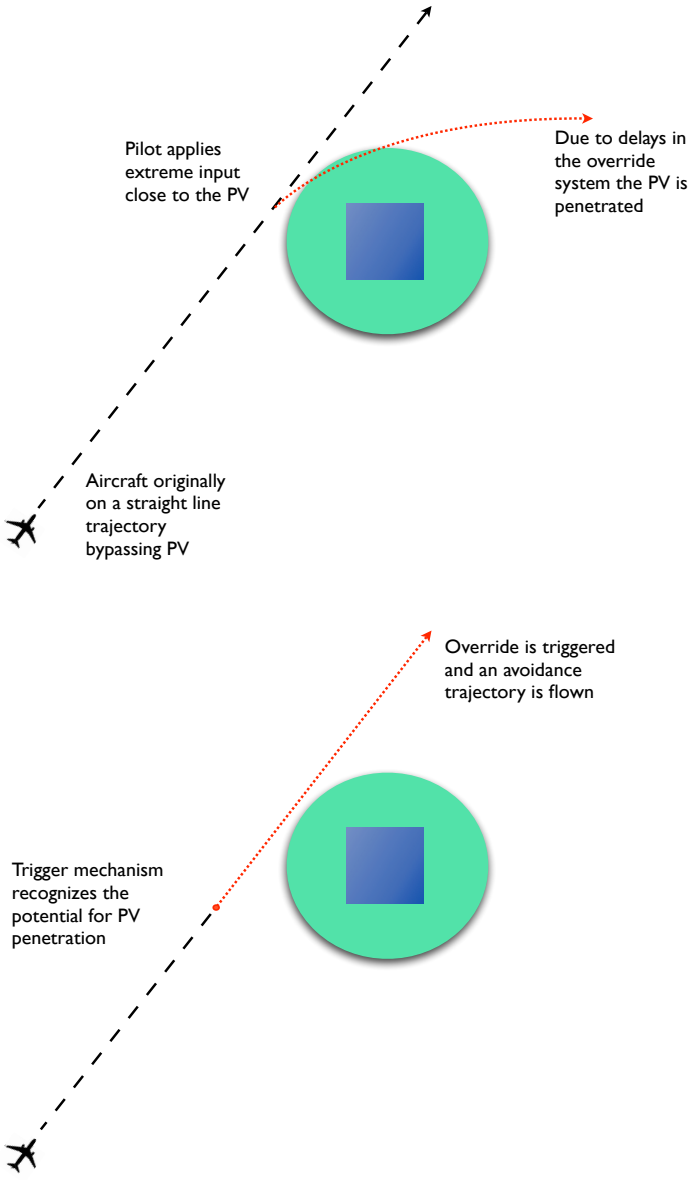


Figure 6.2: Top - pilot applies extreme input and penetrates the PV. Bottom - override activated before PV penetration can occur.

to be possible. One solution for this problem is to define larger PVs with appropriate safety factors calculated. Another solution is to design a trigger based on determining whether a control sequence exists to penetrate a PV from the current aircraft state over a short look ahead horizon (N' multiple shooting segments). In this case the same trajectory generation

algorithm described in Chapter 4 can be used, but reformulated such that the goal is now to find a control sequence $\mathbf{u}$ that results in *any* part of the trajectory passing through a PV. This can be modeled as the set inclusion condition

$$x_i \in \Big(\bigcup_{j=0}^{M} (\operatorname{int} \mathcal{C}_j \oplus r_j) \Big), \quad \text{for one or more } i \in \{0, 1, \ldots, N'\}. \tag{6.23}$$

For the entry problem an assumption that the pilot is able to perform extreme manoeuvres is made and the bounding sets U and X, defined by (3.2) and (3.3) representing the manoeuvring envelope of the aircraft, are expanded accordingly.

One method of constructing the trigger is to solve N' or fewer problems with different horizons and determine if any converge to a solution with the final state $x_{N'}$ constrained to be inside the PV. This would be required for each PV in the local area, and would lead to excessive computations particularly in areas where many PVs are present. For example if N' different horizon lengths are tested to determine whether each end point can be inside a PV, then the total number of optimizations required would be $N' \times$ no. of PVs. An alternative is to construct a slightly different problem, as described below, where only one computation is required for each PV in the local area. The computations for each PV can then be run simultaneously.

The problem can be posed by using auxiliary variables in the following way. Define a vector of auxiliary variables $s := (s_0, \ldots, s_{N'})$ corresponding to a single variable at each multiple shooting interval. Given a prohibited volume *target* constraint,

$$\mathcal{C} = \{x_i \mid h(x_i) \leq 1\}, \quad \text{for one or more } i \in \{0, 1, \ldots, N'\}, \tag{6.24}$$

the following set of constraints can be imposed:

$$h(x_i) \leq 1 + K(1 - s_i), \quad \forall i \in \{0, 1, \ldots, N'\}, \tag{6.25a}$$

$$s_i \geq 0, \quad \forall i \in \{0, 1, \ldots, N'\}, \tag{6.25b}$$

$$\|s\|_\infty \geq 1, \tag{6.25c}$$

where K is some large constant. If $s_i = 0$, then (6.25a) is guaranteed to be satisfied given a sufficiently large K. The non-convex constraint (6.25c) ensures that at least one auxiliary variable along the trajectory is equal to one, and therefore x_i satisfies $h(x_i) \leq 1$ and the

trajectory passes through the PV. The modified optimal control problem is

$$\begin{aligned} &\min_{\mathbf{x},\mathbf{u}} \ 0 \\ &\text{subject to: (6.25)} \\ &\qquad g_{cont}(\mathbf{x}) = 0 \\ &\qquad h_{rate}(\mathbf{x},\mathbf{u}) \leq 0 \\ &\qquad \mathbf{x} \in X, \quad \mathbf{u} \in U. \end{aligned} \tag{MOCP}$$

The objective function is zero resulting in a feasibility problem. If the problem is infeasible PV penetration is not possible from the current aircraft state. Noting that MOCP is only valid for a single PV, multiple problems are run in parallel for each PV in the scope. Defining the feasible solution set as

$$\begin{aligned} \mathcal{X}_{\mathcal{N}'}^{\mathcal{IT}} = \{(x,u) \mid &\text{MOCP is feasible with} \\ &x = x(t_0),\, u = u(t_0) \text{ and horizon } N'\}. \end{aligned} \tag{6.26}$$

the trigger ($\tau : 0 \to 1$) is activated when

$$\mathcal{X}_{\mathcal{N}'}^{\mathcal{IT}}(x(t_0), u(t_0)) \neq \emptyset, \tag{6.27}$$

i.e. if any of the computations (one for each PV) produce a solution. As the time horizon over which the test is performed increases, the trigger mechanism becomes more conservative.

Remark 9. *The non-differentiable constraint of* (6.25c) *can be rewritten in a differentiable form at the cost of a few additional variables as explained in Remark 6 of Chapter 4.*

6.5 Trigger Mechanism Results

6.5.1 Two-Solution Method

Figure 6.3 shows the two problems $\mathcal{P}(x(t_0), u(t_0))$ and $\mathcal{P}(x^r(2), u^r(2))$ with an ellipsoidal PV placed in the aircraft's reference path. The first problem has converged to a solution whilst the forward problem with a look ahead time of 2s has failed to converge ($\mathcal{X}_N(x^r(2), u^r(2) = \emptyset)$). The solution time of 1.1s is for the two problems solved sequentially. However, when solved in parallel, the maximum solution time is a function of the maximum number of optimization iterations allowed for convergence in a single problem and is on the order of 0.5s for the hardware described. For the complete PV Avoidance System this would result in the override being triggered, and the avoidance trajectory generated by the solved problem would be followed.

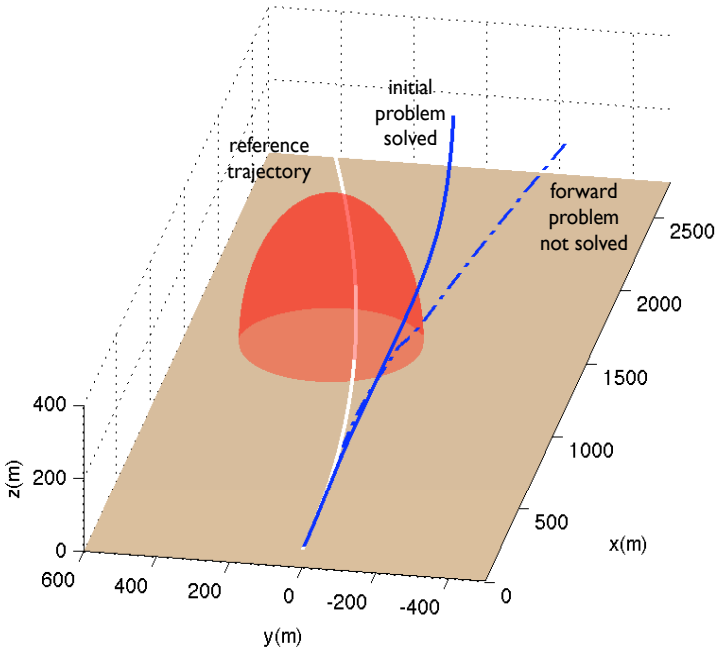


Figure 6.3: Two solution trigger method. Solve time = 1.1s.

6.5.2 Exact Penalty Function Method

By defining a conservative value of $\bar{x} = \|x\|_2$ as the maximum distance the aircraft can reach in the 20s horizon to be 5km and similarly $\bar{u} = \|u\|_2$ as a maximum change in roll angle from $-90°$ to $90°$ a threshold $\rho^* = 412$ is obtained for (6.21). Applying this threshold, Table 6.1 gives results for increasingly difficult problems as the aircraft directly approaches a single ellipsoidal PV.

In the first problem of Table 6.1 the aircraft is placed at a distance of 2000m from the centre of the PV. The third column indicates whether the input constraint $g_{sc} = u_0 - u(t_0) = 0$ is satisfied. As the constraints become more difficult to satisfy by moving the initial position of the aircraft to be closer to the PV, the objective function value increases as expected. At 1200m distance there is a sharp increase, corresponding to the input constraint g_{sc} no longer being satisfied, i.e. the computation can no longer hold the input at the pilot's current choice over the first time step without violating other constraints. In the final case at 1000m distance a solution cannot be found. The column of maximum multipliers shows the values $\max_i\{|\lambda_i^*|\}$ for the equivalent hard constrained problem. The threshold $\rho^* = 412$ comfortably includes the largest of these.

Using this method the trigger would be activated when a mismatch between u_0 and $u(t_0)$ is detected. The advantage over the two solution method is that only a single optimization is computed. More generally if all constraints are implemented as soft constraints, a solution can always be found as there is no longer an infeasible region in the solution space. On the other hand, the disadvantage is that constraints *can* be violated which is unacceptable in some

distance from centre of PV (m)	objective function value	input constraint satisfied	solved	maximum multiplier
2000	0.204	yes	yes	11.72
1800	0.280	yes	yes	25.79
1600	0.412	yes	yes	32.74
1400	0.830	yes	yes	60.32
1200	3.597	no	yes	118.59
1000	n/a	no	no	n/a

Table 6.1: Results for increasingly difficult problems as the aircraft closes on a PV.

problems.

6.5.3 PV Entry Method

Implementing the reformulated problem described in Section 6.4, Figure 6.4 shows the trajectory generated for a 10s time horizon. The plots to the right of the trajectory graph show the six

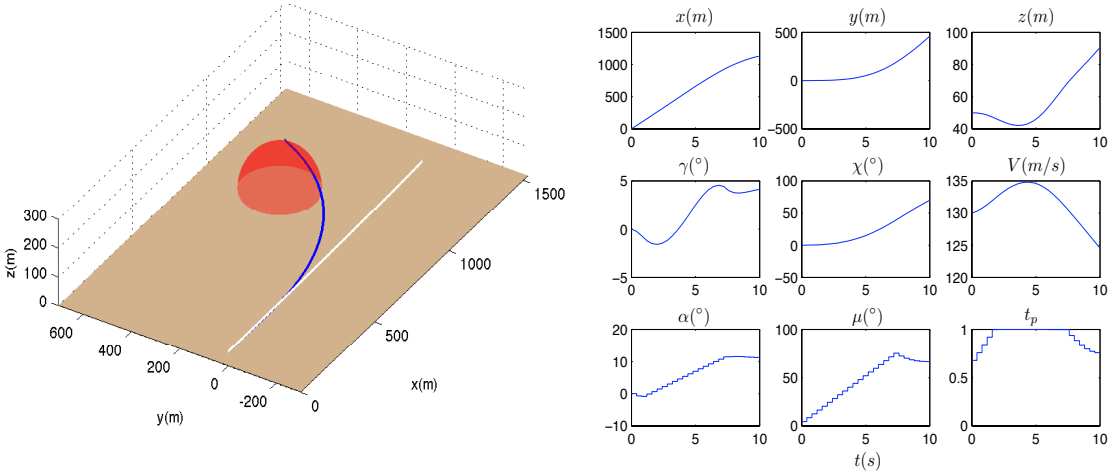


Figure 6.4: PV entry problem for a 10s time horizon. Solve time = 0.32s.

states in the first two rows and the generated inputs in the final row. In the PV Avoidance System this computation is performed in parallel to the original avoidance problem. When the trigger computation converges to a solution (as shown in Figure 6.4) the trigger is activated. Note that the bound constraint $|\mu| = 20^\circ$ is no longer applied as X and U are expanded to allow for the more aggressive manoeuvres that might be performed by a hostile pilot.

In contrast to the previous trigger mechanisms, this method decouples the trigger from the trajectory generation algorithm. In theory the trigger could activate despite a solution from the parallel calculation of the avoidance trajectory not being found. This issue is discussed in the next section.

6.6 Choice of Trigger Mechanism

The exact penalty function method offers computational advantages over the two solution method, but both suffer from the problem of allowing the pilot to retain control when flying adjacent to a PV. The PV entry method mitigates this problem by analysing whether it is possible to enter the PV from a particular state. The disadvantage however is that pilots have a reduced operating airspace, and for example would be unable to fly between two PVs placed closely together. The PV entry method also has an increased computational cost, particularly as the number of PVs present increases.

Recall that the aim of the trigger mechanism is to activate the override before the aircraft enters the infeasible space (x, u) where

$$\mathcal{X}_N(x, u) = \emptyset. \tag{6.28}$$

It is *not* necessarily the case that if $\mathcal{X}_{N'}^{II} = \emptyset$, then $\mathcal{X}_N \neq \emptyset$ holds, i.e. if a solution is not found for the PV entry method, there will be an avoidance trajectory available. In other words, even if a PV is not reachable over a shorter time horizon, it is not necessarily always avoidable over a longer horizon. Therefore a composite trigger using multiple methods is the safest option. In this strategy both the PV entry method and exact penalty function are implemented in parallel. This does not increase the computational burden significantly since the PV entry method requires the trajectory generation algorithm of OCP2 to be run in parallel. The trajectory generation algorithm is simply replaced by the exact penalty function method (which has trajectory generation inbuilt) with input matching at the start of the trajectory as a soft constraint. The control system is then implemented as shown in Figure 6.5.

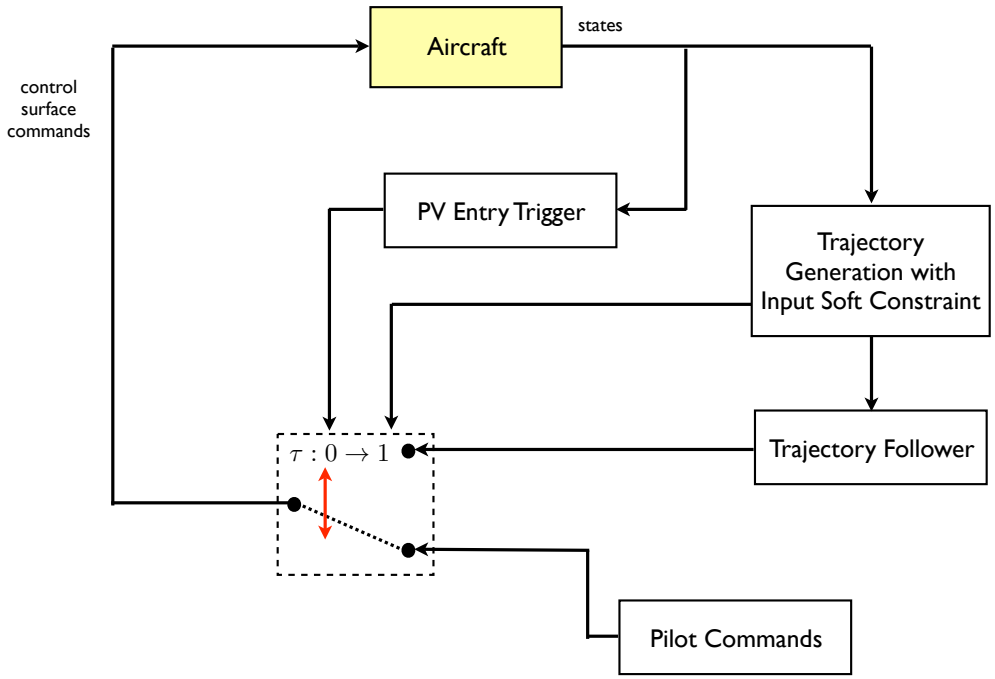


Figure 6.5: Schematic of control system design with composite trigger. The override is triggered if *either* the PV Entry computation converges to a solution *or* the input soft constraint is violated in the trajectory generation computation.

Chapter 7

Trajectory Following Inner Loop Controllers and Testing

In this chapter the trajectory following inner loop control strategy (see Figure 1.2) is outlined. The primary aim of implementing the trajectory following controllers is to validate the trajectory generation and trigger mechanism work of Chapters 4 and 6. Therefore the nonlinear guidance law controller to be described in Section 7.4, is considered adequate for these purposes. Other control design techniques where outer guidance and inner control loops are combined have been investigated for trajectory following such as MPC [51], differential flatness [73] and neural network based adaptive control [48]. The separation of guidance and control is the method commonly used in flight control applications as it has simpler, well established design methods.

The chapter starts with a description of the simulation software used for validation and the chosen aircraft model. It then outlines the trajectory following strategy with some heuristics to deal with control from extreme attitudes. Finally, results from human testing of the PV Avoidance System in the Imperial College London full motion simulator are given.

7.1 Simulation Software

The PV Avoidance System is validated using an aircraft simulation program. X-Plane [3] is the program chosen, due to its recognized realism, flexibility and its use on the full motion aircraft simulation hardware at Imperial College London.

X-Plane is a closed-source, commercially available package that allows interaction with external computing processes. It has a highly developed flight model that analyses the geometric shape of the aircraft model and calculates the forces along each element of the structure through a process know as 'blade element theory'. This is done 'in-the-loop', i.e. as the program is running, many times per second. The forces acting on the aircraft are converted

into accelerations which integrate to give velocities and positions. X-Plane contains two user interfaces for designing a vehicle to be flown in the simulation environment. The first is *Airfoil Maker* where characteristics for all the airfoils on the aircraft are defined. The lift, drag and moment curves for the wings, fin and tailplane are specified here. X-Plane uses this data to determine the forces on each element during the simulation. *Plane Maker* is used to build the aircraft geometry (including fuselage, wings, engines etc.). Engine performance, weight, control surfaces, cockpit layout and a host of other parameters are all defined in this program. Both programs produce files that are accessed by the simulator to allow the simulator's built in physics to interact properly with the model. However, the equations for the simulation physics are not available to the user.

For this research a Boeing 767 model was constructed in Plane Maker. This was done by taking a freely available 767 model and modifying it to correctly resemble the Boeing 767 geometry. A screenshot of the model in flight can be seen in Appendix B (Figure B.1). The geometric data comes from the manufacturer [1] and Jane's All the World's Aircraft [47]. Information on airfoil types and engine performance are not in the public domain, therefore estimates are used. Details of the development of the aircraft flight model, including lift, drag and thrust data, can be found in Appendix C. The same aerodynamic force data is used in both the X-Plane model and the 3DOF model (3.1) used for trajectory generation. In doing this it must be noted that the behaviour of the real 767 aircraft would be different. However any override system installed on the real aircraft would use a similar model derived from the actual performance data of the aircraft and engine obtained from flight testing.

7.2 Control System Architecture

The PV Avoidance System is implemented in Simulink. The trajectory generation code (Chapter 4) and parts of the trigger mechanism (Chapter 6) and trajectory following code (Chapter 7) are written in C++ and embedded in Simulink blocks via Matlab *mex* functions. These represent the most computationally intensive elements of the system and hence the extra effort in writing the software is required to run the controller at an acceptable rate. The trigger mechanism and trajectory generation code are run in parallel, using the parallel computing toolbox in Matlab, for the same reason. The testing described in this chapter was performed on a 2.4GHz Intel Core 2 Duo Mac Pro. Communication between the X-Plane simulation software and Simulink is described in Appendix B. Figure 7.1 shows a basic schematic of the control system design. The aircraft states are fed into both the trigger mechanism and trajectory generation code. This outer loop runs at a rate of 1Hz*. Once the trigger is activated, control is transferred from the pilot to the trajectory following element which tracks the last trajectory

*This rate is sufficient for the proof-of-concept system. However a future real implementation would ideally run at a higher rate with better processors.

produced by the trajectory generation code. Recalling that the trajectory generation code

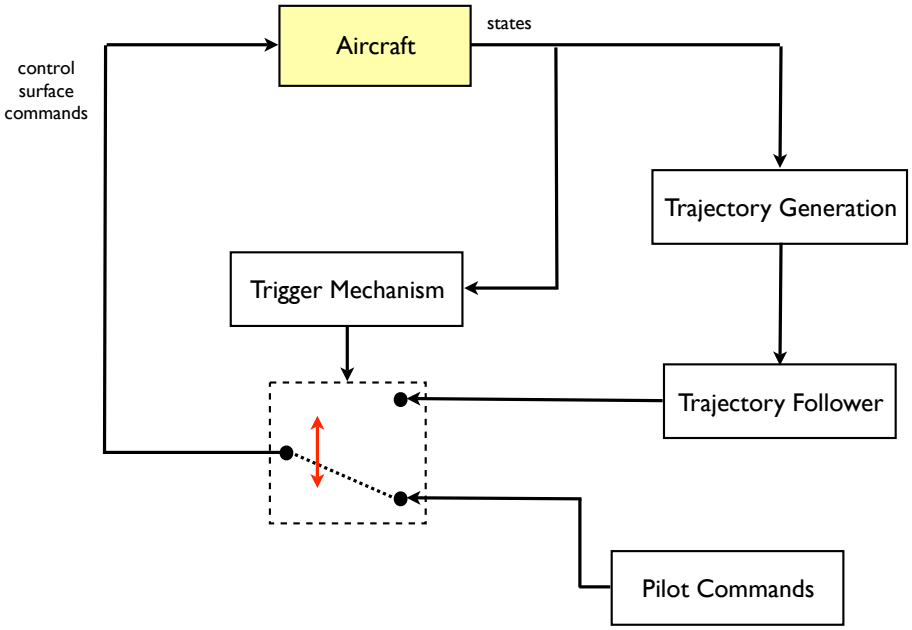


Figure 7.1: Schematic of control system design.

outputs a discrete state sequence $\mathbf{x}$ corresponding to a control sequence $\mathbf{u}$, linear interpolation is used to deduce the desired trajectory set point for the trajectory follower.

In the following sections two methods of trajectory following are described. The first takes the roll angle μ and throttle t_p control set points and flight path angle γ state set point from the trajectory generation scheme and uses proportional-integral-derivative (PID) control to track these variables. The second uses a nonlinear guidance law to track the Cartesian (x, y, z) states of the trajectory in order to overcome uncertainties associated with disturbances such as wind or modeling errors. In the simulation results shown within this chapter, the desired trajectory generated via the trajectory generation scheme is displayed in blue and the actual trajectory flown by the aircraft is displayed in red. The white line shows the original trajectory of the aircraft. In simulations where the full system is tested, the override trigger of choice is the PV entry method of Section 6.5.3 with a lookahead time of 6s. The lookahead time was chosen through experimentation as a balance between ensuring an avoidance trajectory was available from the trajectory generation module and not overly constraining the flyable airspace of the pilot. At a typical speed of 130m/s this would result in the override being activated at approximately 780m from the PV boundary if the aircraft trajectory was aiming directly at the PV.

7.3 Set Point Tracking

In the set point tracking scheme the flight path angle and roll set points (γ, μ) of the trajectory generation code are tracked via PID controllers. The throttle t_p is directly fed as a command to the engine. The yaw command is set to be zero over the course of the override horizon which is consistent with our trajectory generation assumption of zero side slip in the 3DOF model (with no wind) and corresponds to a coordinated turn in flight. Figure 7.2 shows a schematic of the trajectory following system. The lateral and longitudinal control is decoupled, with the roll controlling the lateral motion and the flight path angle the longitudinal motion. It should be noted that this decoupling is only valid for normal operating attitudes. If an extreme attitude is encountered this assumption is invalid (e.g. at a 90° roll attitude increasing the angle of attack will result in a greater turn rate rather than an altitude change). This problem is addressed in Section 7.5.

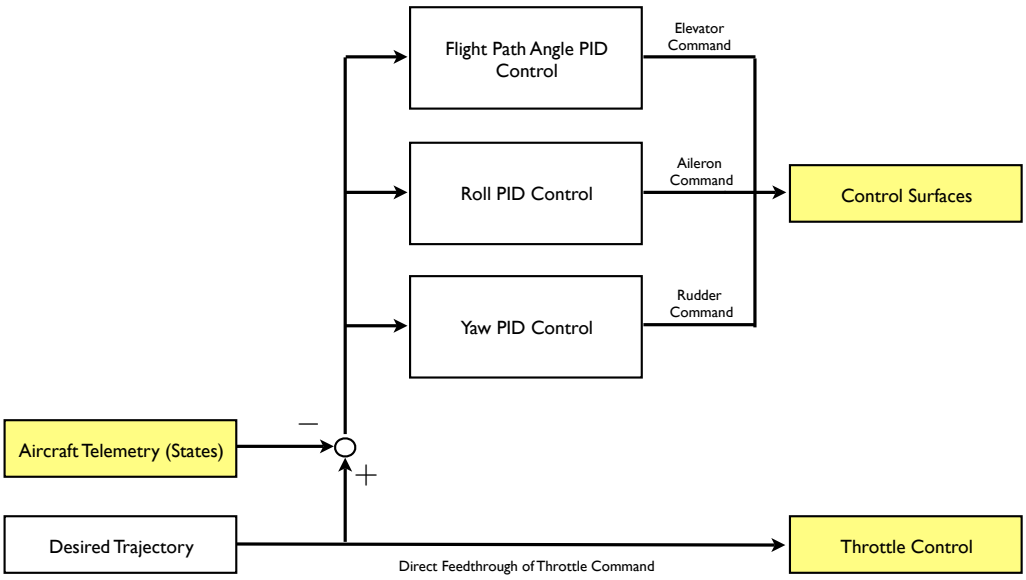


Figure 7.2: Set point tracking

The flight path angle γ is chosen as a set point for longitudinal control instead of the angle of attack α since the latter is highly sensitive to modeling discrepancies between the 3DOF model and the actual (X-Plane) aircraft. This is due to the large variation in lift force caused by a small discrepancy in lift coefficient C_L. If the aircraft velocity is accurate over the time horizon, the tracking of flight path angle will lead to an accurate tracking of the vertical velocity of the aircraft. Figure 7.3 shows an example of an avoidance trajectory flown in X-Plane whilst tracking the angle-of-attack. The x-z longitudinal track graph shows that the error in altitude

z grows over the time horizon and is greater than 60m at the end of the avoidance trajectory. In contrast flight path angle tracking produces an error closer to 15m over the course of the trajectory as shown in Figure 7.4. In both examples the tracking of roll angle μ leads to good accuracy in the desired heading angle, hence the x-y lateral track graph displays little error. There is some deviation in velocity due to model errors in both the drag forces between the 3DOF and X-Plane models, and thrust forces generated by the engine model.

Figure 7.5 shows a second example using PID tracking of flight path angle but now with a significant initial yaw β applied before the override is activated. The inclusion of this example is intended to demonstrate the validity of using a lower order 3DOF model which assumes zero yaw. The graph of yaw angle shows the PID controller driving the yaw to zero enabling a good overall tracking of the desired trajectory. The response is underdamped, however it is very difficult to achieve further damping of the yaw response due to the sensitivity of response to small rudder inputs at high yaw angles.

An immediate problem that arises with the set point tracking is a consequence of the indirect tracking of (x, y, z) through other states. Although the PV constraints exist in the Cartesian space, modeling errors and disturbances lead to imperfect tracking of the true trajectory. This is to be expected since tracking of accelerations in general is unsatisfactory, due to integrators down stream of the control which result in large position errors caused by small perturbations from the required acceleration. One example of this is shown in Figure 7.6 where a crosswind of 30 knots is applied. The lateral track shows the error increasing significantly over time due to the wind, resulting in penetration of the PV. The lateral track error at the end of the trajectory is close to 300m. To counter this problem a trajectory following method which explicitly follows the Cartesian trajectory was sought and is described in the following section.

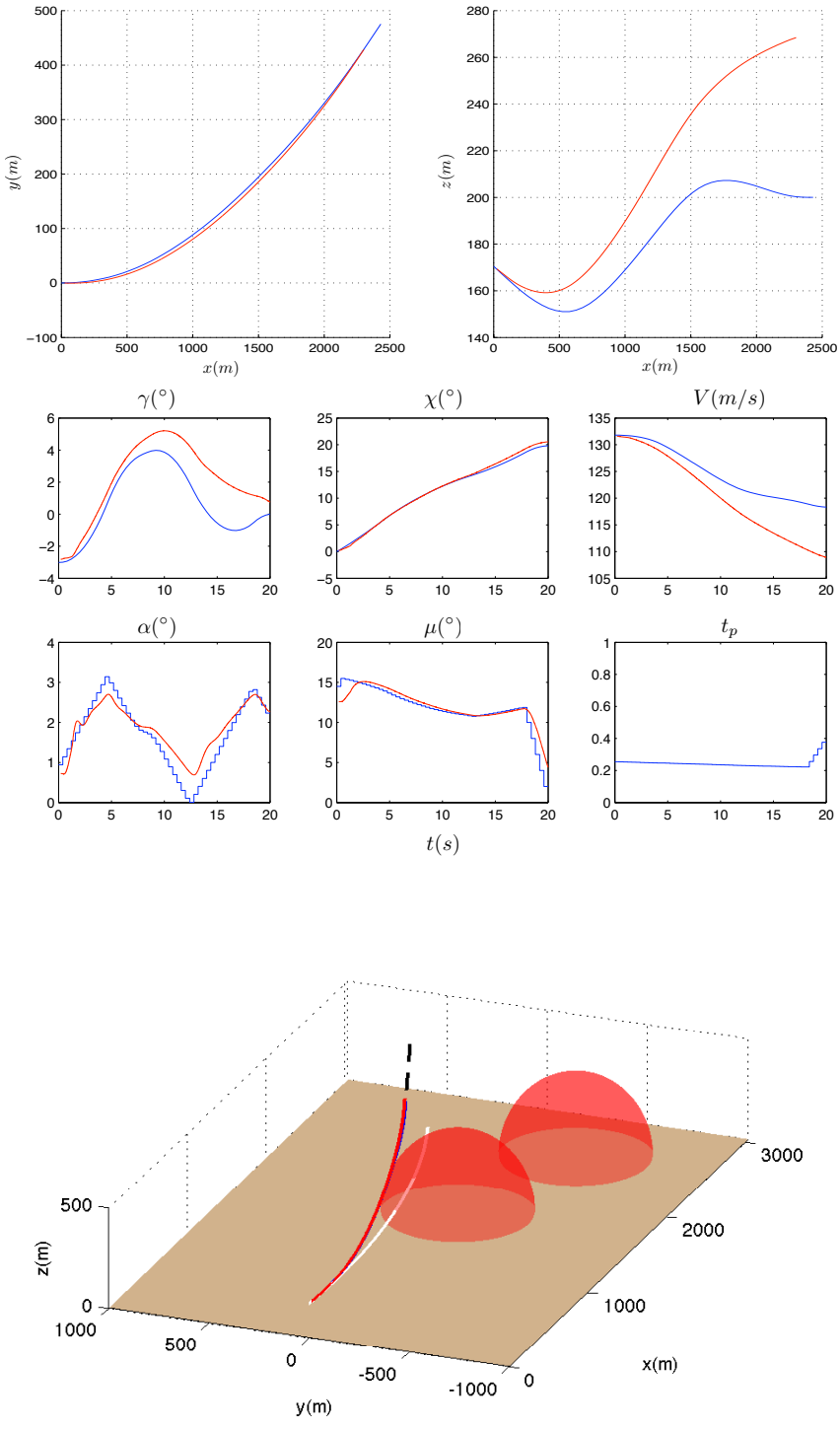


Figure 7.3: Simulation result using PID tracking of (α, μ).

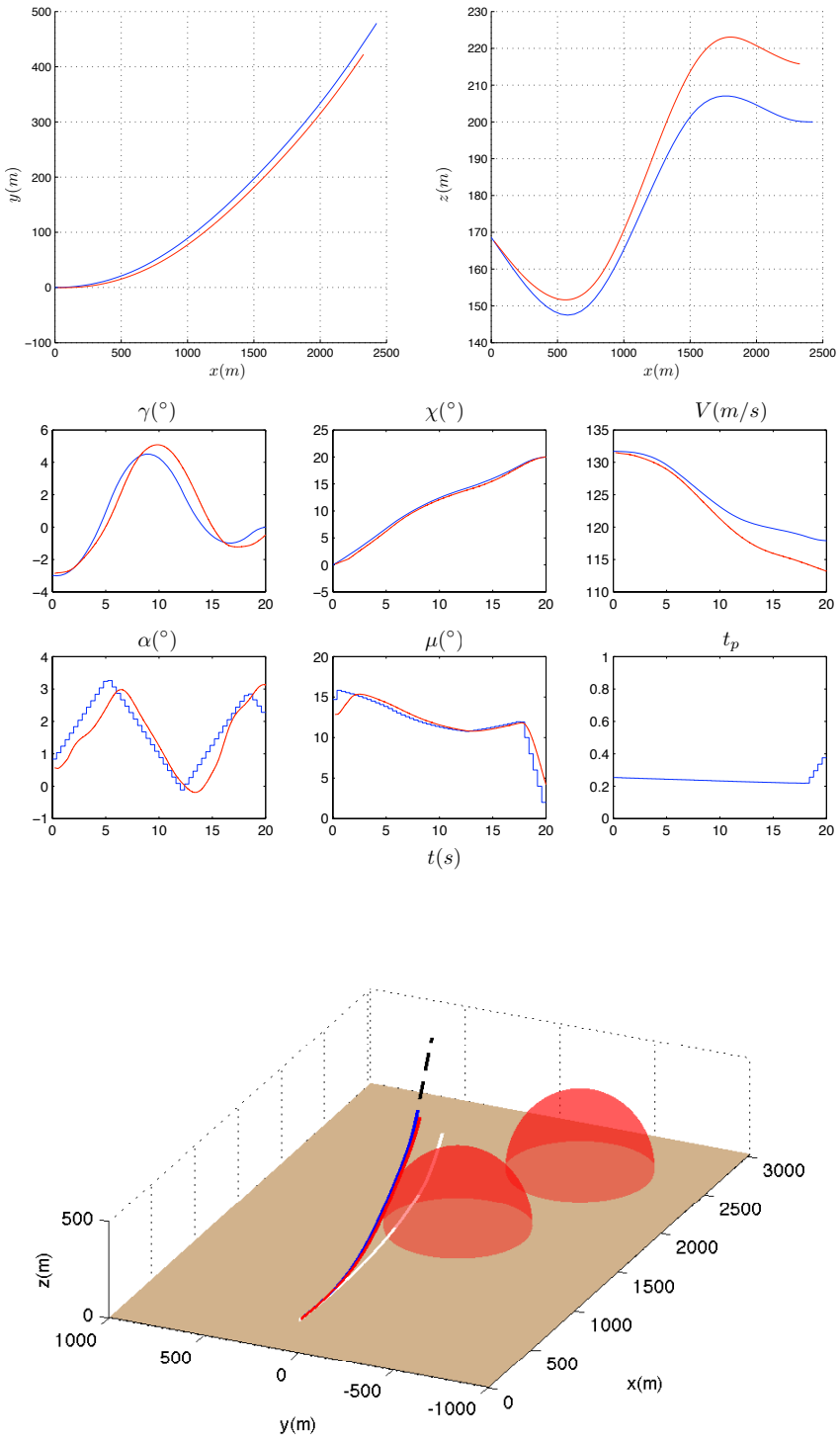


Figure 7.4: Simulation result using PID tracking of (γ, μ).

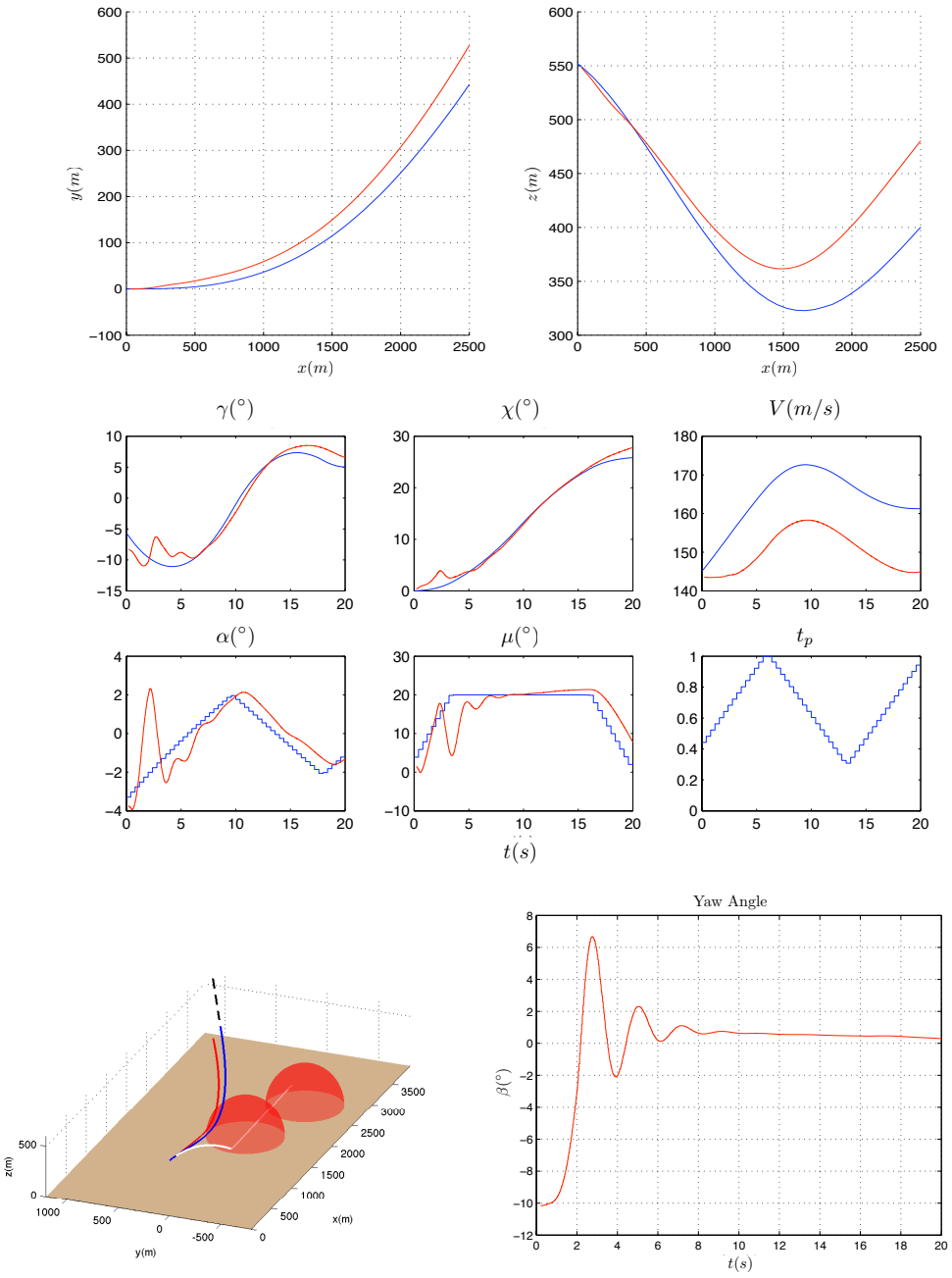


Figure 7.5: Simulation result using PID tracking of (γ, μ) with a large initial yaw input applied.

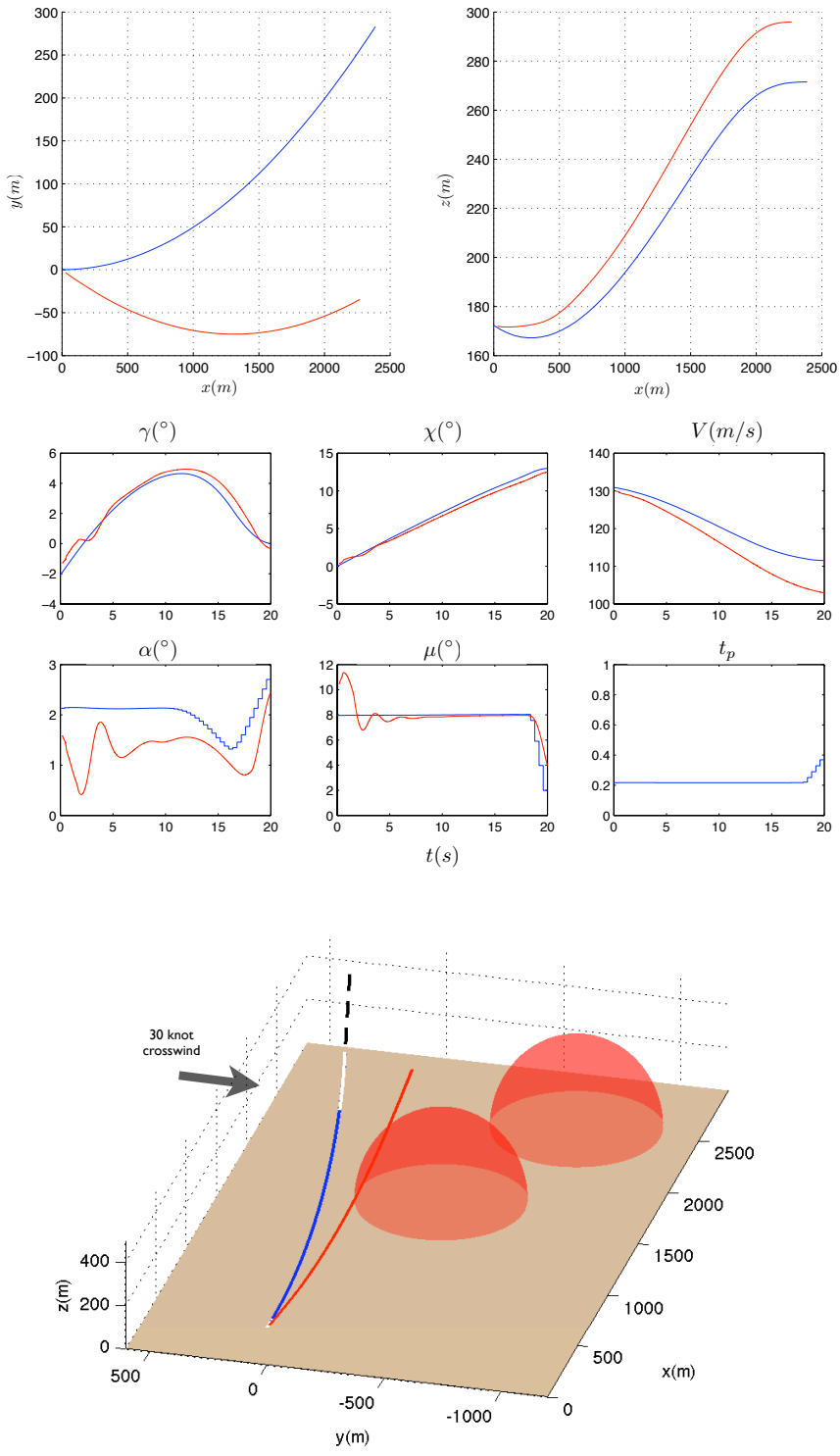


Figure 7.6: Simulation result using PID tracking of (γ, μ) with a 30 knot crosswind.

7.4 Cartesian Trajectory Tracking

The nonlinear Cartesian trajectory tracking described in this section is based on work in [68], extended to include an integral component and implemented in three dimensions. Considering initially the lateral guidance and assuming movement only in the horizontal plane, a reference point on the desired trajectory is designated and a lateral acceleration command determined by

$$a_{lat} = 2\frac{V^2}{L_{lat}}\sin\xi, \tag{7.1}$$

where a_{lat} is the lateral acceleration command, V is the aircraft ground velocity, L_{lat} is the lookahead distance and ξ is the closure angle. A number of properties of the guidance law can be inferred from Figure 7.7. At each point in time a circular path is defined as shown, passing

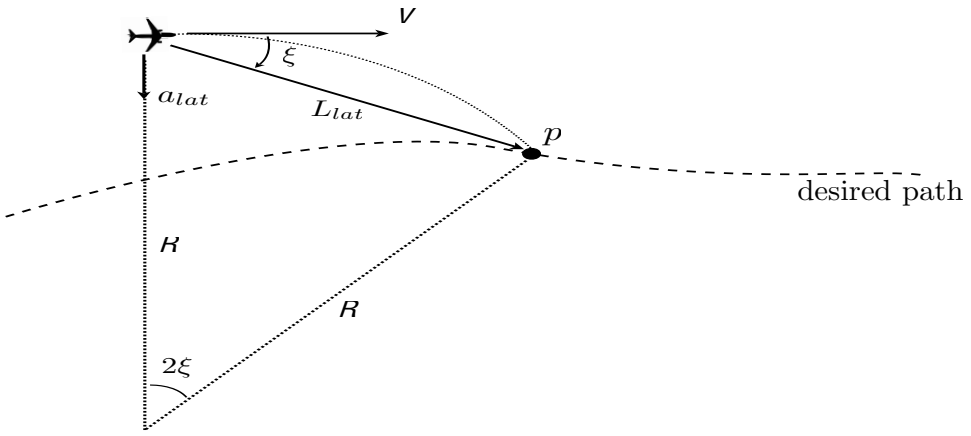


Figure 7.7: Nonlinear guidance law

through the aircraft position and the reference point p. The acceleration command generated by (7.1) is the centripetal acceleration required to follow the circular path. This is easily verified since $L_{lat} = 2R\sin\xi$ and the centripetal acceleration is $V^2/R = 2(V^2/L_{lat})\sin\xi = a_{lat}$. If the turning radius R is within the manoeuvring capabilities of the aircraft, the circular path will be flown. If the aircraft is a large distance away from the desired path, the closure angle ξ is large. The direction of acceleration is dependent on the sign of the closure angle.

A linear analysis of the guidance law shows that it approximates a proportional-derivative (PD) controller [68]. Figure 7.8 illustrates the notation used in the linearization. The desired path is represented as a small perturbation from a nominal straight line. Assuming the magnitude of the angles $\xi_1 \approx (d - d^*_{ref})/L_{lat}$ and $\xi_2 \approx \dot{d}/V$ are small, for a straight line desired trajectory ($d^*_{ref} = 0$) it can be shown that

$$a_{lat} = 2\frac{V^2}{L_{lat}}\sin\xi \approx 2\frac{V}{L_{lat}}\left(\dot{d} + \frac{V}{L_{lat}}d\right), \tag{7.2}$$

where d and d^*_{ref} are lateral perturbations of the aircraft and reference point from the nominal

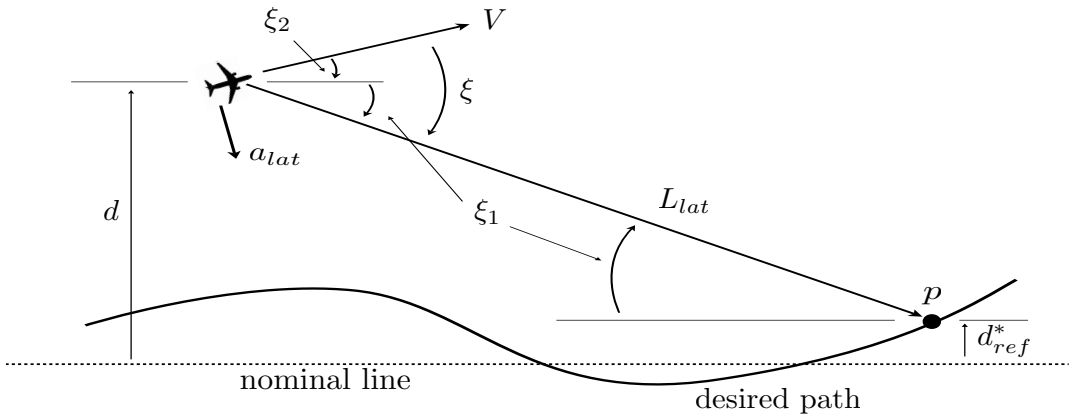


Figure 7.8: Linear analysis for following a perturbed line

straight line. The velocity V and lookahead distance L_{lat} determine the gains for the PD controller. In [68], further analysis showed that L_{lat} should be chosen based on the highest frequency content of the desired path. In the implementation for the PV Avoidance System, the assumption in [68] of a fixed V is dropped and the ratio V/L_{lat} is determined via trial and error methods in order to obtain good tracking over a range of initial conditions (and desired trajectories) for the aircraft. This effectively sets the proportional and derivate gains in (7.2) to be constant.

If the velocity V is taken as the aircraft ground velocity, thus including components of wind velocity, the wind disturbance is implicitly rejected by the guidance law. However it is worth noting that with this approach, an aircraft flying a straight line trajectory with a crosswind will tend to oscillate about the line due to the use of roll to control lateral movement (a piloted aircraft would conventionally yaw in this scenario). One of the drawbacks of the guidance law is the lack of an integral control element. Thus its performance is dependent on unbiased lateral acceleration and bank angle estimates [24]. To counter this, the lateral guidance law employed in this work incorporates an integral component such that

$$a_{lat} = 2\frac{V^2}{L_{lat}}\sin\xi + k_I \int L_{lat}\sin\xi \; dt, \tag{7.3}$$

where k_I is the integral gain. It is trivial to show that the roll command is then $\mu = a_{lat}/g$ which is tracked via PID control as before (Section 7.3).

The longitudinal guidance follows the same methodology. However in this case there is no trivial method for converting desired longitudinal accelerations a_{long} to flight path angle (or angle of attack). However, since $\gamma = 0$ at $a_{long} = 0$, and assuming a linear relationship between the flight path angle and longitudinal acceleration such that $\gamma = k_p a_{long}$, the linear gain k_p can be found by deducing a_{long} at another operating point. Note that this might not produce as good tracking as the lateral case since the relationship between flight path angle and angle of

attack will in fact not be linear, as gravity is now acting in the plane of motion.

Figures 7.9 and 7.10 display results of simulation tests with the nonlinear lateral and longitudinal guidance laws implemented in 30 knot cross wind conditions. Figure 7.9 shows that the nonlinear guidance law performs much better than the simple tracking of control inputs (cf. Figure 7.6) when a crosswind is present. However the lateral track graph shows a steady state error. In Figure 7.10 the integral gain of the lateral guidance law is increased in an attempt to eliminate this behaviour. This has the desired effect, however another complication arises. The high roll command required to track the lateral trajectory results in a worse performance for altitude control due to the coupling between lateral motion and angle of attack at higher roll angles as mentioned previously. A solution for this is to saturate the roll angle and flight path angle commands to prevent extreme attitudes being reached. This would of course lead to imperfect tracking in large crosswind conditions, but prevents the aircraft performing excessive and possibly dangerous manoeuvres. Since the prevailing wind conditions are known (by comparison of the ground and air velocities), they can be added to the equations of motion (see Remark 3) and accounted for in the trajectory generation part of the system. Given the constraints on states and inputs that are present in the optimization, the scheme would deliver a trajectory that would not require excessive attitudes even in the presence of wind.

A logical question to ask is what happens if the aircraft is already in an extreme attitude when the override is triggered. This is the subject of the next section.

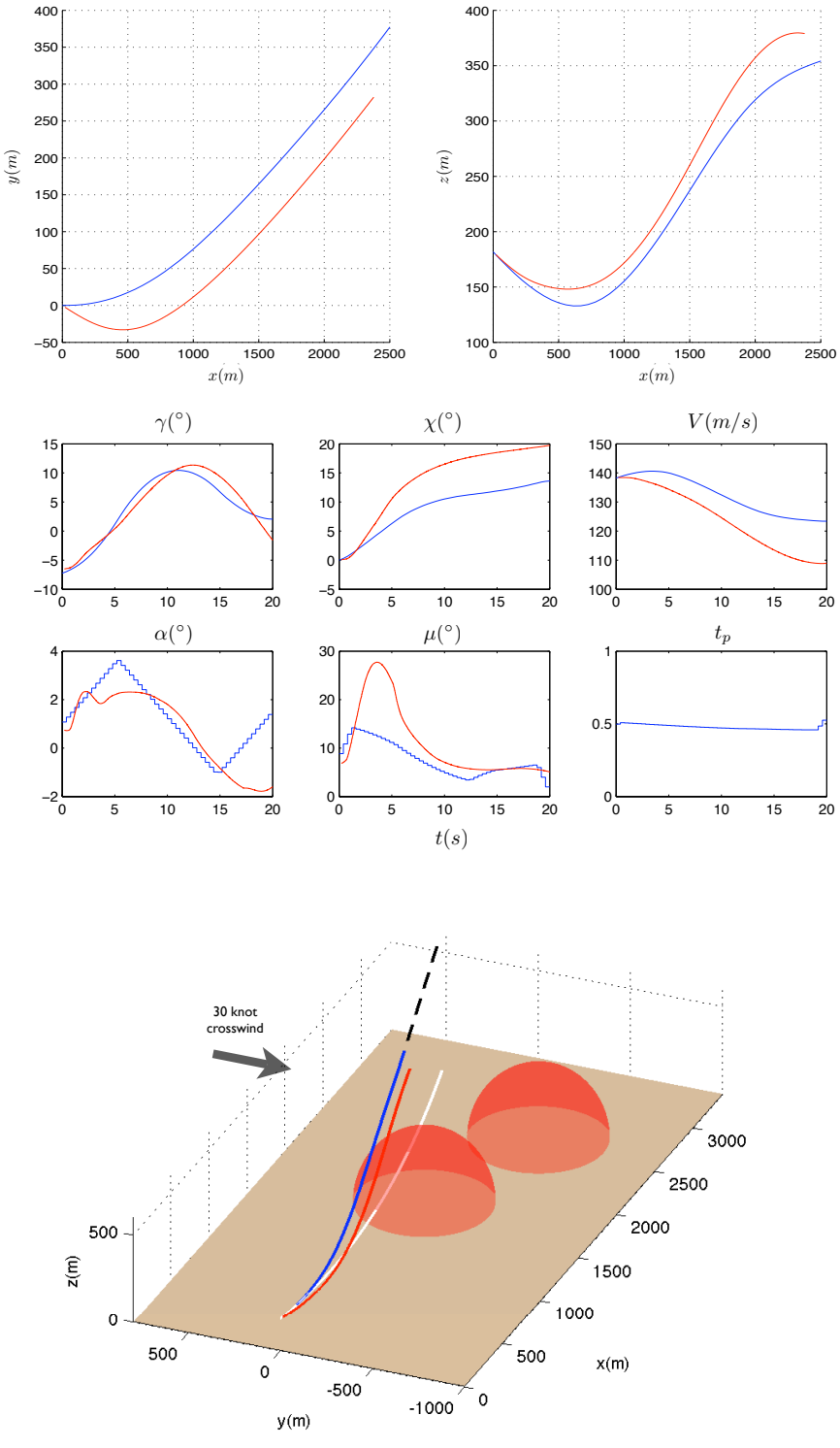


Figure 7.9: Simulation result using Cartesian trajectory tracking with a 30 knot crosswind.

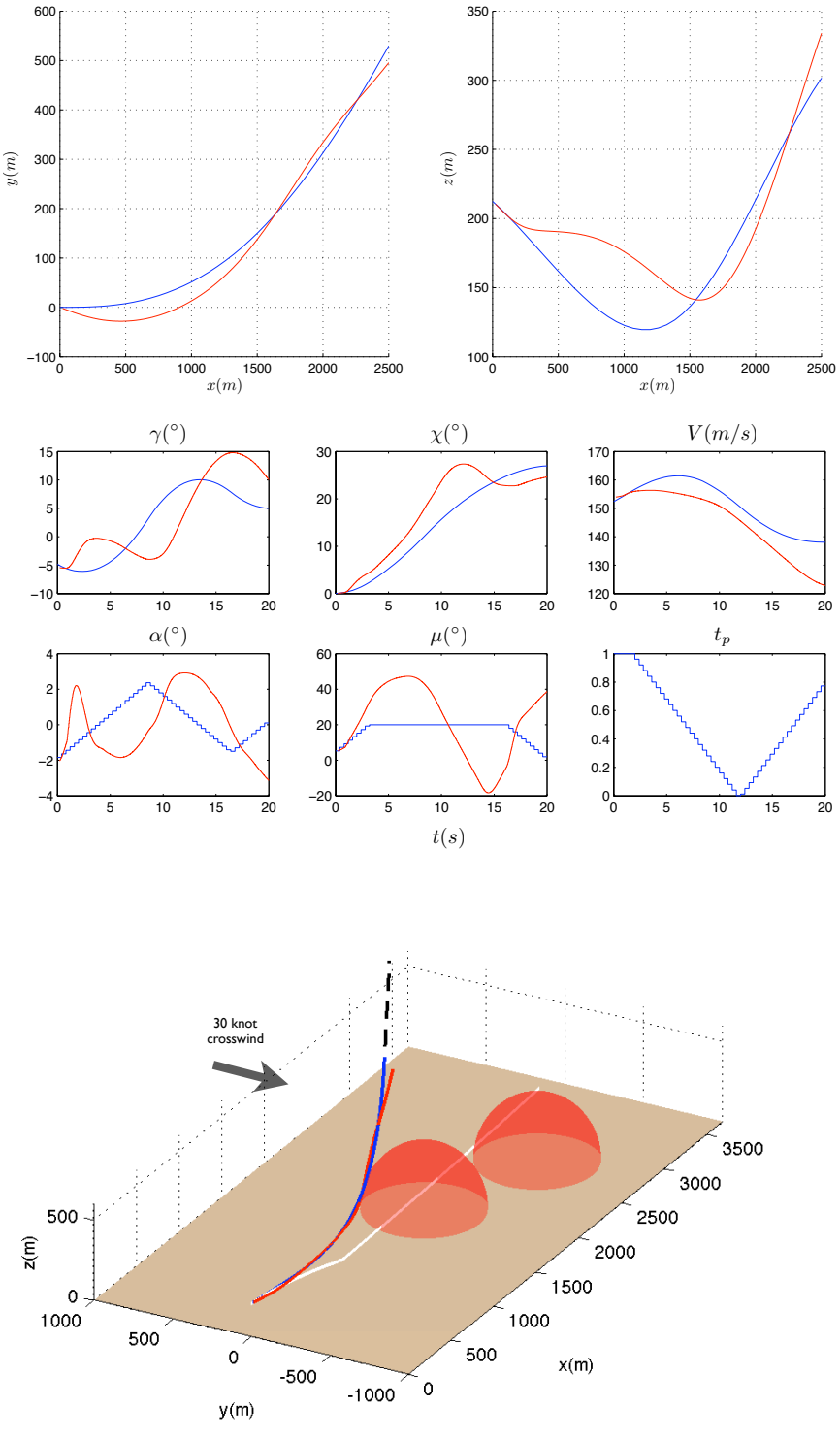


Figure 7.10: Simulation result using Cartesian trajectory tracking (increased integral gain for lateral guidance) with a 30 knot crosswind.

7.5 Control from Extreme Attitudes

Since an assumed hostile pilot is in control of the aircraft prior to the override being activated, if all the natural envelope protection has been disabled, the aircraft can be manoeuvred into extreme attitudes. These can be broadly put into two categories. Those which are *recoverable* by the control system, and those which are *unrecoverable*.

First looking at the recoverable situation, the initial input and state of the aircraft is outside the manoeuvre envelope constraints normally imposed for structural safety and passenger comfort (3.2, 3.3). In simulations the values used are shown in Table 7.1. The stall velocity

State or Input	Max. Value	Min. Value
x position	Inf	-Inf
y position	Inf	-Inf
z position	10 km	0 km
flight path angle (γ)	$30°$	$-30°$
heading angle (χ)	$180°$	$-180°$
velocity (V)	289 m/s	$V_{stall} + a$ m/s
AoA (α)	$20°$	$-5°$
roll angle (μ)	$20°$	$-20°$
throttle	1	0

Table 7.1: State and input bounds.

V_{stall} is calculated dynamically from the current aircraft state and a safety factor a is added. If the override is triggered when the aircraft state is outside these bounds (including being below the stall velocity) the envelope is extended to include the current attitude and subsequently tightened at an achievable rate over the multiple shooting steps back to the original intended operating envelope for the aircraft. For example if the initial roll $\mu_0 = 50°$ and the original roll bound is $\mu = 20°$, then the constraint is formed by

$$\mu_i^+ = 50 - ih\mu_{rate}, \qquad \forall i \in \{0, 1, \ldots, (50-20)/h\mu_{rate}\} \tag{7.4}$$

$$\mu_i^+ = 20, \qquad \forall i \in \{(50-20)/h\mu_{rate} + 1, \ldots, N-1\} \tag{7.5}$$

where μ_i^+ is the upper bound constraint at multiple shooting step i (note the result of the division $(50-20)/h\mu_{rate}$ is rounded down to the nearest integer), μ_{rate} is the achievable roll rate (deg/s) for the aircraft and h is the step size. This approach allows the problem of initial states lying outside the bound constraints to be dealt with, and also demonstrates the flexibility of a multiple shooting control strategy. An example is shown in Figure 7.11 where the initial roll is $-40°$. The trajectory generation scheme draws the roll back to within acceptable bounds. The overall lateral and longitudinal tracking is very good.

The unrecoverable case occurs when the aircraft attitude is substantially outside the normal operating envelope causing the decoupled control strategy for lateral and longitudinal movement

to break down. Figure 7.12 shows a simulation example where this occurs. The initial roll angle is beyond 90° and the flight path angle of −15° is causing a rapid descent into the PV. In this case the control system is not able to recover the aircraft to fly the generated avoidance trajectory. Although the PV was avoided, a loss of control has occurred.

It is questionable whether a single control system could recover an aircraft from a completely arbitrary initial state, although gain scheduling and other techniques could be used to increase the number of initial states that are recoverable. This problem is itself a substantial research topic in its own right [99, 65]. Therefore the approach adopted in this work is not to try and establish control from every initial state, but instead to prevent extreme attitudes from being reached near PVs. Therefore, at some distance from any PV a manoeuvring envelope protection is enforced so that input controls from the pilot are biased to prevent states outside the envelope being reached. This is already available on fly-by-wire aircraft, which in normal operating modes have envelope protection. For some aircraft such as the newer Airbus fleets (A320, A330, A340), these are implemented as 'hard' envelopes which the pilot cannot violate to prevent stall, over-banking, over-stressing and over-speeding [80, 46]. In the Boeing implementation the envelope 'soft' constraints can be overriden by the pilot. Ballistic entry via a stall is also prevented by the application of a hard manoeuvring envelope. For testing with the full motion flight simulator set-up, it is difficult to introduce modifications to the system between the control inputs in the cockpit and the computer running the simulation program. Therefore for demonstration purposes envelope protection is simulated by introducing an additional override trigger which removes control from the pilot once the boundaries of the envelope are violated. The generated trajectory is then flown as before. For testing purposes the manoeuvre envelope near PVs is restricted to the bounds given in Table 7.2.

State or Input	Max. Value	Min. Value
flight path angle (γ)	30°	−15°
velocity (V)	-	$V_{stall} + a$
AoA (α)	15°	−5°
roll angle (μ)	60°	−60°

Table 7.2: Envelope trigger bounds.

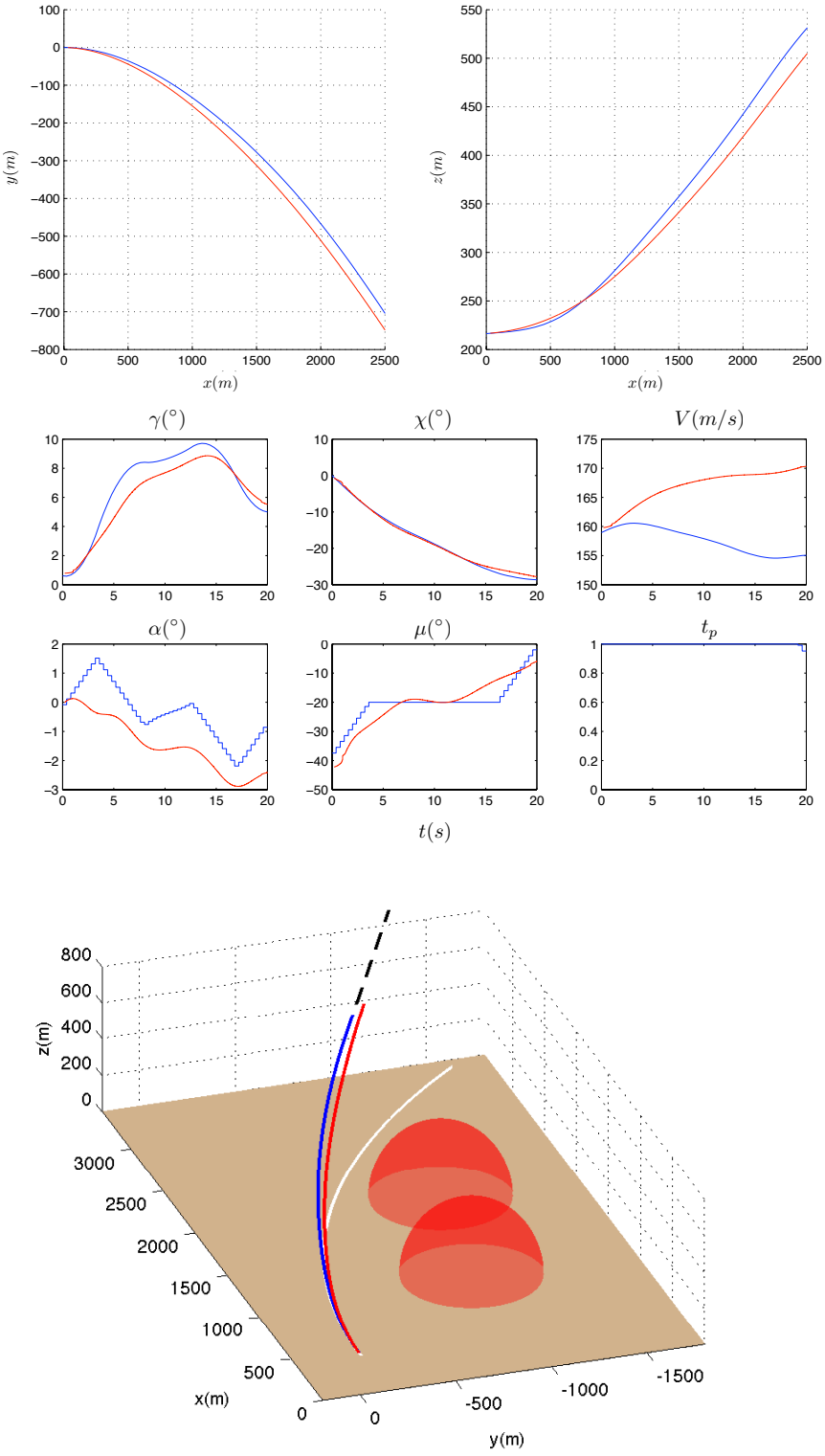


Figure 7.11: Simulation result using Cartesian trajectory tracking. The initial aircraft state is outside the conventional operating bounds of the control system.

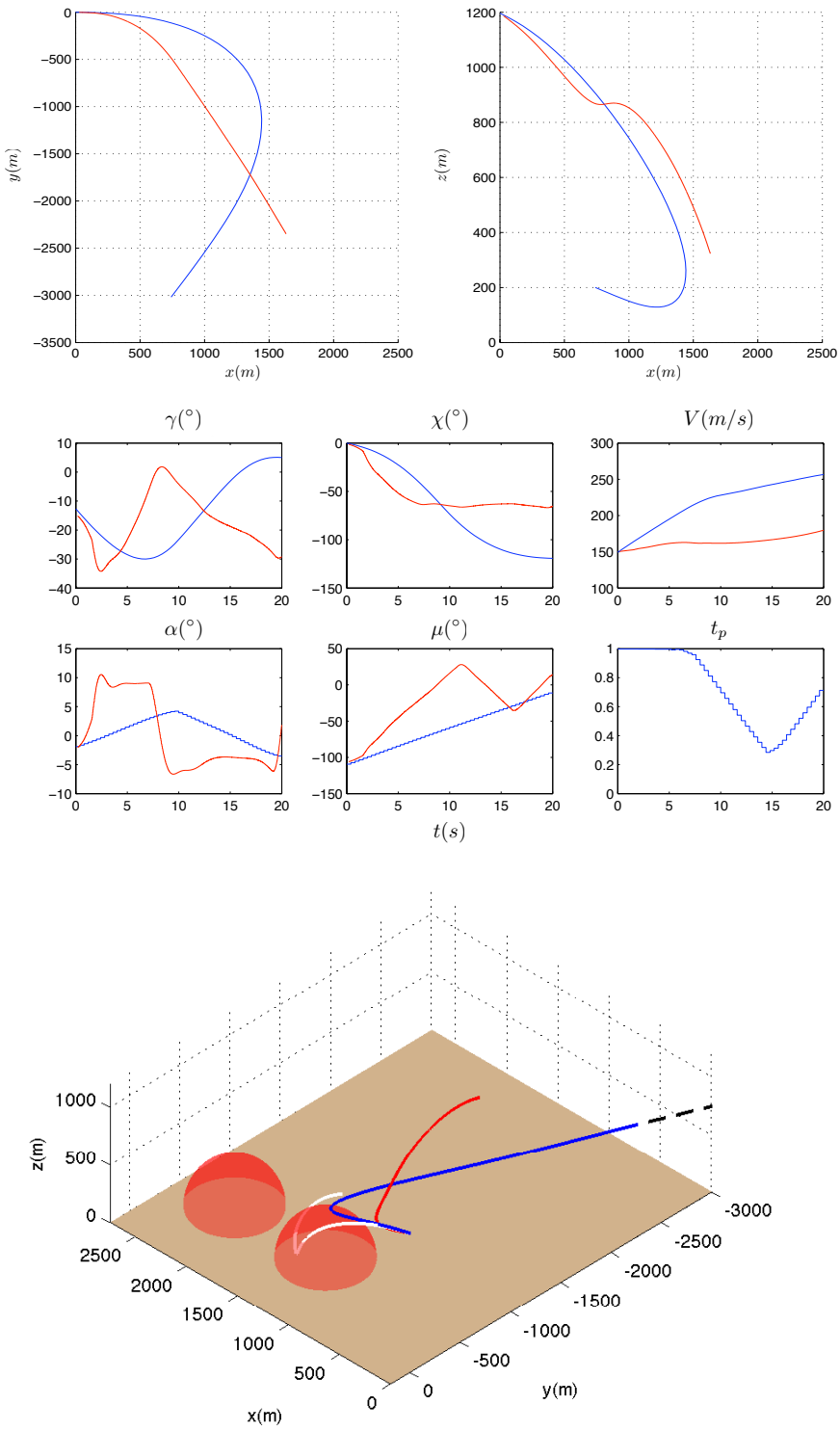


Figure 7.12: Simulation result using Cartesian trajectory tracking. The aircraft is in an extreme initial state and the trajectory following method is unable to fly the generated trajectory resulting in loss of control.

7.6 Testing

In order to test the PV Avoidance System, including the trajectory generation, trigger mechanism and trajectory following modules, the control system was set up in the Imperial College London full motion simulator. The MOTUS simulator is built by Fidelity Flight Simulation [2] and is used for flight training in many parts of the world (see Figure 7.13). The simulator runs the X-Plane simulation program and given the interfacing capabilities of X-Plane for reading data in and out, is ideally suited to control system testing. The simulator operates through six PCs, each running a copy of the X-Plane program. One machine acts as the central server, calculating the aircraft dynamics and relating the position updates to the other machines which include the peripheral visuals, instructor console and motion platform controller.



Figure 7.13: PV Avoidance System testing in the full motion simulator.

7.6.1 Procedures

Six students from Imperial College London (undergraduates and postgraduates) volunteered to test the system using the full motion simulator. Each volunteer had previous flying experience which was a prerequisite for selection since a basic knowledge of the flying controls (yoke, rudder and throttle) was required. None of the volunteers had previous flying experience on a twin engine aircraft. As with the previous simulations described in Sections 7.3 - 7.5, two hemispherical PVs were placed in a flat area and the PV boundaries were visible to the pilot (see Appendix B for screenshots of the simulation visuals). There was no wind applied during

these tests.

The following procedure was carried out with each student:

- The volunteer was briefed on the simulator and position of the relevant controls in the cockpit panel required for the experiment.
- The volunteer was then given ten minutes to become familiar with the handling of the aircraft and the controls.
- Five different starting scenarios were chosen with varying starting headings relative to the PVs, altitudes and aircraft speeds. Each volunteer had three attempts to enter the PV from each of the scenarios, i.e. fifteen attempts in total.
- The volunteers were encouraged to use any means they could think of to enter the PV (whilst maintaing the aircraft in the clean configuration). For the first set of five attempts they were not told about the mechanisms that would trigger an override. Before the subsequent attempts they were informed of the way in which the override system operates.
- Data from the PV Avoidance System and the aircraft telemetry were logged. Post-flight feedback by the volunteers was also noted.

7.6.2 Flight Simulation Results

Volunteer No.	No. of avoidance override trigger activations	No. of envelope trigger activations	No. of unrecoverable attempts	No. of PV penetrations
1	7	8	1	0
2	15	0	0	0
3	6	9	3	0
4	7	8	0	0
5	7	8	0	0
6	10	5	1	0

Table 7.3: Flight simulation results.

The results from the testing are shown in Table 7.3. The 'avoidance override trigger' of the second column refers to the PV entry method with a lookahead time of 6s. The 'envelope trigger' of the third column is activated via the manoeuvring envelope bounds of Table 7.2 applied when the aircraft is within a cylinder centered on the PV with a radius of 2km and a height of 2.5km. These might appear quite conservative and restrictive for pilots. However these bounds exist as hard constraints in some aircraft in current operation, for example the Airbus A320/A330/A340 series [38]. The difference is that they can currently be disabled by shutting down the relevant flight control computer. The fourth column of Table 7.3 is the number of times the aircraft was unrecoverable (resulting in a crash into the ground) whilst the

override was activated. This would happen, for example, when the envelope trigger is activated due to an extreme manoeuvre performed by the pilot very close to ground level. Attempts were ignored if the pilot crashed into the ground before the override was triggered. In these situations the pilot was given another attempt.

Figure 7.14 shows the angle-of-attack, roll and throttle (α, μ, t_p) time series for a typical PV penetration attempt. In the figure the input data is not collected for the first three seconds, hence each input is logged as zero over this initial time period. As the pilot approaches the PV over the first 40s of flight an erratic angle-of-attack α profile is observed as the pilot is pitching unsteadily. Maximum throttle t_p is engaged by the pilot around 24s. The roll profile shows the pilot applying significant bank inputs first one way and then the other, in an attempt to penetrate the PV. At 40s the trigger is activated, and the smoother flight profile corresponds to the avoidance manouevre performed under automatic control by the PV Avoidance System. The final part of the time series shows the pilot behaviour after control authority is returned at 60s.

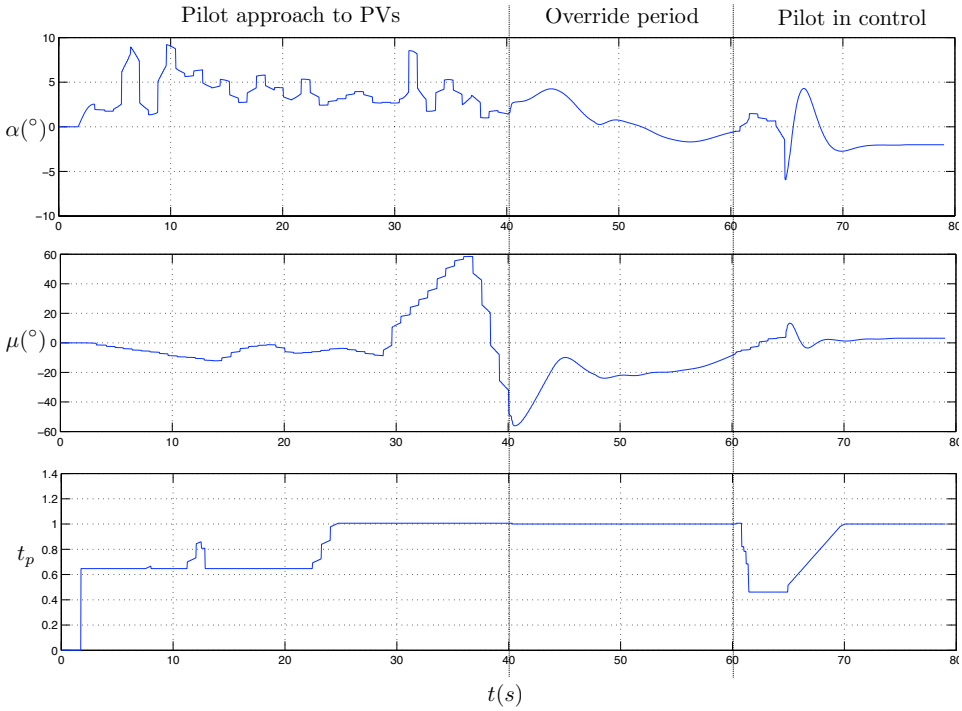


Figure 7.14: Input time series for a typical PV penetration attempt.

A number of observations were made during testing:

- There were no attempts during the testing that resulted in PV penetration.
- In general the volunteers approached the experiment with an attitude of trying to 'fool the system' into inducing a PV penetration. Interestingly this manifested as increasing frustration as more attempts were made (as would be expected from a real hostile pilot).

The qualitative feedback on the effectiveness of the PV Avoidance Sysem from comments made by the volunteers was encouraging. Most described the system as 'difficult to beat'. Some volunteers commented that the aircraft appeared to fly very close to the PVs (which are visualized in the simulation scenery). This is expected since the algorithm is attempting to remain as close as possible to the original aircraft trajectory without violating the PV constraints.

- The volunteers had different flying styles which can be seen in the results. Volunteer 2, for example, did not use aggressive manoeuvring, therefore in all of the attempts the avoidance trigger was activated as opposed to the envelope trigger.

- Use of an envelope trigger produced some attempts that could have entered the PV since the aircraft was not flying the generated trajectory to a high degree of accuracy. If an extreme manoeuvre is performed whilst still outside the envelope trigger cylinder (e.g. inverted flight) the override trigger will activate when the aircraft enters the cylinder. However the inner loop controller will no longer have the capacity to control the aircraft as desired. This leads to poor accuracy in trajectory following as shown in the example of Figure 7.15 and could potentially lead to PV penetration. In a real implementation of the system the envelope bounds would not be enforced as an override, but instead as a bias on the inputs (leaving the pilot in control of the aircraft). Therefore a solution could be to have a graded introduction of the bias, for example at 5km away from the PV no roll angle greater than 90° would be allowed and subsequently the constraint is drawn down to 60° by the 2km cylinder. Better tuned inner-loop controllers would help in these circumstances also.

- A number of attempts were made to stall the aircraft close to the PV. Whilst a full stall is prevented by the envelope bound, when recovering from close to stall the PV Avoidance System might return control to the pilot whilst the aircraft was still close to a stall situation despite ending with full throttle. Although the PV is still avoided, this is potentially dangerous if the pilot does not take the appropriate corrective action when control is returned. One option to prevent this is to increase the conservatism of the stall bound, and remove control authority (or have a stick pusher for the real system) at a higher speed. Another is to have a separate mode in the algorithm that recognises when the aircraft is close to stall speeds and therefore does not try to avoid the PV by increasing the aircraft height, but instead only by manoeuvring left or right. A similar flag is required for the engine-out case where the engines are not operational when the override is triggered. A third possibility is to maintain the aircraft under automatic MPC style control until a safe configuration is achieved (see Future Work section of Chapter 8).

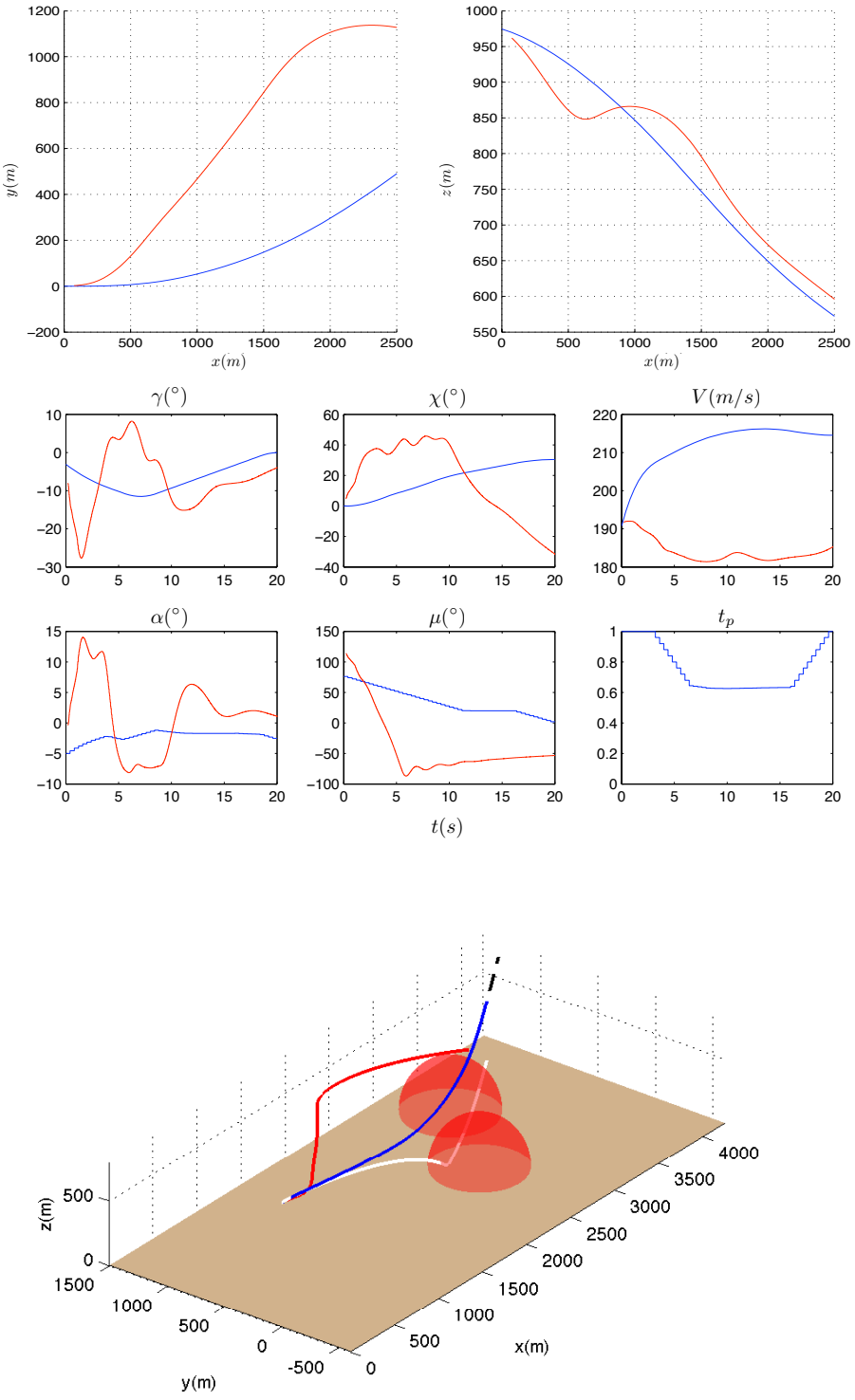


Figure 7.15: Flight simulation results for an extreme manoeuvre performed on entry into the envelope trigger cylinder. Note the poor tracking of the avoidance trajectory due to the high initial roll angle.

Chapter 8

Conclusions for the PV Avoidance System

8.1 Major Contributions of the Thesis

Online Trajectory Generation

The thesis presents a direct multiple shooting scheme for producing avoidance trajectories for aircraft with online computation times sufficiently fast for the PV Avoidance System (Chapters 4 and 5). Obstacles defined by non-differentiable (e.g. polytopic) constraints are dealt with via a dualization method that preserves continuity and allows the use of a gradient-based optimization method. The thesis also describes how similar dual methods can be used to ensure positive invariance, i.e. to ensure a safe straight line trajectory with a constant control input is available at the end of the avoidance trajectory. The combination of the positively invariant terminal condition and warm starting via shift initialization ensures that the non-convex optimization is initialized close to a feasible solution, and is used to demonstrate how to prevent an aircraft entering a trap situation where a solution is no longer available when generating avoidance manoeuvres. The direct multiple shooting scheme provides advantages over past work, by allowing a realistic vehicle model, flexible object constraints which can be time varying (i.e. moving) and dynamic model constraints based on the current vehicle state.

Trigger Mechanisms

The thesis presents three trigger mechanisms that could be used to activate a pilot override for preventing aircraft entering protected areas (Chapter 6). The two-solution method simply computes two avoidance trajectories, one from the current aircraft state and another from a future aircraft state. If the forward problem fails to converge the trigger is activated. The exact penalty function method uses soft constraints in the objective function to determine whether

an avoidance trajectory can be found if the current pilot input is held fixed at the start of the trajectory. It offers computational advantages over the two-solution method, but both suffer from the problem of allowing the pilot to retain control when flying adjacent to a PV. The PV entry method mitigates this problem by determining whether it is possible to enter the PV from a particular state. The disadvantage, however, is that pilots have a reduced operating airspace, and for example would be unable to fly between two PVs placed closely together. The PV entry method also has an increased computational cost, particularly as the number of PVs present increases, however its use is still feasible since the computations can be preformed in parallel. The safest strategy incorporates a composite trigger consisting of both the PV entry and exact penalty function methods. The latter of the two also acts as the avoidance trajectory generator.

Human Testing

Testing shows that the direct multiple shooting scheme, when coupled with suitable inner loop controllers, is capable of preventing an aircraft from entering a PV. This is dependent on the implementation of a hard envelope constraint on the control inputs that prevents pilots from performing extreme manoeuvres close to a PV. The aircraft is shown to be recoverable by the inner-loop controllers when the initial state is inside the envelope. The testing is only preliminary and is *not* sufficient to claim that the system is safe. A good deal of further testing under stringent conditions (with control and test groups) is required. This should be performed on a suitable test rig, with qualified pilots where hard envelope constraints are correctly enforced (as a bias on the controls rather than an override).

8.2 Directions for Future Work

There are a number of tasks required to move the project forward towards a full aircraft implementation of the PV Avoidance System:

- The current control system assumes the ground is flat. Integrating the system with a worldwide terrain database is required. Terrain is an obvious hazard when performing automatic manoeuvres close to the ground and therefore needs to be included as a constraint in the trajectory generation algorithm. For reasonably flat terrain, the highest point in the local area can be used as a fixed constraint. In areas of rapidly changing terrain, for example close to mountains, the land can be modeled as state exclusion regions in much the same way as PVs.

- Aircraft near cities often operate in crowded airspace. Therefore integration with air-to-air collision avoidance systems such as TCAS would be required. Assuming in such a scenario that only one aircraft is under the automatic override, this can be conveyed to

other aircraft (and air traffic control) and the TCAS advisories given with knowledge of the trajectory of the overridden aircraft.

- Human testing has been conducted with a reasonably simple scenario containing two PVs (more complex scenarios have been tested for the trajectory generation algorithm alone). Testing with more realistic scenarios where many PVs are present is required. Since aircraft will be flying at low level close to airports, on approach to land, and for take-off, testing in this environment is essential. A potential issue arises if PVs are present close to airports. There is flexibility available in the algorithm design to independently characterize the trigger lookahead time for each PV. Therefore PVs close to airports potentially have shorter lookahead times to allow aircraft to approach closer to them.

- Some consideration was made of external disturbances such as wind. However a thorough analysis of the maximum allowable wind conditions is advisable. An upper bound on the error in trajectory tracking would allow a suitable safety factor to be incorporated when the PVs are defined.

- The choice of a 20s override time is somewhat arbitrary. In the proof-of-concept system the avoidance trajectory is not re-generated during the override period (i.e. the inner loop controllers follow the single trajectory generated before the override was triggered). In a full implementation of the system it would be advisable to re-solve the open loop optimization problem as often as computationally feasible i.e. implement conventional MPC. In addition the trigger algorithm (PV entry method) should also be re-solved as the (now) shifting horizon avoidance trajectory is being followed. Once the trigger calculation no longer produces a positive result the override can be disengaged anywhere along the trajectory since there is no longer any threat of PV penetration. This should be preempted by some warnings to the pilot to ensure that they are aware that control authority will soon be returned to them. Some work would be required to ensure that these methods provide good theoretical behaviour for the overall system.

- A detailed system integration study is required to establish how the PV Avoidance System will be implemented into an aircraft avionics structure. Chapter 3 describes a top-level placement of the system. Since the avionics structure is aircraft dependent, system integration will be different for each family of aircraft.

- The technology will not be commercially successful without the approval of regulatory bodies. The concept of a pilot override is radical, and could meet serious criticism. Therefore it is important to obtain input from regulators and pilots as early in the process as possible. Given that this thesis has proved the feasibility of the technology, if further work is intended, it should begin with input from regulators. They can define

acceptable standards for the system in terms of PV size, shape and placement, and the conservativeness of the trigger mechanism. A study of the human factors issues is also required since that aspect has not been addressed in this work.

Appendix A

Linearized Model Optimization

A.1 Optimization of the Linearized System

There are inherent advantages to formulating the nonlinear problem OCP as a linearized optimization problem, which is the motivation of this chapter. Robustness to model and environmental uncertainties can be dealt with more easily using standard linear control techniques, allowing the avoidance trajectory selection to be limited to proven safe sets.

The nonlinear model dynamics in OCP are given by

$$\dot{x} = f(x, u). \tag{A.1}$$

Given an initial state and input (x_0,u_0), a reference trajectory ($\{x_i^r\}, \{u_i^r\}$) can be found by holding the input constant over the time horizon. The continuous time linearized model dynamics are given by

$$\delta\dot{x} = A_c\delta x + B_c\delta u. \tag{A.2}$$

The matrices A_c and B_c can be constant and defined as $\frac{df}{dx}|_{x_0,u_0}$ and $\frac{df}{du}|_{x_0,u_0}$, respectively, if the system is linearized about the initial point. Alternatively they can be time varying if the system is linearized about each point along the reference trajectory.

In discrete time (A.2) becomes

$$\delta x_{i+1} = A_d\delta x_i + B_d\delta u_i, \tag{A.3}$$

which for small step size h is approximately equal to

$$\delta x_{i+1} = [I + A_c h]\delta x_i + [B_c h]\delta u_i, \tag{A.4}$$

where the discrete time state transition matrix A_d and control effect matrix B_d have been

rewritten as their continuous-time approximations [86].

The equivalent optimization problem of OCP is now formulated in terms of parameters $(\delta x, \delta u)$ and can be stated as:

$$\begin{aligned} \min_{\delta x, \delta u} \; & J(\delta x_i, \delta u_i) = \delta x_N P \delta x_N' + \sum_{i=0}^{N-1} (\delta x_i Q \delta x_i' + \delta u_i R \delta u_i') \\ \text{subject to:} \quad & g_{cont}(\mathbf{x}) = 0 \\ & h_{rate}(\mathbf{x}, \mathbf{u}) \leq 0 \\ & \mathbf{x} \in X, \quad \mathbf{u} \in U \\ & p(\mathbf{x}) \notin \bigcup_{j=1}^{M} (\operatorname{int} \mathcal{C}_j \oplus r_j), \qquad \forall j \in \{1, \ldots, M\}. \end{aligned} \tag{A.5}$$

with $\delta x_i = x_i - x_i^r$ and $\delta u_i = u_i - u_i^r$. Note that this remains a non-convex optimization problem due to the presence of convex state exclusion regions. For the case of polytopic state exclusion regions the problem could be solvable exactly using a mixed integer linear program (MILP). To test whether the linearization is accurate enough to be used in place of a nonlinear model the input trajectory calculated by the linear optimization can be applied to the nonlinear model to establish the agreement with the state trajectory given by the optimization.

A.2 Test Cases

The linearized optimization (with A_c and B_c constant and linearized about the initial point x_0, u_0) is run with a number of test cases and the resulting trajectories tested against the nonlinear model. Two of the cases are shown in Figures A.1 and A.3. In these cases the cost matrices $P = Q = I$ and $R = I$, i.e. deviations of all the states and inputs away from the reference values are penalized. Angular states in the model are measured in radians and position states in kilometers.

Figure A.1 shows the trajectory and state/input plots for a simulation with an infinite height cylinder test case. As with the nonlinear case a feasible trajectory is produced. The computation time per iteration has been reduced through the linearization as expected because the intermediate integral computations are no longer required. By applying the initial state and control input back through the nonlinear model, the error introduced by the linearization can be calculated and is shown graphically in Figure A.2. The error is small.

A second test case is shown in Figure A.3. In this case two hemispherical PVs are placed in the original trajectory of the aircraft. Again the linearization error is shown in Figure A.4. In both cases the errors are small, suggesting the linearized model might be a viable option. However, a thorough analysis of the maximum error in the proceeding section is required to validate this.

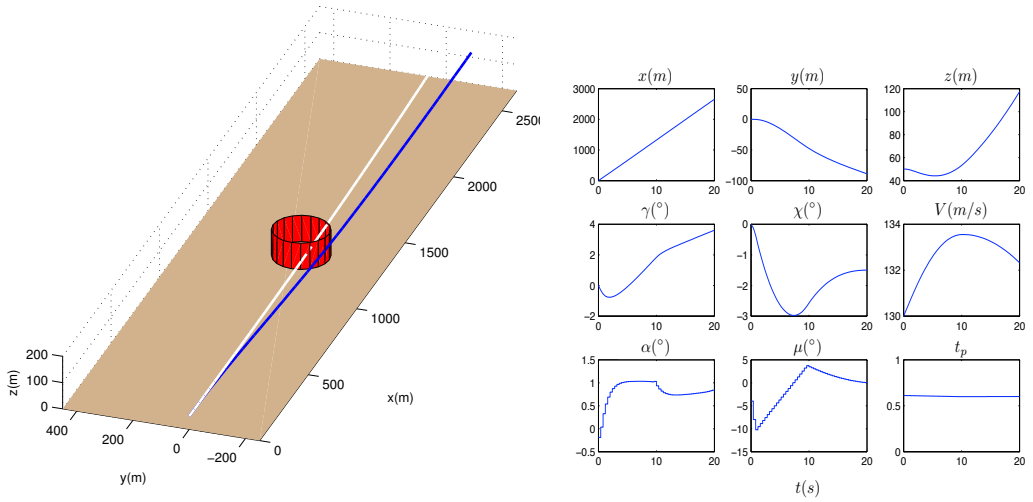


Figure A.1: Linearized solution: solve time = 0.24s, 16 IPOPT iterations.

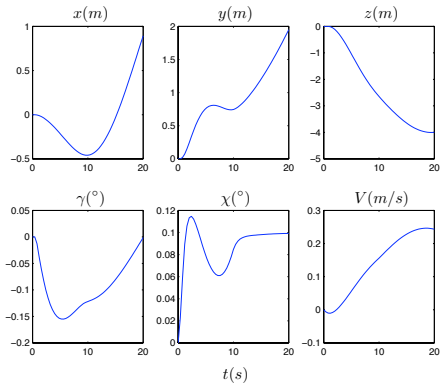


Figure A.2: Linearization error corresponding to solution of Figure A.1

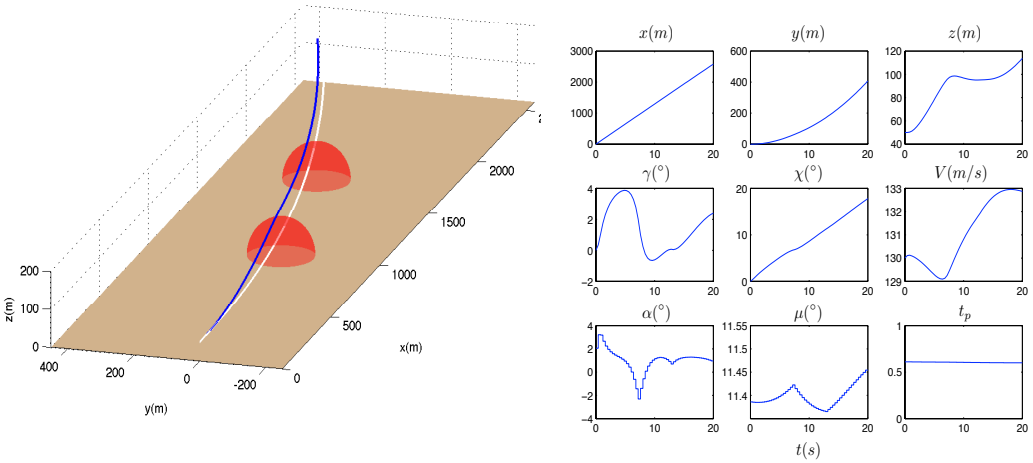


Figure A.3: Linearized solution: solve time = 0.38s, 23 IPOPT iterations.

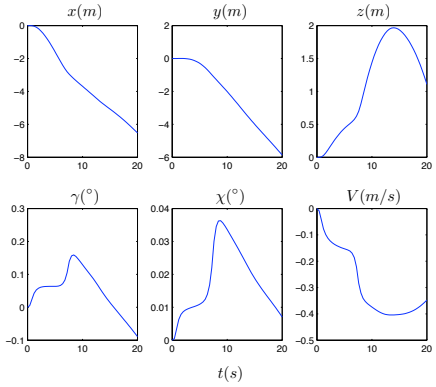


Figure A.4: Linearization error corresponding to solution of Figure A.3.

A.3 Quantifying the Maximum Trajectory Prediction Error due to Linearization

The motivation of this section is to determine the maximum error introduced by linearizing the system at the initial point of the trajectory. It is important to note that this would *only* be the error introduced by the linearization, and would not encompass other errors such as model uncertainty and disturbances like wind. Fixing $x_0 \in X$ and $u_0 \in U$ to correspond to straight and level flight at 130 m/s, the following problem can be posed for a 1s time step:

$$\begin{aligned} \max_{dx,du} \quad & \|Adx+Bdu - [f(x_0+dx, u_0+du) - f(x_0,u_0))]\|_2 \\ \text{subject to:} \quad & A = \frac{df}{dx}(x_0,u_0) \\ & B = \frac{df}{du}(x_0,u_0) \\ & x_0 + dx \in X \\ & u_0 + du \in U, \end{aligned} \tag{A.6}$$

where X and U are bounding sets determined by calculating values for the maximum change from initial state x_0 and initial input u_0 achievable over a trajectory of 20s. The solution to (A.6) therefore upper bounds the maximum error in the rate of change of states due to the linearization.

The maximum errors in flight path angle, heading and velocity rates ($\dot{\gamma}$, $\dot{\chi}$ and $\dot{V}$) are found to be 10.3 deg/s, 2.5 deg/s and 11 m/s^2 respectively. These errors are too large to be acceptable and arise due to inputs where the angle of attack α and roll angle μ are at their max/min values. In the equations of motion, $\dot{\gamma}$ and $\dot{\chi}$ contain terms of the form $\sin\alpha \cdot \cos\mu$ and $\sin\alpha \cdot \sin\mu$ leading to highly nonlinear behaviour and large errors as shown when α and μ are far from the linearization point.

This analysis demonstrates that optimization of the avoidance trajectories through linearization of the 3DOF model can be used to produce feasible trajectories and in the cases simulated the linearization error is shown to be minimal. However, analysis of the maximum error shows that it is prohibitively large for use in the actual system. This justifies use of the nonlinear model.

Appendix B

X-Plane Simulation Software

B.1 X-Plane Flight Model

X-Plane performs the following steps to propagate the flight model:

- *Element break down*

  Performed during initialization only, the airfoil elements of the aircraft (defined when an aircraft model in X-Plane is constructed) are broken down into a finite number of elements.

- *Velocity determination*

  Performed twice per cycle. The velocity vector of each airfoil blade element is calculated from the aircraft linear velocities and the longitudinal, lateral and vertical arms of the element. Downwash, propwash and induced angle of attack from lift-augmentation devices are all considered.

- *Coefficient Determination*

  X-Plane converts the 2-D wing data defined in the aircraft model into finite values given the current aircraft state. Compressible flow effects are considered using Prandtl-Glauert and transonic effects (if required) are simulated with a mach-divergent drag increase.

- *Force build-up*

  Using the coefficients determined in the previous step with element areas of the first step and dynamic pressures (determined separately for each element based on speed, altitude, temperature, propwash and wing sweep), the forces are found and summed for the entire aircraft. These are subsequently used with the current mass and moments of inertia of the aircraft to determine linear and angular accelerations. The complete cycle then begins again from the second step and is repeated at a frame rate of at least 15 Hz.

B.2 External Communication with X-Plane

Despite the source code of X-Plane being closed, a plug-in package is freely available that allows users to read and override the X-Plane data variables. Aircraft control inputs (normally the joystick) can be commanded via an external program such as Simulink. Aircraft telemetry can then be output to Matlab for analysis. Alternatively X-Plane's flight model can be overriden through direct input of position variables, enabling X-Plane to be used as a display.

The C code required to manipulate data hooks in X-Plane is implemented through S-functions in Simulink. This allows controllers to be developed in Simulink that can close the loop for automatic control. This is feasible as long as the cycle time for the controller in Simulink is less than that of X-Plane's. X-Plane and Simulink communicate over specified ports (one for input and one for output), however Simulink does not run in real time therefore code is required to stop the Simulink thread until data is received from X-Plane. Communication over a network is also possible, thus a controller can be implemented on one computer with the X-Plane simulator running on another allowing both operations to run smoothly. Controller design has been performed previously using X-Plane and [26] describes how this can be done without the use of the plug-in package.

B.3 Simulation Screenshots

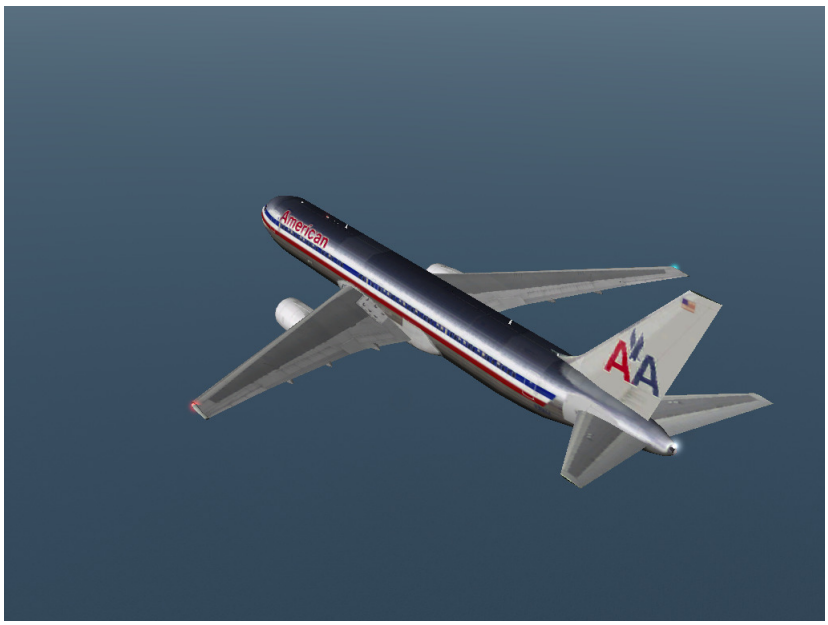


Figure B.1: X-Plane screenshot of the Boeing 767 model in flight



Figure B.2: Pilot's cockpit view in X-Plane.

Appendix C

Boeing 767 Flight Model Development

This chapter describes how the lift and drag characteristics of the Boeing 767 aircraft are determined. It is possible to estimate coefficients of lift and drag, C_L and C_D, solely from the geometry of the aircraft as described in Raymer [74]. However, another approach is to analyze the behaviour of a given airfoil shape and assume that is the airfoil in use on the aircraft. The latter method is selected because this better mirrors the approach that would be taken if the system is to be used with an actual aircraft, i.e. the data from wind tunnel tests of the real 767 airfoil shape would be used. A list of terms is provided at the end of the chapter (Table C.3) to aid the reader.

C.1 Calculation of Lift and Drag Polars Using XFoil

The XFoil code developed by Mark Drela for the design and analysis of subsonic isolated airfoils is used as a substitute for wind tunnel testing [22]. The inviscid formulation in the far field flow is determined using the linear-vorticity stream function panel method and the equations are closed with the Kutta condition. A compressibility correction is also added, however this breaks down in supersonic flow. The aircraft does not operate in the supersonic region therefore this is not considered to be a problem. The viscous region of the boundary layer and wake is described by a two-equation lagged dissipation integral boundary layer formulation. The interaction between the incompressible potential flow and viscous region is determined through the transpiration model. The code is known to work well in subsonic flows where there is little separation. As angle of attack is increased towards stall and the separation point moves forward along the airfoil, accuracy is diminished. This can be seen by comparison with experimental data. Therefore to predict lift and drag around the stall region correction factors are used.

The airfoil shapes of modern airliners are closely held secrets and therefore it is not possible to analyse an actual airfoil. They are known to use supercritical airfoils which are designed to delay the onset of wave drag in the transonic speed range. The aircraft is thus able to fly at higher speeds without the drag penalty associated with approaching Mach 1. For the purposes of this research a NACA 65-412 is assumed to be the airfoil in use. This airfoil is chosen as it has similar characteristics to a supercritical airfoil, although would not be as well optimised. This allows validation of the XFoil predictions with experimental data from Abbot and Von Doenhoff [5] which contains wind tunnel test results for families of NACA airfoils. Correction factors around the stall region are determined by comparison with the experimental data at the Reynolds number for which data is available. These correction factors are subsequently used with the XFoil code at the take-off and cruise Reynolds numbers of a 767 to give the coefficient of lift, drag and moment plots at these two extremes. The airfoil C_l and C_d data at the take-off Reynolds number is shown in Figure C.1. The X-plane simulation software calculates the Reynolds number based on the aircraft speed during each cycle and interpolates to find C_l and C_d from these plots. In the 3DOF model the data is approximated with polynomial functions. The C_l calculated by XFoil is applicable for an infinite wing. However, in reality aircraft have

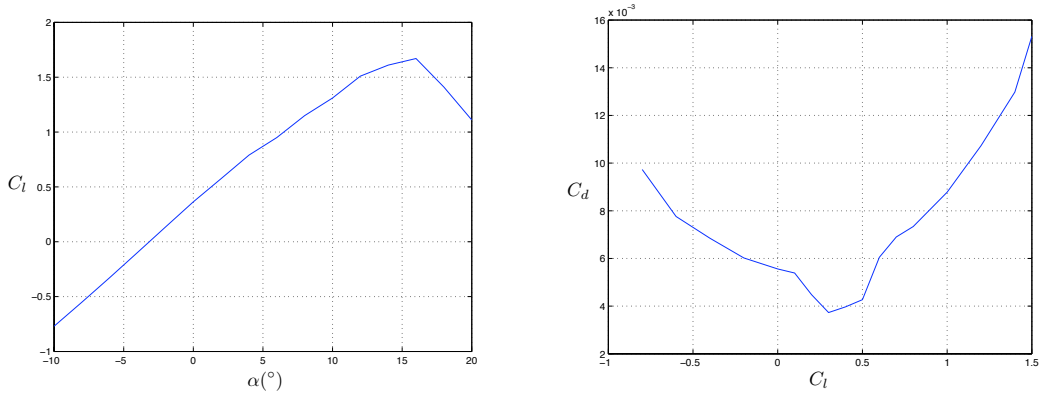


Figure C.1: Lift and drag curves, NACA 65-412 corrected at $Re = 10.5 \times 10^6$

finite wing spans and a correction is required to account for the lift reduction due to transfer of fluid from the upper to lower surface around the wing edge [74]. For aircraft with an aspect ratio (A) greater than four,

$$a = \frac{a_0}{\sqrt{1 + \frac{a_0}{\pi e A}}} \tag{C.1}$$

where a is the slope of the finite wing lift curve and a_0 is the slope of the infinite case given by XFoil. For swept-wing aircraft the Oswald span efficiency number (e) is based on data from actual aircraft [74] and given by

$$e = 4.61(1 - 0.045A^{0.68})(\cos \Lambda_{LE})^{0.15} - 3.1 \tag{C.2}$$

where Λ_{LE} is the sweep angle at the leading edge of the wing. During the model validation stage described later, e is tuned to improve the model match between the 3DOF model and X-Plane. Having calculated a, the finite wing lift coefficient C_L is given by

$$C_L = C_l \frac{a}{a_0} + C_{L_{fuse}} + C_{L_{tail}}, \tag{C.3}$$

where $C_{L_{fuse}}$ and $C_{L_{tail}}$ are the coefficients of lift for the fuselage and tail respectively. $C_{L_{fuse}}$ is approximated by $2(S_b)/S_{ref}$ [86], with S_b as the maximum cross-sectional area of the fuselage. The parameter $C_{L_{tail}}$ is a complex function of the tail's dynamic pressure and the rate of elevator deflection with angle of attack [86]. However, as the angular moments of the aircraft are not accounted for in the 3DOF model, $C_{L_{tail}}$ is simply neglected with it's influence incorporated into the tuned value for e from the wing contribution to lift.

The lift-induced drag for the finite wing C_D is approximated by

$$C_D = C_{D0} + KC_L^2. \tag{C.4}$$

Determination of the parasite drag C_{D_0} is described in the next section and the parameter K is approximated from XFoil data for this particular airfoil. Again, during the model validation stage K is adjusted for accuracy.

C.2 Calculation of Parasite Drag, C_{D_0}

As well as the lift induced drag calculated using XFoil in the previous section there are a number of other drag components that need to be considered. These are amalgamated into a single coefficient, C_{D_0}, which is only applicable at subsonic aircraft speeds (i.e. wave drag is not accounted for). The component buildup method found in Raymer [74] is used to determine C_{D_0}, with a calculated flat plate skin-friction drag coefficient (C_f) and a component form factor (FF) that estimates the pressure drag due to viscous separation. Interference effects on the component drag are estimated as a factor Q (determined from past aircraft design data) and the total component drag is determined as the product of the wetted area (S_{wet}), C_f, FF and Q:

$$(C_{D_0})_{subsonic} = \frac{\sum_{c=1}^{N} C_{f_c} FF_c Q_c S_{wet_c}}{S_{ref}} + C_{D_{misc}} + C_{D_{L\&P}}. \tag{C.5}$$

The c subscript denotes the component in question and N is the number of components. The parameters $C_{D_{misc}}$ and $C_{D_{L\&P}}$ are the miscellaneous and leakage & protuberance drag respectively. The aircraft model in the clean configuration includes skin friction contributions from the wings, tailplane, fin, fuselage and engine nacelles. The wetted areas are estimated using the geometry of the Boeing 767. The flat plate skin friction coefficient depends on the

Reynolds number, Mach number and skin roughness. For the purposes of the model, a single Reynolds number (Re_{cruise}) is used with turbulent flow assumed over the entire wetted surface. The drag coefficient C_f is then given by the following empirical relationship,

$$C_f = \frac{0.455}{(log_{10}R)^{2.58}(1+0.144M^2)^{0.65}}. \tag{C.6}$$

Form factors for subsonic drag estimation are given by:

$$FF = \left[1 + \frac{0.6}{(x/c)_m}\left(\frac{t}{c}\right) + 100\left(\frac{t}{c}\right)^4\right]\left[1.34M^{0.18}(\cos\Lambda_m)^{0.28}\right] \tag{C.7}$$

for the wing and tail,

$$FF = 1 + \frac{60}{f^3} + \frac{f}{400} \tag{C.8}$$

for the fuselage,

$$FF = 1 + \frac{0.35}{f} \tag{C.9}$$

for the nacelle, with

$$f = \frac{l}{d}, \tag{C.10}$$

where l/d is the fuselage length to diameter ratio, t/c is the thickness to chord ratio of the airfoil, x/c the chordwise location of the airfoil maximum thickness point and Λ_m is the sweep angle of the maximum thickness line.

Table C.1 gives the values found for the 767 geometry with data from [47]. The interference factor Q is taken from [74] which gives suitable values for a given aircraft component layout. The final two terms in (C.5) make up the remaining parasite drag. The coefficient $C_{D_{misc}}$

Component	C_f	FF	Q	$S_{wet}(m^2)$
Wing	0.0024	1.559	1.0	586.4
Tail	0.0025	1.454	1.05	155.6
Fin	0.0024	1.417	1.05	92.4
Fuselage	0.0018	1.082	1.0	720.9
Nacelle	0.0024	1.174	1.5	67.4

Table C.1: Component breakdown method for parasite drag.

accounts for factors which influence the drag that are not taken account of elsewhere. Airliners such as the 767 have an upsweep to the aft fuselage which contributes to drag. The extra drag is a function of the fuselage cross sectional shape and the aircraft angle of attack but can be approximated by

$$C_{D_{misc}} = 3.83u^{2.5}S_b, \tag{C.11}$$

where u is the upsweep angle of the aft section from the horizontal in radians [86]. Additional

factors contributing to $C_{D_{misc}}$ include the extra drag due to deployment of flaps and landing gear. In this work the aircraft is assumed to be operating in the clean configuration and so these can be ignored.

The coefficient $C_{D_{L\&P}}$ is the drag contribution from leaks and protuberances that are difficult to predict. Leakage drag is due to the tendency for the aircraft to 'suck in' air through holes and gaps in high pressure zones causing momentum loss, and to 'blow out' air in low pressure zones causing separation. Both these effects lead to an increase in drag. Protuberances include antennas, lights, door edges, fuel vents etc. which increase the drag. For a normal production aircraft, leaks and protuberance drags can be estimated as about 2-5% of the total parasite drag for jet transports [74].

C.3 Thrust Characteristics

The thrust produced by a turbofan engine is a function of the throttle setting, the altitude and Mach number of the aircraft. In the 3DOF aircraft model ((3.1d),(3.1e) and (3.1f)) thrust force is a direct input into the model. However in reality the throttle setting is the input selected by the pilot and the actual thrust is unknown during flight (manufacturers provide test data). Therefore it is necessary to model the engine accurately to determine how the thrust will vary given the aircraft state. This is done using code written by Mattingly [41].

Mattingly's code allows the user to design an engine for a specific operating point known as the 'on-x' condition and then determines the engine performance at other operating points referred to as 'off-x'. The Pratt and Whitney 4056 (PW4056) for the Boeing 767 was chosen as the engine to be modeled due to the availability of data for it. The installed performance of the engine is matched at a single operating point, the cruise conditions, and then parameters adjusted to ensure a match at a second operating point, the take-off condition. Table C.2 shows the values for the PW4056 used to produce the thrust data.

TO Thrust, kN	252.3
TO SFC, mg/Ns	10.170
Cruise Thrust, kN	50.4
Cruise SFC, mg/Ns	15.21
Overall pressure ratio	29.7
Fan pressure ratio	1.71
Bypass ratio	4.85
Total temperature (T_{t4}), K	1521
Mass flow rate ($\dot{m}$), kg/s	773

Table C.2: PW4056 engine parameters.

The Mattingly code predicts the specific fuel consumption (SFC) to be higher than that of the actual engine. This can be attributed to the fact that modern engines use various techniques

to lower fuel usage that are not accounted for in the older Mattingly code. Therefore a correction factor is used giving data for the SFC and thrust as a function of altitude and Mach number.

Symbol	Definition
Λ_{LE}	Sweep angle of the leading edge of the wing
A	Wing aspect ratio
a	Slope of the finite wing
a_0	Slope of the infinite wing
C_L	Coefficient of lift of the finite wing (3-D)
C_l	Coefficient of lift of the 2-D airfoil
$C_{L_{fuse}}$	Coefficient of lift due to fuselage
$C_{L_{tail}}$	Coefficient of lift due to tail
C_D	Coefficient of drag of the finite wing (3-D)
C_d	Coefficient of drag of the 2-D airfoil
C_{D_0}	Parasite drag
$C_{D_{misc}}$	Miscellaneous drag
$C_{D_{L\&P}}$	Leakage and protuberance drag
C_f	Flat plate skin-friction drag
d	Maximum diameter of fuselage
e	Oswald span efficiency number
FF	Form factor
K	Drag parameter
l	Length of fuselage
Q	Interference factor
S_b	Maximum fuselage cross-sectional area
S_{ref}	Reference wing area
S_{wet}	Wetted wing area
t/c	Thickness to chord ratio
x/c	Chordwise location of the maximum thickness point

Table C.3: List of terms.

Bibliography

[1] *767 Airplane Characteristics for Airport Planning, D6-583238, Boeing Manual.*

[2] Fidelity Flight Simulation Inc. http://www.fidelityflight.com. Last accessed 23 August 2010.

[3] Laminar Research. http://www.x-plane.com. Last accessed 23 August 2010.

[4] Preliminary incident report. Technical report, National Transportation Safety Board, http://www.faa.gov/data_research/accident_incident/1549, January 2009.

[5] I.H. Abbot and A.E. Von Doenhoff. *Theory of Wing Sections.* Dover Publications Inc., 1959.

[6] U.M Ascher, R.M.M. Mattheij, and R.D. Russell. *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations.* Society for Industrial and Applied Mathematics, 1995.

[7] R.A. Bartlett, A. Wächter, and L.T. Biegler. Active set vs. interior point strategies for model predictive control. In *Proceedings of the American Control Conference, Chicago, Illinois*, 2000.

[8] D.P. Bertsekas, A. Nedić, and A.E. Ozdaglar. *Convex Analysis and Optimization.* Athena Scientific, 2003.

[9] J.T. Betts. Survey of numerical methods for trajectory optimization. *Journal of Guidance, Control and Dynamics*, 21:193–207, 1998.

[10] J.T. Betts. *Practical Methods for Optimal Control Using Nonlinear Programming.* Society for Industrial and Applied Mathematics, 2001.

[11] F. Borrelli, D. Subramanian, A. U. Raghunathan, and L.T. Biegler. MILP and NLP techniques for centralized trajectory planning of multiple air vehicles. In *Proceedings of the American Control Conference*, 2006.

[12] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[13] B.C. Breen. Controlled flight into terrain and the enhanced ground proximity warning system. *IEEE AES Systems Magazine*, 1999.

[14] D.I.K. Brouwer, F.H. Grootendorst, C. Bedoya, M. Mulder, and M.M. van Paassen. Experimental evaluation of a safety augmentation system for commercial aircraft. In *Proceedings of the AIAA Guidance, Navigation and Control Conference, Providence, Rhode Island*, 2004.

[15] H. Chitsaz and S.M. LaValle. Time-optimal paths for a Dubins airplane. In *Proceedings of the 46th IEEE Conference on Decision and Control, New Orleans, Louisiana*, 2007.

[16] J.E. Corban, S. Twigg, T. Ries, B.J. Yang, E. Johnson, and A. Calise. On-line trajecotry optimization for autonomous air vehicles. Technical report, Air Force Office of Scientific Research, 2007.

[17] I.D. Cowling, O.A. Yakimenko, J.F. Whidborne, and A.K. Cook. A prototype of an autonomous controller for a quadrotor UAV. In *Proceedings of the European Control Conference, Kos, Greece*, 2007.

[18] C. Dever, B. Mettler, E. Feron, J. Popović, and M. McConley. Nonlinear trajectory generation for autonomous vehicles via parameterized maneuver classes. *Journal of Guidance, Control and Dynamics*, 29:289–302, 2006.

[19] M. Diehl, H.G. Bock, H. Diedam, and P. Wieber. Fast direct multiple shooting algorithms for optimal robot control. In *Fast Motions in Biomechanics and Robotics*, 2005.

[20] M. Diehl, H.G. Bock, and J.P. Schlöder. A real-time iteration scheme for nonlinear optimization in optimal feedback control. *SIAM Journal on Control & Optimization*, 43:1714–1736, 2004.

[21] M. Diehl, H.J. Ferreau, and N. Haverbeke. Efficient numerical methods for nonlinear MPC and moving horizon estimation. In *International Workshop on Assessment and Future Directions of NMPC, Pavia, Italy*, 2008.

[22] M. Drela. Xfoil - An analysis and design system for low reynolds number airfoils. In *Low Reynolds Number Aerodynamics; Proceedings of a Conference (Lecture Notes in Engineering)*, 1989.

[23] L.E. Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal postions and tangents. *American Journal of Mathematics*, 79:497–516, 1957.

[24] G.J.J. Ducard. *Fault-tolerant Flight Control and Guidance Systems*. Springer, 2009.

[25] D. Ensor, J. King, S. MacVicar, B. Starr, and T. Frieden. Missiles, fighters ready to defend D.C. *CNN.com*, http://157.166.255.31/2003/US/05/21/terror.main/index.html, 2003. Last accessed 23 August 2010.

[26] D. Ernst, K. Valavanis, R. Garcia, and J. Craighead. Unmanned vehicle controller design, evaluation and implementation: From matlab to printed circuit board. *Journal of Intelligent and Robotic Systems*, 49:85–108, 2007.

[27] D. Evans. Anti-hijacking avionics. *Avionics Magazine*, pages 53–54, 2001.

[28] D. Evans. Robolander revisited. *Avionics Magazine*, pages 45–46, 2002.

[29] R. Fletcher. *Practical Methods of Optimization.* John Wiley & Sons, Inc., 1987.

[30] E. Frazzoli, M.A. Dahleh, and E. Feron. A hybrid control architecture for agressive maneuvering of autonomous helicopters. In *Proceedings of the 38th Conference on Decision and Control, Pheonix, Arizona*, 1999.

[31] E. Frazzoli, M.A. Dahleh, and E. Feron. Real-time motion planning for agile autonomous vehicles. *Journal of Guidance, Control and Dynamics*, 25:116–129, 2002.

[32] P.E. Gill, W. Murray, and M.H. Wright. *Practical Optimization.* Academic Press, 1981.

[33] C. Goerzen, Z. Kong, and B. Mettler. A survey of motion planning algorithms from the perspective of autonomous UAV guidance. *Journal of Intelligent and Robotic Systems*, 57:65–100, 2010.

[34] J. Gondzio. Warm start of the primal-dual method applied in the cutting-plane scheme. *Mathematical Programming*, 83:125–143, 1998.

[35] K.H. Goodrich, P.C. Schutte, F.O. Flemisch, and R.A. Williams. Application of the h-mode, a design and interaction concept for highly automated vehicles, to aircraft. In *Proceedings of the 25th Digital Avionics Conference, Portland, Oregon*, 2006.

[36] K.H. Goodrich, P.C. Schutte, and R.A. Williams. Piloted evaluation of the h-mode, a variable autonomy control system, in motion-based simulation. In *Proceedings of the AIAA Guidance, Navigation and Control Conference, Honolulu, Hawaii*, 2008.

[37] P.J. Goulart. *Affine Feedback Policies for Robust Control with Constraints.* PhD thesis, University of Cambridge, 2007.

[38] F.H. Grootendorst, D.I.K. Brouwer, M.M. van Paassen, and M. Mulder. Design of a safety augmentation system for commercial aircraft. In *Proceedings of the AIAA Guidance, Navigation and Control Conference, Providence, Rhode Island*, 2004.

[39] H.L. Hagenaars, J. Imura, and H. Nijmeijer. Approximate continuous-time optimal control in obstacle avoidance by time/space discretization of non-convex state constraints. In *Proceedings of the IEEE International Conference on Control Applications, Taipei, Taiwan*, 2004.

[40] A.S. Hahn. Next generation NASA GA advanced concept. Technical report, Nasa Langley Research Center Report, 2006.

[41] W.H. Heiser, D.T. Pratt, and J.D. Mattingly. *Aircraft Engine Design, Second Edition.* AIAA Education Series, 2002.

[42] A.C. Hindmarsh and R. Serban. User documentation for CVODES v2.5.0. 2006.

[43] J. Hoffren and T. Sailaranta. Maneuver autopilot for realistic performance model simulations. In *Proceedings of the AIAA Modeling and Simulation Technologies Conference, Montreal, Canada*, 2001.

[44] G.J. Holland, P.J. Webster, J.A. Curry, G. Tyrell, D. Gauntlett, G. Brett, J. Becker, R. Hoag, and W. Vaglienti. The aerosonde robotic aircraft: A new paradigm for environmental observations. *Bulletin of the American Meteorological Scoiety*, 82:889–901, 2001.

[45] P. Huber. Disable the pilots. *Forbes Magazine*, 2001.

[46] A.Z. Ibsen. The politics of airplane production: The emergence of two technological frames in the competition between boeing and airbus. *Technology in Society*, 31:342–349, 2009.

[47] P.A. Jackson, editor. *Jane's All the World's Aircraft.* Jane's Information Group, 2006.

[48] E.N. Johnson and A.J. Calise. A six degree-of-freedom adaptive flight control architecture for trajectory following. In *Proceedings of the AIAA Guidance, Navigation and Control Conference, Monterey, California*, 2002.

[49] K.B. Judd and T.W. McLain. Spine based planning for unmanned air vehicles. In *Proceedings of the AIAA Guidance, Navigation and Control Conference, Montreal, Cananda*, 2000.

[50] E.C. Kerrigan and J.M. Maciejowski. Soft constraints and exact penalty functions in model predictive control. In *Control 2000 Conference, Cambridge*, 2000.

[51] T. Keviczky and G.J. Balas. Software enabled flight control using receding horizon techniques. In *Proceedings of the AIAA Guidance, Navigation and Control Conference, Austin, Texas*, 2003.

[52] T. Keviczky and G.J. Balas. Receding horizon control of an F-16 aircraft: a comparative study. *Control Engineering Practice*, 14:1023–1033, 2006.

[53] H.J. Kim, D.H. Shim, and S Sastry. Nonlinear model predictive tracking control for rotorcraft-based unmanned aerial vehicles. In *Proceedings of the American Control Conference, Anchorage, Alaska*, 2002.

[54] Z. Kopal. *Numerical Analysis*. Chapman and Hall Ltd., 1961.

[55] J.K. Kuchar and L.C. Yang. A review of conflict detection and resolution modeling methods. *IEEE Transactions On Intelligent Transportation Systems*, 1:179–189, 2000.

[56] Y. Kuwata, T. Schouwenaars, A. Richards, and J. How. Robust constrained receding horizon control for trajectroy planning. In *Proceedings of the AIAA Guidance, Navigation and Control Conference, San Franciso, California*, 2005.

[57] S.M. LaValle. Rapidly-exploring random trees: A new tool for path planning. Technical report, Computer Science Department, Iowa State University, 1998.

[58] S.M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

[59] S.M. LaValle and J.J. Kuffner Jr. Randomized kinodynamic programming. *The International Journal of Robotics Research*, 20:378–400, 2001.

[60] S.M. LaValle and P. Konkimalla. Algorithms for computing numerical optimal feedback motion strategies. *The International Journal of Robotics Research*, 20:729–752, 2001.

[61] E.A. Lee. Soft walls - Modifying flight control sytems to limit the flight space of commercial aircraft. Technical report, UC Berkeley, 2001.

[62] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36:789–814, 2000.

[63] A. Miele. *Flight Mechanics*. Addison-Wesley Pub. Co., 1962.

[64] I. Moir and A. Seabridge. *Civil Avionics Systems*. Professional Engineering Publishing, 2003.

[65] J.F. Monaco, D.G. Ward, and A.G. Barto. Automated aircraft recovery via reinforcement learning: Initial experiments. In *Proceedings of the 1997 Conference on Advances in Neural Information Processing Systems*, 1997.

[66] Y. Nesterov. *Introductory Lectures on Convex Optimization*. Kluwer Academic Publishers, 2004.

[67] M.J. Van Nieuwstadt and R.M. Murray. Real-time trajectory generation for differentially flat systems. *Robust and Nonlinear Control*, 8:995–1020, 1998.

[68] S. Park, J. Deyst, and J.P. How. Performance and Lyapunov stability of a nonlinear path-following guidance methods. *Journal of Guidance, Control and Dynamics*, 30:1718–1728, 2007.

[69] K. J. Plitt and H. G. Bock. A multiple shooting algorithm for direct solution of optimal control problems. *International Federation of Automatic Control*, 4, 1984.

[70] D.T. Pratt, W.H. Heiser, and J.D. Mattingly. *Aircraft Engine Design.* AIAA, 2002.

[71] R.W. Pratt, editor. *Flight Control Systems.* The Institution of Electrical Engineers, 2000.

[72] S.V. Raković and D.Q. Mayne. Robust time optimal obstacle avoidance problem for constrained discrete time systems. In *Proceedings of the 44th IEEE Conference on Decision and Control, Seville, Spain*, 2005.

[73] M. Rathinam and R.M. Murray. Configuration flatness of Lagrangian systems underactuated by one control. *SIAM Journal on Control & Optimization*, 36:164–179, 1998.

[74] D.P. Raymer. *Aircraft Design: A Conceptual Approach. Third Edition.* AIAA Education Series, 1999.

[75] J. Redding, J.N. Amin, and J.D. Bošković. A real-time obstacle detection and reactive path planning system for autonomous small-scale helicopters. In *Proceedings of the AIAA Guidance, Navigation and Control Conference, South Carolina,*, 2007.

[76] A. Richards, Y. Kuwata, and J. How. Experimental demonstrations of real-time MILP control. In *Proceedings of the AIAA Guidance, Navigation and Control Conference*, 2003.

[77] A.G. Richards and J.P. How. Aircraft trajectory planning with collision avoidance using mixed integer linear programming. In *Proceedings of the American Control Conference, Anchorage, Alaska*, 2002.

[78] R.T. Rockafellar. *Convex Analysis.* Princeton University Press, 1970.

[79] R.T. Rockafellar and R.J.B. Wets. *Variational Analysis.* Springer-Verlag, 2004.

[80] R. Rogers. Flight test results of the controlled flight into terrain avoidance maneuver in fly-by-wire transports. Technical report, Air Line Pilots Association, 1999.

[81] A. Roldao-Lopes, A. Shahzad, G.A. Constantinides, and E.C. Kerrigan. More flops or more precision? Accuracy parameterizable linear equation solvers for model predictive

control. In *17th IEEE Symposium on Field Programmable Custom Computing Machines, Napa, California*, 2009.

[82] E. Rosenberg. Exact penalty functions and stability in locally Lipschitz programming. *Mathematical Programming*, 30:340–356, 1984.

[83] Z. Schiller. On-line sub-optimal obstacle avoidance. In *Proceedings of the IEEE International Conference on Robotics and Automation, Detroit, Michigan*, 1999.

[84] N. Slegers, J. Kyle, and M. Costello. Nonlinear model predictive control technique for unmanned air vehicles. *Journal of Guidance, Control and Dynamics*, 29:1179–1188, 2006.

[85] R.J. Starling and C.M. Stewart. The development of terrain following radar: An account of the progress made with an airborne guidance system for low flying miliary aircraft. *Aircraft Engineering and Aerospace Technology*, 43:13–15, 1993.

[86] R.F. Stengel. *Flight Dynamics*. Princeton University Press, 2004.

[87] B.L. Stevens and F.L. Lewis. *Aircraft Control and Simulation*. John Wiley & Sons, Inc., 1992.

[88] S. Sundar and Z. Schiller. Time-optimal obstacle avoidance. In *Proceedings of the IEEE International Conference on Robotics and Automation, Nagoya, Japan*, 1995.

[89] A.S. Usmani, Y.C. Chung, and J.L. Torero. How did the WTC towers collapse: A new theory. *Fire Safety Journal*, 38:501–533, 2003.

[90] A. Voelker, M. Dördelmann, K. Kouramas, E.N. Pistikopoulos, M. Shanmugavel, A. Tsourdos, B.A. White, and R. Żbikowski. Three-layer aircraft control: Path-planning, guidance, model predictive controller as autopilot. In *18th IFAC Symposium on Automatic Control in Aerospace (ACA), Nara, Japan*, 2010.

[91] A. Wächter and L.T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106:25–57, 2006.

[92] G. Wanka, R.I. Bot, and E. Vargyas. Duality for location problems with unbounded unit balls. *European Journal of Operational Research*, 179:1252–1265, 2007.

[93] P.J. Wong and A.J. Korsak. Reachable sets for tracking. *Operations Research*, 22:497–509, 1974.

[94] S. Wright and J. Nocedal. *Numerical Optimization*. Springer, 1999.

[95] S.J. Wright. *Primal-Dual Interior-Point Methods*. Society for Industrial Mathematics, 1997.

[96] Y.C. Wu and A.S. Debs. Initialisation, decoupling, hot start, and warm start in direct nonlinear interior point algorithm for optimal power flows. *IEE Proceedings - Generation, Transmission and Distribution*, 148:67–75, 2001.

[97] O.A. Yakimenko. Direct method for rapid prototyping of near-optimal aircraft trajectories. *Journal of Guidance, Control and Dynamics*, 23:865–875, 2000.

[98] H.I. Yang and Y.J. Zhao. Trajectory planning for autonomous aerospace vehicles amid known obstacles and conflicts. *Journal of Guidance, Control and Dynamics*, 27:997–1008, 2004.

[99] H.M. Youssef, K.G. Gousman, and S.R. Combs. Fuzzy logic approach to automatic recovery system. In *Proceedings of the IEEE 1995 National Aerospace and Electronics Conference*, 1995.