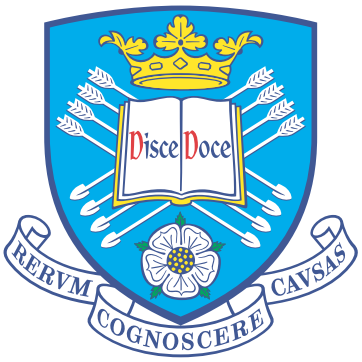
The University of Sheffield

Dept. of Automatic Control and Systems Engineering
Faculty of Engineering



Multi-agent Collision Avoidance Using Interval Analysis and Symbolic Modelling with its Application to the Novel Polycopter

James Alexander Douthwaite

A thesis submitted in partial fulfilment
of the requirements for the degree of
Doctor of Philosophy.

April 2020

1^{st} Supervisor: Prof. Lyudmila S. Mihaylova
2^{nd} Supervisor: Dr. Shiyu Zhao
3^{rd} Supervisor: Prof. Sandor Veres

This thesis is dedicated to my family, and
to Nicole, for their unconditional love,
support and encouragement.

In loving memory of Margret Douthwaite.

Abstract

Coordination is fundamental component of autonomy when a system is defined by multiple mobile agents. For unmanned aerial systems (UAS), challenges originate from their low-level systems, such as their flight dynamics, which are often complex. The thesis begins by examining these low-level dynamics in an analysis of several well known UAS using a novel symbolic component-based framework. It is shown how this approach is used effectively to define key model and performance properties necessary of UAS trajectory control. This is demonstrated initially under the context of linear quadratic regulation (LQR) and model predictive control (MPC) of a quadcopter.

The symbolic framework is later extended in the proposal of a novel UAS platform, referred to as the "Polycopter" for its morphing nature. This dual-tilt axis system has unique authority over is thrust vector, in addition to an ability to actively augment its stability and aerodynamic characteristics. This presents several opportunities in exploitative control design.

With an approach to low-level UAS modelling and control proposed, the focus of the thesis shifts to investigate the challenges associated with local trajectory generation for the purpose of multi-agent collision avoidance. This begins with a novel survey of the state-of-the-art geometric approaches with respect to performance, scalability and tolerance to uncertainty. From this survey, the interval avoidance (IA) method is proposed, to incorporate trajectory uncertainty in the geometric derivation of escape trajectories. The method is shown to be more effective in ensuring safe separation in several of the presented conditions, however performance is shown to deteriorate in denser conflicts.

Finally, it is shown how by re-framing the IA problem, three dimensional (3D) collision avoidance is achieved. The novel 3D IA method is shown to out perform the original method in three conflict cases by maintaining separation under the effects of uncertainty and in scenarios with multiple obstacles. The performance, scalability and uncertainty tolerance of each presented method is then examined in a set of scenarios resembling typical coordinated UAS operations in an exhaustive Monte-Carlo analysis.

Acknowledgements

First and foremost I would like to sincerely thank my research supervisor Prof. Lyudmila Mihaylova for her continued encouragement and support. Together, her knowledge and experience has been invaluable in helping coordinate and ultimately decide my direction of research. On a non-academic note, Mila's advice and support during my time at Sheffield has helped me persevere through some moments of hardship, while her enthusiasm has lead me to consider taking further steps towards a career in research.

Secondly I would like to thank Dr Shiyu Zhao, for joining us on this project and the experience and rigour he was able to bring - albeit in the later stages of the project. Shiyu's drive to see me produce my best helped tremendously; in the breaking down of new concepts and ideas, whilst having the confidence to challenge opinions when improvement could be found.

My time as a Ph.D student would have been very different without the contributions of Dr James Law and Dr Johnathan Aitken. The number of second opinions, enthusiastic discussions and opportunities they have offered has helped greatly shape my experience as postgraduate and as an aspiring academic. Also, to my colleagues at Sheffield Robotics; Matthew Watson, Jõao Marques and Matthew Doyle, who shared in my experience and whose friendship has helped me persevere.

I would also like to thank my girlfriend, Nicole Salomons, who joined me in this adventure. Her patience and understanding, whilst completing a Ph.D of her own, never ceases to amaze me. I owe a great deal of gratitude to her, to my parents, and to my sister Katie, who together made this work possible through their continued love and encouragement.

Finally, I would like to acknowledge the financial support provided by the European and Physical Sciences Research Council (EPSRC) under grant number EP/M506618/1.

Contents

List of Nomenclature

Acronyms & Abbreviations

GPS	Global Position Satellite System
LOAM	Laser Obstacle Avoidance Marconi
2D/3D	Two/Three-Dimensions
AABB	Axis-Aligned Bounding Box
ADS-B	Automatic Dependent Surveillance - Broadcast
AFRL	United States Air-force Research Laboratory
ATM	Air Traffic Management
AVO	Acceleration Velocity Obstacle
CAA	Civil Aviation Authority
CG	Centre of Gravity
DARPA	The Defence Advanced Research Projects Agency
EFR	Extended Flight Rules
ELOS	Equivalent Level of Safety
ENU	East-North-Up Coordinates
ESC	Electronic Speed Controller
FAA	Federal Aviation Authority
FOV	Field of View
GA	Genetic Algorithm

GCAS	Ground Collision Avoidance System
HRVO	Hybrid-Reciprocal Velocity Obstacle
IA	Interval Avoidance
IMU	Inertial Measurement Unit
KVO	Kinematic Velocity Obstacle
LIDAR	Light Detection And Ranging
LQG/LQR	Linear Quadratic Gaussian/Regulation
LTV	Linear Time-Variant
MAS	Multi-Agent System
MAV	Micro Aerial Vehicle
MDP	Markov Decision Process
MPC	Model Predictive Control
MTOW	Maximum Take-Off Weight
NASA	National Aeronautical and Space Agency
NATS	National Air Traffic System
NDB	Non-directional Beacons
NED	North-East-Down Coordinates
NN	Neural Network
OBB	Object-Aligned Bounding Box
OpenMAS	Open-source Multi-Agent Simulator
ORCA	Optimal Reciprocal Collision Avoidance
PCB	Printed Circuit Board
PID	Proportional, Integral, Differential Control
PWM	Pulse-Width Modulation
RADAR	Radio Detection And Ranging

RNAV	Area navigation
RNE	Recursive Newton-Euler Method
RPA	Remotely Piloted Aircraft
RVO	Reciprocal Velocity Obstacle
S&L	Straight & Level
SDA	Sense, Detect and Avoid
SISO	Single Input, Single Output systems
SMC	Sliding-Mode Control
TCAS	Traffic Collision Avoidance System
TLOS	Target Level of Safety
UAS	Unmanned Aerial System
UAV	Unmanned Aerial Vehicle
VFR	Visual Flight Rules
VO	Velocity Obstacle
VS	Vector Sharing

Mathematical Terms

$[\boldsymbol{a}_{i,max}]$	Viable acceleration interval for agent i	$m \cdot s^{-2}$
$[\boldsymbol{r}_m]$	Geometric miss interval	m
$[\boldsymbol{r}_{ca}]$	Point of closest approach	m
$[\boldsymbol{r}_{vsi,vsj}]$	The shared resolution intervals for agents i and j	m
$[\boldsymbol{U}_i]$	The resolution interval vector of i	$m \cdot s^{-1}$
$[\boldsymbol{U}_i^*]$	The interval of mutual avoidance for agent i	$m \cdot s^{-1}$
$[\tau]$	Time to collision interval	s
$[\theta_i], [\psi_i]$	The elevation and heading interval	rad
$[r_{ca}]$	Interval of closest approach	m

α_j	Angular width in the azimuth j	rad
$\boldsymbol{\chi}_i$	The state of object i in the inertial frame	–
$\boldsymbol{\delta}, \boldsymbol{\eta}$	Nacelle pitch and roll deflection vectors	rad
$\boldsymbol{\nu}_{aero}, \boldsymbol{\nu}_{grav}$	Symbolic aerodynamic and gravitational forces	N, Nm
$\boldsymbol{\nu}_{prop}, \boldsymbol{\nu}_{gyro}$	Symbolic propulsive and gyroscopic forces	N, Nm
$\boldsymbol{\omega}_i, \dot{\boldsymbol{\omega}}_i$	Angular velocity and acceleration of i	$rads^{-1}, rad^{-2}$
$\boldsymbol{\Theta}$	Instantaneous Euler pose of the airframe	rad
$\boldsymbol{f}_{r_i}, \boldsymbol{\tau}_{r_i}$	Thrust and hub torque enacted by rotor i	N
$\boldsymbol{l}_i$	Geometric length of component i	m
$\boldsymbol{p}_i, \boldsymbol{v}_i, \boldsymbol{a}_i$	Cartesian position, velocity and acceleration of i	–
$\boldsymbol{p}_{cg}$	Centre of gravity position	m
$\boldsymbol{q}$	Quaternion or generalised coordinate vector	–
$\boldsymbol{r}, \boldsymbol{e}$	State reference and error	–
$\boldsymbol{r}_j^i, \boldsymbol{r}_c^i$	Vector from link i to link j and i to the CG of i	m
$\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}$	The dynamical state, input and noise vectors	–
$\ddot{\boldsymbol{p}}_i^c$	Linear acceleration of the CG of i	$m \cdot s^{-1}, m \cdot s^{-2}$
Δt	Sample period	s
$\dot{\boldsymbol{p}}_i, \ddot{\boldsymbol{p}}_i$	Linear velocity and acceleration of i	$m \cdot s^{-1}, m \cdot s^{-2}$
Γ	Christoffel Symbol	–
γ_i	Rotation of rotor i	–
$\hat{\boldsymbol{n}}^{xy}, \hat{\boldsymbol{n}}^{xz}$	XY and XZ planar normal vectors	m
$\hat{\boldsymbol{z}}$	Component axis of revolution	m
ι	Object separation/collision tolerance.	m
κ_i	The rotation direction of rotor i	–
λ_i	The angular position of rotor i	rad

$\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$	State space coefficient matrices	–
$\mathbf{A}_i$, $\mathbf{W}_i$, $\mathbf{O}_i$	Agent, obstacle and waypoint vectors	–
$\mathbf{F}, \mathbf{G}$	Plant and input horizon prediction matrices	–
$\mathbf{I}_i$	Inertia tensor of component i	kgm^2
$\mathbf{K}$	Optimal feedback matrix	–
$\mathbf{M}, \mathbf{C}$	Inertial and Coriolis/centripetal matrices	–
$\mathbf{Q}, \mathbf{R}, \mathbf{N}$	State error, input and terminal state weighting matrices	–
$\mathbf{R}_c^b$	Body axes to control axes transformation	–
$\mathbf{R}_j^i$	Rotation matrix from frame i to frame j	–
Ω_{r_i}	Rotational speed of rotor i	rad
ϕ, θ, ψ	Roll, pitch and yaw Euler angles	rad
ρ	Air density	$kg \cdot m^{-3}$
σ_k	The quaternion normalising components at step k	–
θ_j, ψ_j	Spherical positions of obstacle j	rad
a	Rotor cross-sectional area	m^2
C_t, C_h	Thrust and hub force coefficients	$N \cdot rad^{-2}$
g	Gravitational constant	$m \cdot s^{-2}$
h	Prediction horizon	–
H_a	Acceleration Hamiltonian	$m \cdot s^{-2}$
k	Discrete step	–
k_p, k_h	Thrust and hub force constants	$N \cdot rad^{-2}$
m_i	Mass of component i	kg
O_i	Coordinate frame of component i	–
p, q, r	Angular velocity components	$rad \cdot s^{-1}$
r_{ca}	Point of closest approach	m

t_k	Discrete time step	s
u, v, w	Linear velocity components	$m \cdot s^{-1}$

Notation

$\boldsymbol{x}$	Vector parameter
$\hat{\boldsymbol{x}}$	Unit vector parameter
$\mathbf{X}$	Matrix parameter
$\tilde{x}$	Measurement of x
x^*	The optimal value of x
x^{ϕ},x^{θ},x^{ψ}	The Euler rotational dimensions
x^x, x^y, x^z, x^w	Cartesian dimensions
x_b,x_c	Body/control coordinate frame
x_H, x_V	Horizontal/vertical elements of x
x_i, x_j	Agent i and j
x_k	Discrete time step
x_{min},x_{max}	Minimum/maximum x value
x_{pref}, x_{wp}	Preferred or way-point property

List of Figures

List of Tables

Chapter 1

Introduction

Over the last two decades numerous technological advances in the field of aerial robotics have meant that unmanned systems are now being considered for applications in almost every sector. This is partly due to increase in public interest in such systems, but also the availability of inexpensive yet highly sophisticated hardware. It is clear that with the continued advent of increasingly powerful compact devices coupled with widely available open-source software, the applications for *unmanned aerial vehicles* (UAV) and the broader classification of *unmanned aerial systems* (UAS) are set to expand [175].

Today such systems predominantly fall into two classifications i) *remotely piloted aircraft* (RPA) and ii) *autonomous aerial vehicles* (AAV); based on the systems interaction with the operator and their corresponding level of sophistication [175]. Within these classifications the term *autonomy* can be used to describe multiple levels of capability as seen in Figure 1.1; with more complex systems supporting advanced levels of coordination, failure tolerance and adaptive behaviour [248, 249].

For both RPA and AAV systems, the ability to command a pose of trajectory is fundamental for basic locomotion. To achieve this there is a prerequisite understanding of the system's flight dynamics before an effective control strategy can be introduced. This type of low-level control is often not a trivial task. A result of the manoeuvrability of UAVs and increasing degrees of freedom, the dynamics are often complex, nonlinear and unstable. For this reason, effective approaches that provide a standard for analysing various UAS topologies and dynamics are becoming increasingly necessary. This is a subject that is addressed as part of these works, in the proposal of a novel framework for UAS modelling and control design.

A distinction is therefore drawn between the low-level control necessary for locomotion, and the higher-level control associated with autonomous behaviour. While both

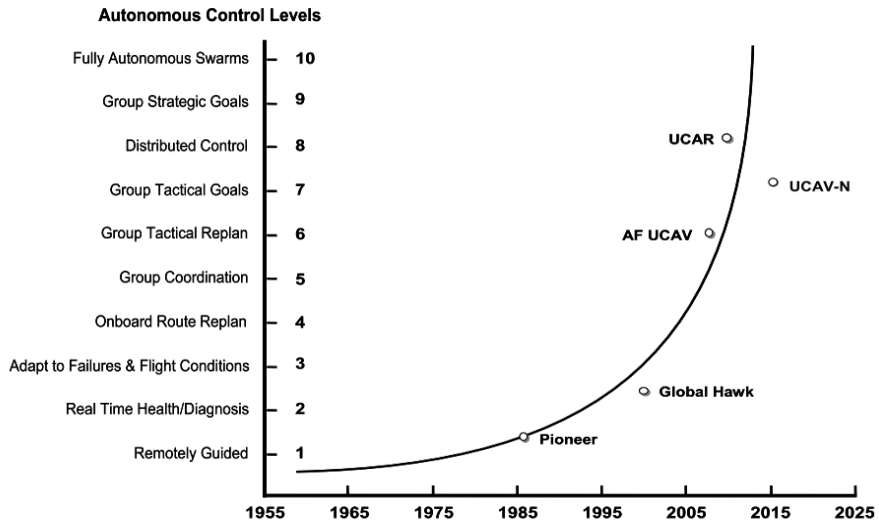


Figure 1.1: *Categorisation of several well known* unmanned aerial systems *(UAS) based on their functionality and level of autonomy. Coordination refers to several control levels, low level control 1 & 2, with more sophisticated group behaviour defined as levels 5 and above. Fundamental operations such as collision avoidance and adaptive re-planning are defined as levels 3 & 4 [248, 249].*

are implicit in the operation of AAVs, there exists a another set of challenges once control over the vehicles dynamics can be assumed. These challenges are associated with establishing a critical level of autonomy required for versatile and reliable operation of the AAV with limited interaction with a human operator. Some of the factors contributing to this critical level of autonomy are reliable mechanisms for automatic collision avoidance, failure tolerance, communication and coordination with other systems and AAVs.

An example where a critical level of autonomy must be achieved is in the utilisation of swarms of AAVs as *multi-agent systems* (MAS). Here the system is comprised of many collaborating physical systems with some overarching objective defined by a common schema. In such systems low-level control of the individual agent is often assumed and tasks such as group coordination, collision avoidance and adaptive failure handling become integral to the MAS. This is because in such scenarios it becomes favourable to interact with the MAS at a supervisory level; as meaningful control of each individual is difficult if not impossible. There is therefore a reliance on a critical level of autonomy to be able to manage low-level operations, communication and decisions. The shift in responsibility then enables the supervisor to focus on the mission objectives whilst operations such as command interpretation, collision avoidance and agent coordination become behaviour inherent to the MAS [5, 8, 175].

Once low-level control can be assumed, one of the distinct components limiting the autonomy of MAS and coordinated UAV systems is reliable and automatic collision

avoidance. In this thesis, the challenges and requirements for modern UAS operating in a modern airspace are examined from the perspective of collision avoidance within coordinated UAV systems. As part of the investigation, reference is frequently made to existing standards and methods applied to conventional air-traffic control. This is due to the fact that many of the challenges faced in the coordination of multiple UAVs(MASs) can, in a broader sense, be considered analogous to those faced by modern air-traffic control. The assumption in such cases is that trajectory information is available unilaterally across an established network [104, 123].

In the event that communication is not unilateral, or cannot be guaranteed, then modern AAVs are required to have *sense, detect and avoid* (SDA) mechanisms in place in order to be able to operate alongside conventional manned aircraft [11, 225, 228]. Historically, the SDA approach has been presented in the context of conventional, manned, air-traffic as a basis for local conflict negotiation. Interest in this area originally stems from the redundancy of any a dedicated *air traffic management* (ATM) service in favour of conflict resolution occurring on a peer to peer basis. This concept is widely cited in the literature as the *free flight* principle [77, 106, 204, 229]. The need for more sophisticated automation tools for handling aerial conflicts due to increasing air-traffic is already highlighted in [123, 184] without reference to the emerging body of unmanned air-traffic.

Its clear from the challenges faced by modern UAS that to both i) contribute towards achieving a critical level of autonomy through the design of a reliable collision avoidance method ii) meet the requirements for safe operation in a mixed airspace, a reliable SDA procedure is required. This is the focus of the later chapters of these works. The use of SDA based avoidance in coordinated UAS is well established in the literature and naturally rely on sophisticated sensing methods to provide obstacle information that cannot be otherwise communicated or inferred. While SDA methods make no assumptions about the communication capabilities of the obstacle, they present several opportunities for approaches that can handle multiple obstacles and the uncertainty associated with realistic sensing conditions.

1.1 Challenges & Opportunities

The potential for MAS and collaborative systems over conventional UAS (such as RPAs) is extensive and well documented. Coordination of multiple agents allows for more optimal coverage in search and rescue missions, optimality in the distribution of mobile

sensor networks, capacity in transport and reliability through redundancy to name a few. In addition, collaborative systems bring about new opportunities for systems in the areas of multi-tasking and delegation, or in the completion of tasks that cannot be negotiated without collaboration. With their benefits however, comes numerous challenges to overcome in ensuring reliable communication, scalability, effective coordination and safety assurance. Ensuring safety in multi-agent systems is a complex task, referring not just to the safe operation of the system but also with respect to non-cooperative entities such as foreign aircraft, bystanders and obstacles that may be unable to communicate their intent.

One of the most fundamental tasks necessary in establishing a basic level of safety in collaborative systems is effective collision prevention, both at a level of coordination and in response to unforeseen changes. It is therefore clear that effective SDA algorithms able to tolerate real-world sensor conditions demonstrate an immediate opportunity to enhance the autonomy of swarming AAVs and UAS.

1.2 Thesis Outline

The thesis structure is outlined below:

Chapter 1 begins by introducing the motivation and context for the research presented in this thesis. Some emphasis is also given to some of the challenges faced in the enhancement autonomy within coordinated UAS. The structure of the document is also outlined in addition to the key contributions and publications associated with the thesis.

Chapter 2 presents a comprehensive review of the techniques currently applied in the coordination of unmanned aerial systems. It begins by introducing common control topologies in relation to the various UAV and UAS subsystems. Established axes conventions, notation and principles used to describe UAS are then presented for the reader. Focus then shifts to the discuss many of the control techniques currently being applied in the context of UAS control. Classical control notions are introduced in relation to more modern techniques for establishing stability, trajectory control and more sophisticated techniques for adaptive behaviour and failure tolerance in modern unmanned systems.

Chapter 3 begins with an introduction to *OpenMAS*; an open-source multi-agent simulator developed as part of these works. Here, the mathematical representation of UAS pose in both 2D and 3D space is presented in relation to flight-path notation, collision events, and other fundamental procedures. An overview of the structural representation of

agents, technical assumptions is given in addition to a description of techniques employed to emulate data received from on-board sensors.

The chapter then moves onto the proposed symbolic framework for the generation of analytic UAS and *mirco aerial vehicle* (MAV) descriptions. It is demonstrated how using this framework several known MAV configurations may be characterised by defining their symbolic representation. Additionally it is shown how this technique may be incorporated into preliminary controller design by defining key control and stability parameters without numeric parameterisation. This principle can be seen initially applied to quadcopter configurations; in which stability is demonstrated using both *linear quadratic regulator* (LQR) and *model predictive control* (MPC) methodologies. To further demonstrate the technique, the symbolic representation of tricopter and deltacopter MAV configurations are presented. These more sophisticated MAV descriptions are then presented to highlight further opportunities for future work.

Chapter 4 introduces a novel MAV design based on the symbolic framework introduced previously. This new topology, referred to as the "Polycopter", presents a number interesting motion characteristics due to the active morphing airframe and it's three independently vectored nacelles. The chapter initially presents some of the background literature surrounding morphing aerial systems and their distinction from traditional MAV topologies. The opportunities these systems present are then discussed with reference to the flight characteristics of the Polycopter.

The chapter then presents derivations of the proposed systems dynamics under two conditions. The first presents a reduced form of the systems dynamics under the assumptions that the deflections of the nacelles are small; and so the inertial characteristics of the system are largely unchanging. This is shown to be synonymous to several existing dual-tilt axis systems where only the rotor-disk is actuated. This assumption is then challenged by considering each of the nacelle assemblies as serial-link kinematic chains in order to better represent the system through larger control-surface deflections. The chapter concludes with closing remarks about the numerous opportunities the system presents for adaptive and bio-mimetic control that may be able to take advantage of the MAV's versatility in future work.

Chapter 5 presents a review of literature surrounding collision avoidance in the context of unmanned aerial systems. Here several fundamental principles and concepts are introduced, in addition to some common assumptions within the multi-agent literature. A review of established methods to collision avoidance is presented based on their description of the conflict problem. An emphasis on geometric techniques can also be seen in light of

the contributions of this thesis.

Chapter 6 begins by forming an analogy between UAV conflict resolution within a defined altitude and planar collision avoidance in a multi-agent system. The representation of local sensor models, dynamic assumptions and conventions are introduced. Here, several popular geometric methods, known to the literature, are demonstrated and discussed in the form of a comparative study. The algorithms are assessed with respect to several key performance parameters in a series of presented conflict scenarios. Finally, the resilience of each algorithm is compared through the introduction of imperfect sensor assumptions. The results of a comprehensive Monte-Carlo analysis comparing the performance of each algorithm in these conditions is then presented.

The results from this analysis are used to provide context for the proposed *interval avoidance* method. The algorithm is presented and discussed in relation to other geometric methods. The performance of the algorithm is demonstrated and compared to the state of the art methods introduced previously with reference to associated works.

Chapter 7 builds on some of the principles proposed in Chapter 6 in order to define optimal avoidance regions in 3D space. The chapter begins by introducing an adapted sensor model used to observe the relative motion of obstacles in 3D space. Based on the respective sensor uncertainties, it is shown how it is possible to abstract the avoidance problem to consider a set of possible conflict planes. A geometric method is then presented that allows a 3D region to be defined that encloses the corresponding optimal avoidance manoeuvres. Using some of the concepts demonstrated previously, it is then shown how simultaneous avoidance of multiple obstacles can be achieved through a prioritised intersection of the optimal regions.

The performance of the proposed algorithm is then demonstrated in three key scenarios representing typical and problematic aerial encounters. The agents are then tasked with assuring collision avoidance under representative sensor conditions. A comprehensive Monte-Carlo analysis of these scenarios is then used to demonstrate the algorithms ability to maintain safe separation in both singular and multi-agent conflicts.

Chapter 8 then concludes the thesis with a synopsis of the technical findings, methods and results presented previously. Speculation is then given toward future research opportunities given some the key outcomes of the thesis.

1.3 Research Contributions

In this section the significant contributions of the thesis are presented in relation to each of the thesis chapters and the authors associated publications listed in Section 1.4.

Chapter 3 - A symbolic framework for the parameterised modelling and control of UAV's.

- Proposes a symbolic framework for modelling and control of MAV's. Defining different configurations as symbolic profiles is shown to facilitate investigation of generic analytical forms for the purposes of control and stability without numerical parameterisation.

- The concept is initially applied to a quadcopter style UAV. It is then demonstrated how using this approach, symbolic linearised forms can be obtained to define key parameters for LQR and MPC control. A comparison of the applied controllers demonstrates that the model behaviour is as expect and sufficient for preliminary control design.

- The concept of symbolic profiles is demonstrated further in the construction of analytical representations of the Tricopter and Deltacopter style MAV systems.

Chapter 4 - The contributions of this chapter centre on the Polycopter UAS concept, dynamical model and preliminary control design.

- The novel concept of the Polycopter is introduced for the first time using the framework introduced in Chapter 3. By being able to actively orientate each nacelle through two *degrees of freedom* (DOF), the Polycopter is able to maintain a fixed body orientation through complex manoeuvres due to it's enhanced authority over the centre of thrust. As the nacelles represent a significant proportion of airframe mass, their actuation gives the Polycopter a unique ability to modify it's flight and stability characteristics by augmenting it's centre of mass. The Polycopter's properties are presented as a novel platform for preliminary controller design for morphing aerial systems.

- The open-loop dynamic behaviour of the Polycopter is derived analytically under the assumption of "small nacelle deflections". Here the formulation is shown to be analogous to some of the constant inertia, dual tilt-axis systems in the literature with additional authority over the thrust vector.

- Based on the limited ability of first formulation to capture the morphology of the Polycopter, a second model formulation based on the principle of kinematic chains is proposed. A *recursive Newton-Euler* (RNE) method is used to characterise the motion of each component in the airframe, including the variable attitude nacelles. It is shown how using this approach, a platform for the design of intelligent control strategies may be achieved.

Chapter 6 - The concept of interval avoidance (IA) applied to the resolution of planar conflicts in communication denied environments.

- A novel technical review of established geometric collision avoidance approaches is presented, emphasising several key performance parameters in addition to their tolerance to corrupt measurement sources.

- A statistical validation is presented examining the *hybrid-reciprocal velocity obstacle* (HRVO) and *optimal reciprocal collision avoidance* (ORCA) algorithms in different sensor conditions.

- The interval avoidance algorithm is presented for the first time in the context of an SDA scenario occurring between UAVs at a constant altitude. This approach demonstrates how the principles of interval analysis may be used to generate optimal avoidance trajectories in the presence of corrupt sensor measurements without linearisation or approximation.

- Utilising fundamental principles of interval analysis, it is demonstrated how the IA algorithm can be extended to allow the simultaneous avoidance of multiple obstacles. This is shown to be possible through the construction of proximity-prioritised trajectory interval sets.

Chapter 7 - Extension of the interval avoidance method to a generalised case for conflict resolution in 3D aerial encounters.

- The proposed IA concept is reinvented outside of the planar assumptions in Chapter 6 to facilitate conflict resolution in 3D airspace. This is demonstrated to be possible by abstracting the interval problem introduced previously to set of possible conflict planes within a defined interval. Using this technique, the obstacle measurement intervals are used to define a 3D geometric region containing the appropriate optimal avoidance manoeuvres.

- It is shown how under the notation of interval analysis, the new 3D IA concept may again be extended to define a 3D resolution volume describing trajectories that satisfy multiple obstacle constraints simultaneously.
- A statistical analysis of the proposed method demonstrates the algorithms ability to assure safe separation in an array of typical aerial encounters in the presence of corrupt sensor measurements.

1.4 Publications

The authors publications associated with the topics of thesis are outlined below:

Peer reviewed journal publications

1. J.A. Douthwaite, S. Zhao, L.S. Mihaylova, "Velocity Obstacle Approaches for Multi-Agent Collision Avoidance", Unmanned Systems, 2019.

Peer reviewed conference proceedings

1. J.A. Douthwaite, S. Zhao, L.S. Mihaylova, "A Comparative Study of velocity obstacle approaches for multi-agent systems", Proceedings from the *12th UKACC International Conference on Control*, Sheffield, UK, September 2018, pages 1-8.

2. J.A. Douthwaite, A. De Freitas, L.S. Mihaylova, "An Interval Approach to Multiple Unmanned Aerial Vehicle Collision Avoidance", Proceedings of the *11th Symposium Sensor Data Fusion: Trends, Solutions, and Applications*, Bonn, Germany, September 2017, pages 1-8.

3. J.A. Douthwaite, L.S. Mihaylova, S.M. Veres, "Enhancing Autonomy in VTOL aircraft Based on Symbolic Computation Algorithms", In *Towards Autonomous Robotic Systems (TAROS-16)*, Sheffield, UK, August 2016, pages 99-110.

Workshops

1. J.A. Douthwaite, A. De Freitas, L.S. Mihaylova, "An Interval Approach to Multiple UAV Collision Avoidance", Proceedings from the *10th Summer Workshop on Interval Methods and 3rd International Symposium on Set Membership Applications, Reliability and Theory (SWIMSMART 2017)*, Manchester, UK, June 2017, pages 63-64.

Chapter 2

Literature Review - Coordination of Unmanned Systems

This chapter provides a review of the background literature surrounding modelling and control techniques currently being applied in the coordination and control of *unmanned aerial systems* (UAS) and *micro Aerial Vehicles* (MAVs). Networks of coordinated UAS are at the forefront of modern technology with numerous civil, industrial and military entities backing research into their integration in existing infrastructure and legislation [4, 8, 123, 193, 217]. Within Modern UAS, control systems exist at every level; applied in the form of trajectory maintenance, formation control as well as systems for higher level autonomy. It is because of the breadth of literature available, the following section draws up work explicitly in the domains of trajectory control and coordination of unmanned systems. Associated surveys on the topic of unmanned and terrestrial aircraft control can be found in [8, 37, 39, 47, 95, 168, 228, 234]. In Section 2.1 the OpenMAS simulation environment developed as part of these works is introduced. Due to breadth of UAS configurations in the literature, sizes and capabilities, a generalised schema was conceived to allow the simulation of mixed agent groups with varying decision making topologies and physical dynamics. After this, focus shifts to examine the hierarchical control structure adopted to represent the behaviour of decision making agents in Section 2.2 along with some basic kinematic assumptions. The simulation definition of flight plans, waypoints and associated logic is given in Section 2.2.2.

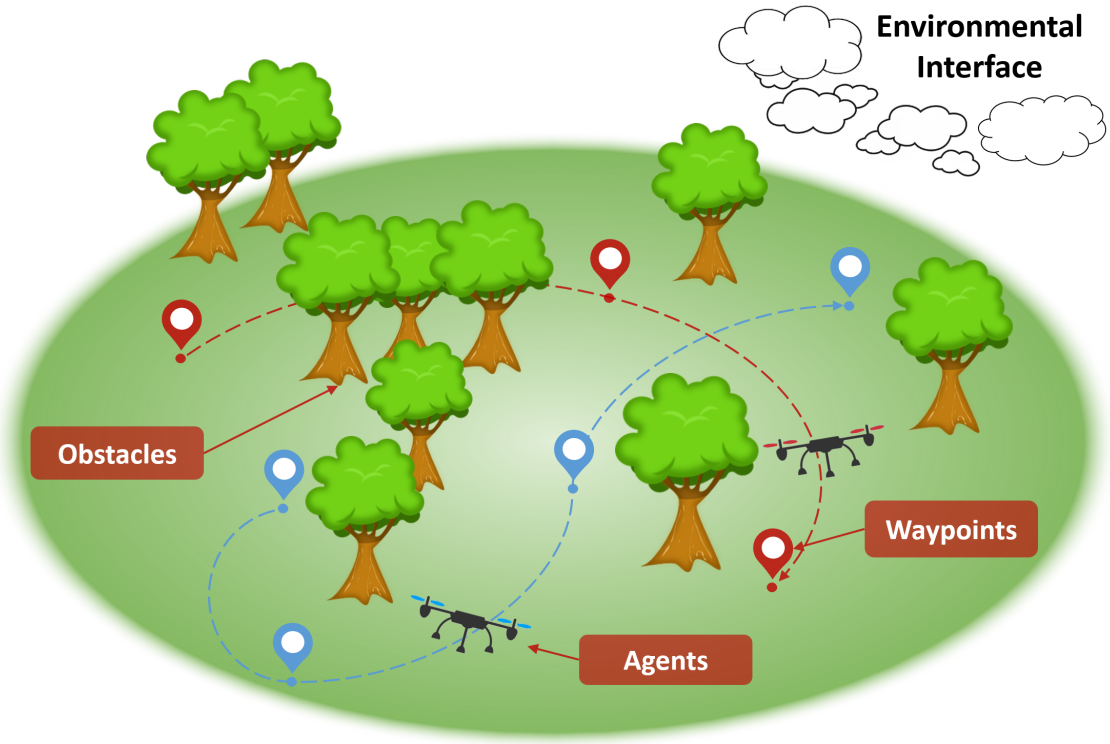


Figure 2.1: *A depiction of a typical MAV application scenario, including the definition of obstacles, agents and waypoints in the context of this thesis.*

2.1 OpenMAS - An Open-source Multi-Agent Simulator

As part of these works the *open-source multi-agent simulator* (OpenMAS) was developed. OpenMAS is an *three-dimensional* (3D) object-orientated Matlab® software package for the simulation of multi-agent systems with sophisticated and diverse characteristics. Entities (i.e. agents, obstacles or waypoints) are represented as an object with unique (or inherited) capabilities, decision logic or dynamics. Representing all entities in the simulation through this layer of abstraction then allows objects to be parameterised for the simulation of small-scale systems or larger, conventional air-traffic operations without any loss of generality.

2.1.1 Object Representation

In this thesis, the term "scenario" is used to characterise a unique set of environmental conditions and object configurations that define a given simulation instance. The object configuration is defined as an index containing specific agents A_i, obstacles O_i and waypoints W_i that represent a given scenario (see Figure 2.1). All objects are represented in shared 3D Cartesian space by their position $\boldsymbol{\chi}_i^{xyz} = [\chi_i^x, \chi_i^y, \chi_i^z]^T$, velocity $\dot{\boldsymbol{\chi}}_i^{xyz} = [\dot{\chi}_i^x, \dot{\chi}_i^y, \dot{\chi}_i^z]^T$ and it's quaternion pose $\boldsymbol{\chi}_i^q = \boldsymbol{q}_i = [q_0, q_1, q_2, q_3]^T$ in the inertial frame. The objects

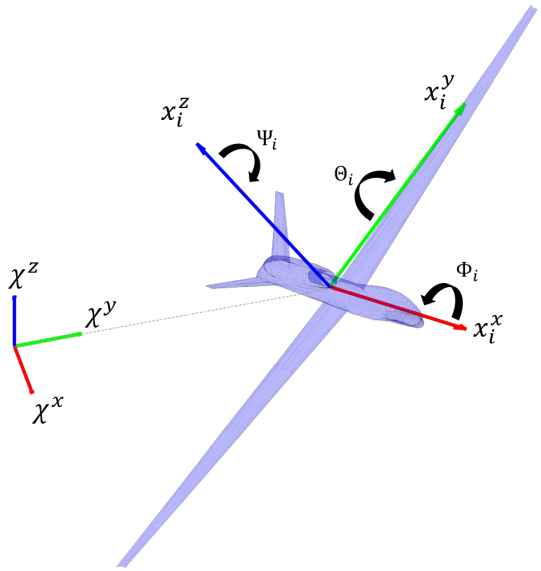


Figure 2.2: *Transformation between agent i's body axes and the inertial reference frame [66].*

state at any given step k (or time t_k) within the inertial *east-north up* (ENU) coordinate system is then defined as $\boldsymbol{\chi}_{k,i} = [\boldsymbol{\chi}_{k,i}^{xyz}, \dot{\boldsymbol{\chi}}_{k,i}^{xyz}, \boldsymbol{\chi}_{k,i}^{q}]^T$. Given a quaternion pose $\boldsymbol{\chi}_{k,i}^{q}$ the rotation matrix $\mathbf{R}_{k,i} \in SO(3)$ transforming motion within the frame of i into the inertial frame is defined by Equation (2.1) [75]. The relation between the two axes can be seen in Figure 2.2.

$$\mathbf{R}_{k,i} = \mathbf{R}(\boldsymbol{q}_{k,i}) = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_0 q_2 + q_1 q_3) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_0 q_1 + q_2 q_3) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{2.1}$$

Here $\mathbf{R}_{k,i}$ has the properties such that $\mathbf{R}_{k,i}(\mathbf{R}_{k,i})^T = \mathbf{I}_{3\times 3}$.

The motion of the object i expressed in inertial coordinates can therefore be expressed as $\dot{\boldsymbol{\chi}}_{i,k}^{xyz} = \mathbf{R}_{k,i}\boldsymbol{v}_{i,k}$, where $\boldsymbol{v}_{i,k} = [u, v, w]^T$ are the linear velocities of object i in it's own frame of reference. The object's quaternion representation at t_{k+1} can similarly be updated from the object's rates about it's body axes $\omega_{i,k}$, by representing them as rotations about the inertial axes $\mathbf{R}_{k,i} \cdot \boldsymbol{\omega}_{i,k} = [\omega^x, \omega^y, \omega^z]^T$. The attitude update procedure for object i is then the result of the integration of it's quaternion pose (see Equation (2.2)).

$$q_{k+1,i} = q_{k,i} + \frac{\Delta t}{2} \begin{bmatrix} \sigma_k & -\omega^x & -\omega^y & -\omega^z \\ \omega^x & \sigma_k & \omega^z & -\omega^y \\ \omega^y & -\omega^z & \sigma_k & \omega^x \\ \omega^z & \omega^y & -\omega^x & \sigma_k \end{bmatrix} \cdot q_{k,i} \tag{2.2}$$

where σ_k defines the normalising diagonal component at step k;

$$\sigma_k = 1 - (q_0^2 + q_1^2 + q_2^2 + q_3^2) \tag{2.3}$$

The equivalent rotations of the objects *field of view* (FOV) in the inertial ENU coordinate frame may also be calculated from their relation to the quaternion pose $\boldsymbol{\chi}_{k,i}^{q}$ (see Equations (2.4-2.6):

$$\Phi_i = \tan^{-1}\left(\frac{2(q_0 q_1 + q_2 q_3)}{1 - 2(q_1^2 + q_2^2)}\right) \tag{2.4}$$

$$\Theta_i = \sin^{-1}\left(2(q_0 q_2 - q_3 q_1)\right) \tag{2.5}$$

$$\Psi_i = \tan^{-1}\left(\frac{2(q_0 q_3 + q_1 q_2)}{1 - 2(q_2^2 + q_3^2)}\right) \tag{2.6}$$

The inertial state trajectory $\boldsymbol{\chi}_i$ is then recorded over the prescribed time interval $t_k \in [t_0, t_{max}]$ as a result of each objects defined update procedure. Abstraction of $\boldsymbol{\chi}_i$ from the i's dynamic states $\boldsymbol{x}_i$ allows the object's dynamical representation to be specified externally; in accordance to specific object(agent) configurations (i.e. unique constraints, capabilities, axis conventions etc.) [63]. The representation of an i's orientation as a unit quaternion ensures that it's trajectory can be accurately represented through complex manoeuvres, whilst preventing anomalies such as *gimbal-lock* from occurring (see associated literature [50, 74, 75]). The high level expression for the evolution of $\boldsymbol{\chi}_i$ can then be expressed in Equation (2.7) in terms of the linear and angular rates about it's local body axes.

$$\boldsymbol{\chi}_{k+1,i} = f_i(\boldsymbol{\chi}_{k,i}, \boldsymbol{v}_{k+1,i}, \boldsymbol{\omega}_{k+1,i}, \Delta t) \tag{2.7}$$

Here, Δt defines the simulation sample period. An overview of the OpenMAS simulation procedure can be seen summarised in Figure 2.3. For more information and the original source code please see [66].

2.1.2 Collision Definition & Detection

Objects operate within a shared 3D space in which the chance of collision between two objects exists. In general, collisions between two objects are characterised by the intersection of their designated collider geometries. These geometries allow approximations to be made about the objects physical geometry, which may be complex or concave, for purposes of efficient collision evaluation. As objects i and j enter a collision scenario, their

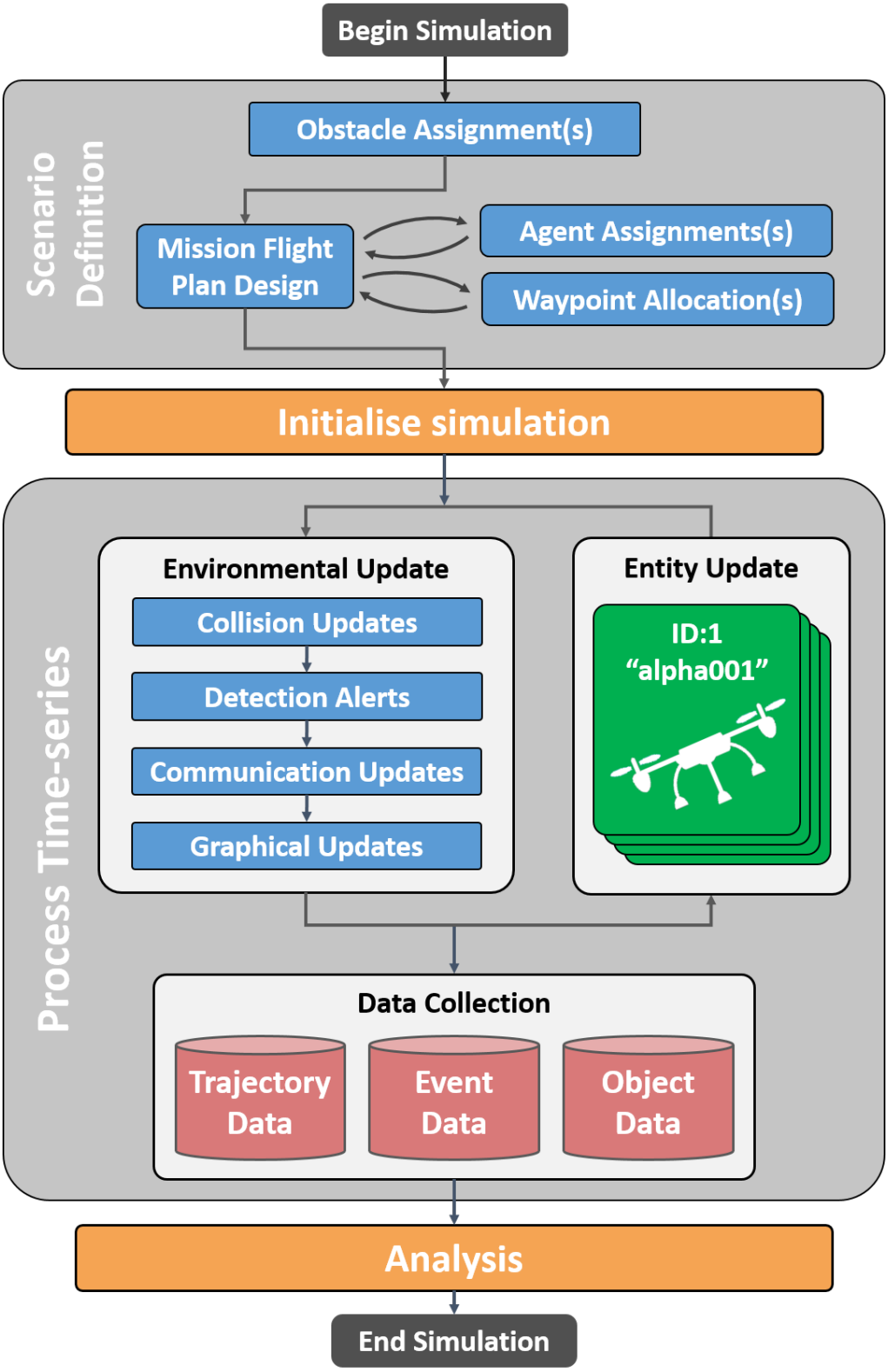


Figure 2.3: *A system representation of the open-source multi-agent simulator (OpenMAS) environment developed as part of these works [66].*

collider geometries are then used to evaluate whether a collision has occurred at time t_k.

In these works *agents*, such as UAS and other *unmanned aerial vehicles* (UAVs), are assigned spherical collider geometries depicting the minimum allowable Euclidean separation in the inertial frame. This condition is defined as $||\boldsymbol{\chi}_i^{xyz} - \boldsymbol{\chi}_j^{xyz}|| < (r_i + r_j) - \iota$, where $||x||$ defines the Euclidean norm of x and ι is a condition tolerance to eliminate ambiguity between collisions and narrow-misses caused by the nature of discrete simulation. Spherical collision geometries are also used in the generation of waypoint achieved events with a defined radius representing their tolerance for the position of i. The use of radial collision constraints is well established within the collision avoidance community [27, 63, 65, 250].

Collision events that occur between with other objects, such as obstacles, are evaluated through sequential *axis-aligned bounding box* (AABB) and *object aligned bounding box* (OBB) checks once the initial radial separation condition is violated. The dimensions of the AABB and OBB geometries characterising object i are parameterised by the extents of the physical geometry of i; specified in body axis coordinates, given relative to it's origin [270].

2.1.3 Sensor Representation

Agents are defined as objects with the unique capability of making observations of their surroundings (O_j, A_j and W_i). Agents are assumed able to observe the trajectory of a second object j using an on-board system. This system is assumed to make measurements within a dedicated coordinate frame, positioned on the agent's geometry. For simplicity however, the coordinate system of the "sensor" is assumed to be aligned with the body axis of the agent and positioned at it's origin. The apparent trajectory of object j in the frame of i can therefore be represented conveniently in terms of their representation $\boldsymbol{\chi}_i$ and $\boldsymbol{\chi}_j$ in the inertial frame via Equations (2.8) and (2.9).

$$\boldsymbol{p}_j = \mathbf{R}_i^T(\boldsymbol{\chi}_j^{xyz} - \boldsymbol{\chi}_i^{xyz}) + \boldsymbol{w}_p \tag{2.8}$$

$$\boldsymbol{v}_j = \mathbf{R}_i^T(\dot{\boldsymbol{\chi}}_j^{xyz} - \dot{\boldsymbol{\chi}}_i^{xyz}) + \boldsymbol{w}_v \tag{2.9}$$

Here the terms $\boldsymbol{w}_p = \mathcal{N}(\mathbf{0}_{3\times1}, \mathbf{I}_{3\times3} \cdot \sigma_p^2)$ and $\boldsymbol{w}_v = \mathcal{N}(\mathbf{0}_{3\times1}, \mathbf{I}_{3\times3} \cdot \sigma_v^2)$ introduce the sensors representative noise, assumed Gaussian with a variance of σ_p and mean of zero. Sensors such as cameras, LIDARS and RADARS, implement a spherical based coordinate system in sampling the position of object j [55, 212]. The equivalent spherical-coordinate representation can be calculated directly from it's relative position to the sensor $\boldsymbol{\lambda}_j$ (seen

in Equation (2.10)).

$$d_j = ||\boldsymbol{\lambda}_j|| = ||\boldsymbol{p}_j - \boldsymbol{p}_{i,sensor}|| \tag{2.10}$$

$$\theta_j = \sin^{-1}\left(\frac{\lambda_j^z}{d_j}\right) \tag{2.11}$$

$$\psi_j = \tan^{-1}\left(\frac{\lambda_j^y}{\lambda_j^x}\right) \tag{2.12}$$

Here use of $||\boldsymbol{\lambda}_j||$ constitutes the Euclidean norm[1] of the vector $\boldsymbol{\lambda}_j$. The measurement variables d_j, θ_j and ψ_j represent the apparent range, elevation and heading of object j respectively, in the coordinates of i's sensor positioned at $\boldsymbol{p}_{i,sensor}$ in the body axes. Objects that are more geometrically complex, such as polygons, are expressed in the frame of i by a similar process (for more information see [66]).

It is assumed that the agent's interpretation of the world is confined to a specified range, representing the effective range of their sensory system. Generally, agent i is considered able to observe object j if their Euclidean separation satisfies $||\boldsymbol{\chi}_{k,j}^{xyz} - \boldsymbol{\chi}_{k,i}^{xyz}|| \leq r_i^{detection}$ where $r_i^{detection}$ is used to parameterise the maximum effective range of i's sensory system.

2.2 Unmanned Aerial Systems

Modern unmanned systems vary greatly in their level of sophistication. In the context of these works, UAS are considered to be intelligent agents A_i with defined sensory capabilities, hierarchical decision procedures and physical dynamics. The decision procedure, while specific to a given system, is generally considered to utilise information gained at t_k on the systems current state $\boldsymbol{x}_{k,i}$ and the state of it's surroundings.

2.2.1 Flight Management & Control

The *flight management system* (FMS), is the general term given to the system responsible for managing mission objectives, auxiliary functions and aircraft trajectory. Full-scale systems, such as the Global Hawk RQ-4 [178] are highly complex, with numerous subsystems responsible for dedicated hardware. These correspond to engine management facilities, health monitoring as well as communication and broadcast systems specifically for integration into existing airspace infrastructure. Conversely, the FMS on-board the average commercially available MAV is now integrated into a single *printed circuit board* (PCB). Developments in this area stem from the growing popularity of *single board computers*

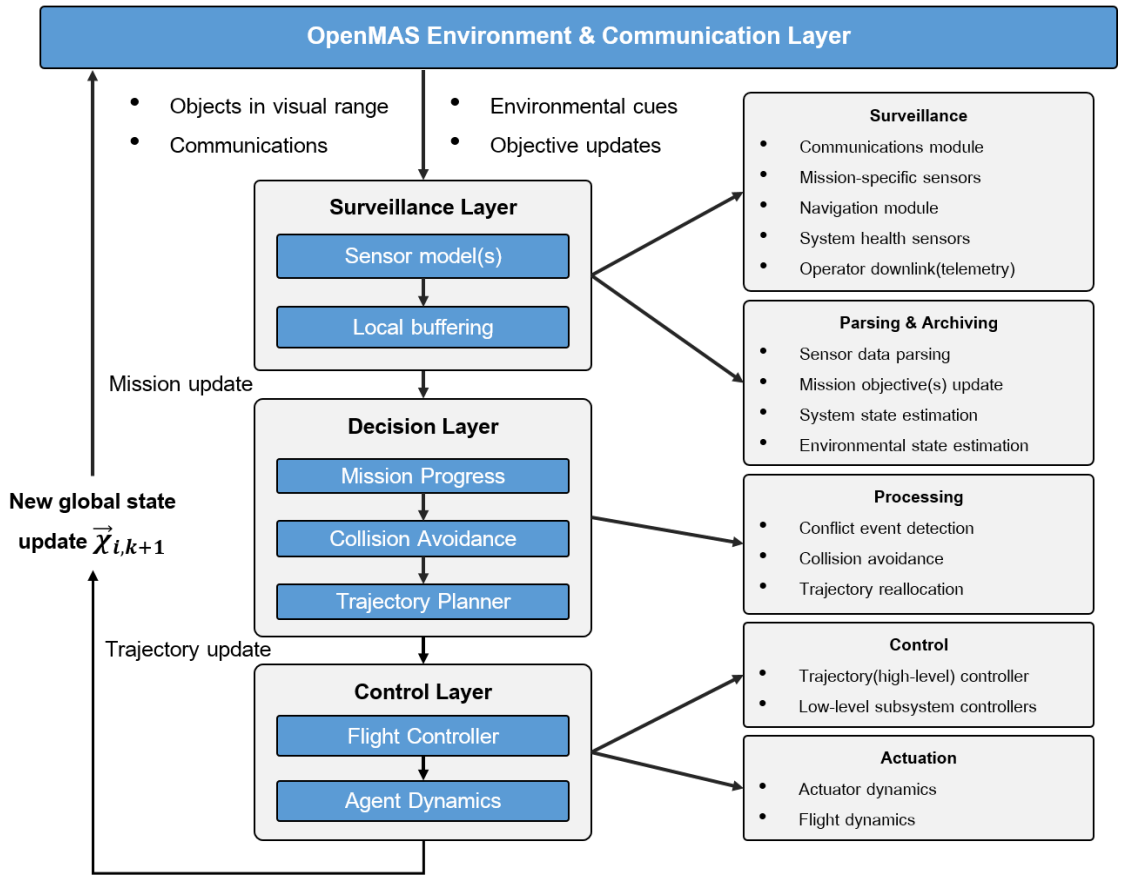


Figure 2.4: *An outline of the digital structure of modern UAS, with dedicated sensing, decision processes and plant dynamics .*

and accessible development platforms such as the Raspberry Pi® and Arduino® [71,167].

The sophistication of a UAVs FMS ultimately dictates the level of autonomy achievable on-board the system; facilitating complex tasks such as adaptive path planning, failure tolerance and collision avoidance all of which are considered steps towards more adaptive and autonomous aircraft. More specifically, the FMS is responsible for flow of information between the aircraft's various subsystems (see Figure 2.4). As a generalisation, the associated hardware and software subsystems constituting the FMS can be grouped into hierarchical layers:

- **Sensory layer** - Responsible for the sampling of the UAV's sensory systems. Typically, UAS are equipped with an *inertial measurement unit* (IMU) containing a three-axis accelerometer, gyroscope and magnetometer. Together these sensors allow a system to estimate the current pose of the aircraft in 3D space. Modern UAS now also implement GPS receivers, barometers and airspeed indicators (e.g. optical flow or pitot tubes), to allow the estimation of the aircraft's position, altitude and speed respectively. Other sensors may provide facilities for command

interpretation and communications, measure system health (e.g. battery voltage, CPU temperature, engine speeds..) as well as the state of the system's actuators. More sophisticated systems with higher sensing capabilities are able to make observations on the state of the environment [42]. Common examples of this include monoscopic(stereoscopic) cameras, RADAR and LIDAR, which either actively or passively sample changes in the environment (see Section 6.1) [212].

Information received from the system's IMU, health and auxiliary sensors are filtered and merged through a defined sensor fusion algorithm (typically a *extended Kalman filter* (EKF)) to formulate a new estimate of system and the state of entities in the environment. In certain cases, agents with sophisticated target tracking algorithms may also utilise object recognition and classification algorithms to attach additional data to known data features [55, 234].

- **Decision layer** - The design of the system's decision layer varies significantly between applications and operational scale. Often cited as the system's *auto-pilot*, the decision layer is responsible for the resolution of the aircraft's trajectory from an enhanced understanding of the environment [9,234]. More specifically, sophisticated "decision" layers may contain procedures for re-evaluating mission progress (in the form of waypoint and objective updates), recognising conflicts with obstacles and compute an appropriate escape trajectory [204]. In addition to this, the decision layer is responsible for interacting with mission orientated subsystems and dedicated hardware.

- **Control/Actuation layer** - Once a viable trajectory is generated it is passed to an actuation layer to be enacted. The *flight control system* (FCS) is often cited as the system responsible for generating inertial accelerations to achieve a desired flight-path [42, 95, 178]. In reality this the FCS may also be a distinct system (or series of subsystems) representing motor drivers, electronic speed controllers, servos and other actuator state regulators. Nowadays, these subsystems can be integrated directly into a single "flight controller" board in some commercial UAVs at little added cost.

  Control signals received from the FCS by the systems actuators, generate accelerations about the aircraft's body axes in order to adjust the aircraft's course. The accelerations are then the result of the dynamical interactions between the aircraft's physical dynamics, aerodynamics and dynamics of the state of it's various actuators (e.g ailerons, rudder, rotor speeds) [225].

In these works, the structure seen in Figure 2.4 represents the agent's procedure as they progress through the environment and the state of obstacles, agents and waypoints are observed. Recent articles on the integration requirements for modern UAS can be found in [5, 37, 93, 126].

2.2.2 Waypoints & Mission Planning

Objectives representing the task assigned to the UAV A_i are given in the form of an intended "flight-path". A flight-path is defined by a vector of waypoints $\mathrm{W}_i \in [W_1, W_2, ..., W_n]$ that can be observed by UAV A_i at time t_k. In each case, each waypoint in W_i defines the desired state $\boldsymbol{\chi}_{i,\mathrm{wp}}$ of A_i at time $t_{k,\mathrm{wp}}$ within the inertial frame, namely; it's desired position, velocity or attitude [107, 225, 228].

Upon initialisation of a given scenario, UAVs are designated a flight path representing the task to be accomplished (e.g. transportation, patrolling etc.). waypoints W_i assigned to agent i are only observable by agent i, and given a defined priority dictating the order in which they are to be completed. UAVs moving through the environment may encounter waypoints in the field and may choose to redefine the flight-path accordingly. Representation of a UAV's flight path as a series of waypoints is also representative of flight-aids typically associated with the navigation of conventional aircraft with operational airspace. UAVs operating in a managed airspace utilise systems such as GPS, *non-directional beacons* (NDB) and other supplementary flight aids as part of their *area navigation* (RNAV) equipment. Such systems are necessary to allow navigation in a highly sophisticated network of segregated traffic channels and geographic constraints.

2.2.3 Dynamic Representation

At any one time t_k the state of agent A_i in the inertial frame $\boldsymbol{\chi}_{k,i}$ can be expressed as a function of it's dynamical state representation $\boldsymbol{\chi}_{k+1,i} = g_i(\boldsymbol{\chi}_{k,i}, \boldsymbol{x}_{k,i})$. Here g_i represents the transformations introduced in Equations (2.1) and (2.2). The evolution of the agent i's dynamical states $\boldsymbol{x}_{k,i}$ is dictated by the systems dynamic contributions, namely; it's aerodynamics, inertial, actuator physics and propulsion dynamics. The nature of these interactions are distinctly configuration-dependant and can expressed generally by the dynamical function $\boldsymbol{x}_{k+1,i} = d_i(\boldsymbol{x}_{k,i}, \boldsymbol{u}_{k,i}) + \boldsymbol{w}_x$. Here d_i represents a set of differential equations in $\boldsymbol{x}$, describing the dynamical configuration of A_i. The agent's input vector $\boldsymbol{u}_{k,i}$ defines the vector of control parameters received from the FCS at time t_k. The parameter $\boldsymbol{w}_x = \mathcal{N}(\mathbf{0}_{3\times1}, \mathbf{I}_{3\times3}\boldsymbol{\sigma}_x^2)$ defines a state noise term with zero mean, with variance σ_x.

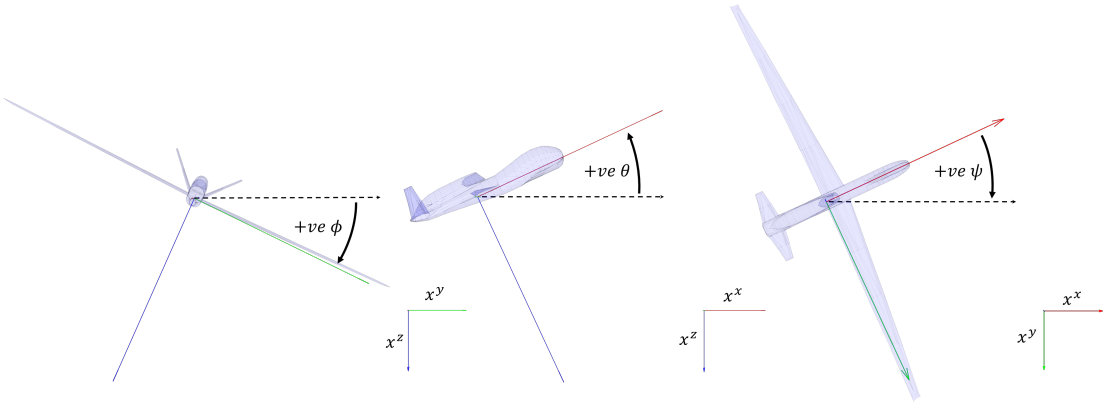


Figure 2.5: *A depiction of the FCS axes of control and definition of it's Euler heading angles* $\boldsymbol{\eta} = [\phi_i, \theta_i, \psi_i]$ *as deflections in roll, pitch and yaw respectively.*

The motion of a UAS is assumed to be handled by an on-board FCS. The structure of the FCS is typically hierarchical in modern unmanned systems, with controllers responsible for altitude, attitude rates and speed control. In these works, a dedicated control convention is assigned within the FCS to allow aircraft dynamics to be specified in the conventional aerospace *North-East-Down* (NED) coordinate system. The description of the FCS rotation convention is given in Figure 2.5.

Under the NED convention, the orientation of the airframe is represented by sequential deflections in roll, pitch and yaw as ϕ_i, θ_i, ψ_i respectively. Together, these angles represent the aircraft's Euler rotations ϕ_i, θ_i and ψ_i about it's control axes x^x, x^y and x^z respectively [51, 85]. The coordinates of the CG $\boldsymbol{x}_{i,cg}$ of the UAS, propulsive systems and moments of inertia are assumed measured relative to the body axis origin. Typically, the CG is assumed aligned with the body axis origin such that $\boldsymbol{x}_{i,cg} = [0, 0, 0]^T$. In this section it has been described how the dynamical representation a given UAS may be defined by assigning it's corresponding descriptor function $\boldsymbol{x}_{k+1,i} = d_i(\dot{\boldsymbol{x}}_{k,i}, \boldsymbol{x}_{k,i}, \boldsymbol{u}_{k,i}, \boldsymbol{w}_{k,i})$. This approach to abstraction allows us to consider arbitrary UAS configurations with a multitude of constraints, controllers and assumptions. In Chapter 3 an accompanying approach to assembling UAS dynamical descriptors is introduced; using a symbolic approach to model synthesis.

2.3 A Review of the State of the Art

2.3.1 Classical Control

Proportional, integral, differential (PID) control are long established control methods within the field of automatic control [57]. The popularity of PID control stems from it's

relative simplicity and effectiveness in a wide range of systems. This makes them an ideal benchmark with which to compare other control mechanisms. The study of unmanned aerial systems is no different, in that PID controllers are regularly cited as the go-to mechanism for the control of various flight systems and off-the-shelf controllers [42, 122].

PID controllers are typically confined to *single input single output* (SISO) systems, or made relevant via a decoupling process [234]. Examples of PID being applied directly to one-dimensional control include altitude control [34, 62] and heading control in [34, 62, 72, 220, 256]. More complex examples of PID control can be found in the context of stabilisation and trajectory control. Here, feedback is typically provided by dedicated PID controllers assigned to each of the UAVs *degree of freedom* (DOF) or motion about each body axis [32, 34, 58, 59, 72, 127, 220, 234]. Feedback linearisation techniques are applied in [58] in order to reduce more sophisticated MAV configurations so that PID control of select aircraft axes can be achieved. In such use cases, the method is shown to be able to stabilise the system within a reasonable settling time and steady-state error as part of a preliminary control investigation.

Due to the popularity of PID control, there are a number of existing PID hybrids citing the use of *gain scheduling* in [127, 132, 194, 234] and fuzzy-logic in [21, 132] as a mechanism of scheduled responses to changes in the aircraft's flight window. Additionally, efforts have been made to improve the robustness and performance of the traditional PID controller without the added complexity of optimisation. The Kestral autopilot [179] is an example of cascaded PID with both feedback and feed-forward techniques applied as a commercially available product. The *robust deadbeat* controller, as an adaptation of traditional PID control, is proposed as a method of controlling higher order plants. Kada et al in [127] demonstrates the general PID control formulation. The technique of *Model Order Reduction* is applied to a complex, fixed wing UAV system in order to derive the variable feedback functions [223].

2.3.2 Linear Quadratic Gaussian/Regulative Control

The field of optimal control is well established in the literature and also within the control of terrestrial flight systems. *Linear quadratic Gaussian* (LQG) control and *Linear quadratic regulation* (LQR) are concepts derived originally from optimal control theory and applied to linear systems. A state-feedback matrix is calculated from the solution to the *algebriac Riccatti equation*, composed of the state and input penalisations and a representation of the linear systems. The solution is typically found numerically via a quadratic cost function [32]. For further information on LQG/LQR control, see [6, 13, 64].

This approach is shown to be effective for attitude stabilisation for MAVs in [15, 32, 64, 118]. Recent examples of the LQR methods applied in highly dynamic manoeuvres can be seen in [83]. Typically, applications of linear control are limited by the assumption of a narrow operating window. However, the author proposes a continued linearisation process to allow the system to dynamically adjust the LQR state feedback in accordance to it's current state. In addition to this, the aircraft's attitude is represented as a unit quaternion to avoid singularities known to occur through more aggressive manoeuvres.

2.3.3 Back-stepping Control

Back-stepping, originally proposed by Koktovic in his paper [135], is a form of linear control that has also seen increasing interest in the field of UAV coordination. Under this regime, a control input is devised that will act to stabilise the system in accordance to it's *Lyapunov stability* conditions. The term *back-stepping* then refers to a retrospective control signal, that is applied to the input in the form of differential feedback [108]. In the literature, there have been numerous applications of back-stepping control in the context of UAV coordination. In the works of Bouabdallah and Siegwart [33], [34], the application of non-linear integral back-stepping can be seen applied to both the attitude and altitude systems of a micro quadcopter. The experiment is evaluated in both simulation and practically on board their prototype the OS4 quadcopter. The stability of the derived back-stepping approach is compared to that of a *Sliding Control* mechanism in [33], but is also evaluated as a mechanism for attitude, position and altitude control of the OS4 in [34].

A comparison is formed between the back-stepping control regime and a derived *model-predictive control* (MPC) method in [152]. Here the performance of the back-stepping controller is shown to be comparable to modern predictive methods, by achieving the same rate of convergence and overshoot, at a reduced computational cost. It is also shown to be much more demanding on the system actuators; unable to consider constraints and enact large and rapid corrections in some conditions. The concept of *adaptive* back-stepping control is outlined in [79, 149], as a mechanism of altering the internal process to modify the aircraft's flight envelope. This method achieves a robust control regime to variety non-modelled, non-linear, effects by enforcing an operative envelope via defined constraints. The authors of [79] present a control *command filter* in order to provide intermediate inputs to the back stepping controller. The effect of the adaptive back-stepping regime is shown to demonstrate a tighter tracking of the reference trajectory and increased resilience to disturbances.

Back-stepping is applied directly to trajectory tracking of a team of multiple agents in [198]. The controller is framed as a 2D trajectory tracking problem under a back-stepping control regime. The same concept is adapted to the 3D case in [253], except the back-stepping control laws are supplemented by a potential-based collision avoidance algorithm. The proposed system is shown to be effective at maintaining the desired formation in the presented conditions.

2.3.4 Model-Predictive Control

MPC is a well established form of optimal control typically associated with industrial applications and more recently in the control of UAVs [8, 122]. A model describing the systems behaviour is used to predict performance over a defined horizon. Based on known system parameters and operational constraints, a cost function is optimised to determine the optimal control inputs to be implemented [155]. In the literature there are examples of MPC being applied at all levels of a UAVs control hierarchy, with significant advances being made in centralised, decentralised and distributed approaches in recent years [155, 169, 262].

Centralised approaches to MPC UAV coordination typically consider a system describing the complete UAV squadron. In [68,202], a centralised controller is used to maintain the relative separation of agents in the squadron and minimise distance from target region. In [201], the author compares previous centralised MPC control to a reduced decentralised formulation. The author demonstrates comparatively similar behaviour, with computational performance and scalability whilst guaranteeing constraint satisfaction (no collisions). A similar sentiment is made in [221] in the development of several extended-Kalman filter based control architectures. Although the centralised formulation yielded a lower overall cost, the computation time was significantly higher than approaches adopting other topologies.

The decentralised and distributed approach to MPC UAV coordination is well documented in the literature [43, 47, 69, 98, 201, 272]. Moving away from the assumption of a centralised topology brings new challenges; the need for reconfigurable communication topologies is stressed in [98]. In [69], the author demonstrates the relationship between the input update time and the stability of the formation controller. Nonlinear and heterogeneous subsystem dynamics are considered under the notion of compatibility constraints which are shown able to guarantee formation convergence in their presented conditions. In [43], a neighbourhood is considered with assigned priorities to its members. Here, the problem is formulated locally and considers non-linear agent motion in relation to a virtual

formation reference point. Using their proposed cost function, collision avoidance between members is shown to be guaranteed in ideal conditions. In [272], a similar assertion is made in the systems convergence towards a "regular lattice" formation. This is achieved using a decentralised cost function penalising the distance from known neighbours. While shown to converge on the desired formation with collision avoidance explicitly guaranteed in their lemma, computation time and therefore feasibility is not considered. A dual mode MPC method is applied to the formation control of a team of robots in [257]. To guarantee formation stability, the controller must alternate between their MPC controller and a second input-output feedback linearisation controller targeting a terminal system state.

Within the literature, MPC is also widely applied in the context of UAV trajectory tracking. These methods are typically associated with systems where linear and traditional methods are insufficient to control a highly unstable or non-linear plant. Joelinato et al in [121] demonstrates how multiple linear approximations can be used to transition between flight conditions on-board an autonomous helicopter. The abstraction of linear behaviour is shown only to work under strict stability constraints, but as a result is able to produce feasible computation times. In [151] non-linear MPC is employed towards a similar goal using predictions of high-order state variables in the design of a closed-form MPC algorithm. More recently, MPC has been applied to the control of quadcopter style aircraft also. In [152] and [64] linear MPC is demonstrated to sufficiently stabilise a quadcopter around a hover condition in order to track a trajectory under a single layer control regime. The performance of the presented approach is then compared to traditional PID and LQR control whereby it is shown to yield less aggressive and smoother control responses.

Recent development towards "fast" MPC have also made the use of MPC in systems with fast dynamics more feasible. In [139] the flat outputs of the non-linear helicopter system are identified and a *linear time variant* (LTV) approximation is introduced for state prediction. In more recent articles [129,170] non-linear MPC attitude control is cited in conjunction with higher level optimal control strategies for sophisticated trajectory tracking. Their approach utilises geometric trajectory integration in combination with an optimal position controller to design inputs to a low level MPC controller tracking a target thrust vector. In this article tolerance to component failure is demonstrated; a point also addressed in the formation of an adaptive UAV MPC controller in [39].

The theme of exploiting highly dynamic UAVs through non-linear MPC techniques is also demonstrated in [159] in the design of a non-linear MPC controller for guidance

in deep stall conditions. Learning-based MPC is a field that has seen increasing interest in the robotics community. Online learning of system parameters combine sophisticated statistical learning techniques with classical control theory. In [17], a learning MPC controller is extended from a robust-tube MPC controller to incorporate statistical updates describing the plants un-modelled dynamics. Later in [18,35] this method is shown to be able to improve performance by updating the model live and demonstrating convergence toward the true dynamical representation if sufficiently excited.

2.3.5 Fuzzy Logic and Gain Scheduling

The *fuzzy-logic* and *gain-scheduled* control approaches are historically popular adaptive techniques that have since been applied to various aspects of UAS coordination. Fuzzy logic based control is frequently seen in conjunction with gain scheduling and other classical control approaches in the literature. Both techniques attempt to address highly non-linear problems through the creation of discrete operating states, with both methods emphasising the creation of smooth transitions between them as control solution. Interest in fuzzy-logic within classical control applications stems from it's use of established rule sets, as apposed to conventional quantified approaches to signal control. These rules are defined as *fuzzy* conditional statements which, through the use of *linguistic variables*, allow constraints to be literal and descriptive [21, 190, 214, 267, 268].

Instances of adaptive flight controllers using gain scheduling and fuzzy logic can be found regularly in the literature. In [247] gain-scheduling is applied in the context of robust altitude control of a UAV. The presented controller is able to generate adaptive altitude feedback in response to changing airspeed. This is achieved by transitioning between discrete operating conditions and associated aerodynamic profiles. Fuzzy gain-scheduling is used in the augmentation of a conventional "sliding-mode" control-based attitude controller in [265]. Here a fuzzy logic system is introduced to manage the *sliding-mode control* (SMC) gains in order to reduce chatter between operating states and incorporate parametric uncertainties. The resulting system is shown to have increased tolerance to external disturbances in their presented conditions.

These approaches to adaptive flight control can also be seen in the context of fault tolerance in [12, 70, 210]. In the first paper, a fuzzy-based rule set is used as a high-level tuning regime for a PID-based attitude controller. The system is shown able to dynamically adjust the PID gains in response to both symmetric and asymmetric reductions in lift generation. A similar approach is presented in the control of a quadcopter in [12], comparing the performance of a gain-scheduled PID controller against a "model reference

adaptive control" strategy. Both methods are shown to respond to loss of thrust from a control surface and return to the desired trajectory. However the MRAC method is shown to be more effective with respect to ease of use and mean trajectory error.

2.3.6 Biologically-Inspired Methods

The concept of mimicking nature for the purposes of control and navigation has been around for decades within the robotics community. In [156, 263] genetic algorithms are applied in the context of UAV navigation. In [156] this is motivated by maximising the information gained on a known target. The coefficients of the value function are revised, subject to a concurrent genetic algorithm. This is achieved by valuing the equivalent cost of alternative trajectories based on the information gained in their surveillance task. Coordination of UAVs through cluttered environments is addressed in 2D [263] and later in 3D considering fluctuations in terrain in [173].

A more literal example of biological emulation can be found in [3]. Here a UAV's trajectory is adjusted based on an automatic terrain following system mimicking the retina of an eye. Assuming contrasting features could be identified, altitude feedback could be achieved with only a small number of pixels. A similar study extends this concept for the speed control of a UAV using a low complexity optical sensor [205, 206]. The UAV is shown to respond to a change in relative elevation and maintain a set speed. A more recent example of generating guidance signals from visual stimuli can be found in [237]. An optical flow sensor is used to track the relative movement of object features. A virtual force is computed from the relative motion of the environment based on the displacement and size of features reactively.

The advancement of small, yet powerful computing systems has lead to an increase in interest of neural net based navigation of UAVs [42,90,145]. In [145] a neural network is used to compute an input response to attitude error. A three layer neural network is used to approximate a non-linear attitude controller for a fixed wing aircraft. A higher-level adaptive layer is used to adjust the neural weightings within a bounded range and producing significantly tighter error tracking. This principle is extended in the development of an adaptive flight controller for a quadcopter UAV in [146]. The author presents a feedback function that modifies the coefficients of a neural network based on the systems Lyapunov stability. The results are compared to a traditional state-feedback controller, in which their controller demonstrates increased resilience to disturbance in a trajectory tracking exercise.

2.3.7 Other Methods

Other techniques that do not explicitly belong to the aforementioned categories are also known within the context of these works. Consensus algorithms as a mechanism for UAV coordination are also known to the literature. In [24], it is shown how a centralised control regime may be extended to decentralised regime through a proposed optimisation process. The cooperation of agents within a system is rewarded with respect to the presented decision variables, coordination is also parameterised by a cost penalising trajectory incoherence. In [232], the concept of connectivity robustness is used to create connectivity constraints in mobile robotic networks. This is achieved by optimising a cost function parameterising the systems connectivity and maximal flow of information between nodes. The method is however not extended to sets where the movement of the nodes is inhibited by obstacles. Using methods from information theory, a dual mode control approach is developed in [10] using a learning *Markov decision process* (MDP). A receding horizon control approach is used to control the plant dynamics within discrete MDP states, with a higher level entropy-based learning objective acting to reduce uncertainty of the system.

2.4 Summary

A review of the associated methods for the modelling and control of UAS has shown the field to be highly active. Development in this area has seen the emergence of a breadth of UAV design topologies with unique opportunities where conventional configurations may be challenged. The information presented in Section 2.3 can be seen collated in Table 2.2 as a summary of the contributions of the literature. The terminology used to describe the different aspects of modern FCSs are also given in Table 2.1.

It is clear by observing Table 2.2 that while there have been numerous examples of different control methodologies applied in the context of UAS coordination, they have also been applied in response to differing control objectives. Each of these objectives in turn contributing to the overall autonomy of the coordinated UAS by providing utility at different levels of autonomy. Many of the challenges associated with classical control techniques stem from the non-linearities and cross-coupling present in the dynamics of most UAS. This is exacerbated further when considering highly agile systems or systems with morphing capabilities. As a result, multiple cascaded control mechanisms are often seen necessary in order to achieve even low-level functionality. One of the most promising and versatile forms of control utilise an understanding of the systems dynamics in the online calculation of their feedback signal. Examples of this type of control are found

Control Objective	Symbol	Control Locality	Symbol
Altitude	Al	Local Only	L
Attitude	At	Centralised	C
Full-trajectory	T	Decentralised	De
Multi-Vehicle/Formation	F	Distributed	Di
Collision Avoidance	CA		

Table 2.1: *The terminology used in Table 2.2 to define the control objectives of the presented methods and their utility towards autonomous UAS.*

in LQR and MPC based FCS design; as two distinct branches of optimal control theory. Model based controllers are also seen most commonly associated with multi-objective control (see Table 2.2) due to their ability to consider low-level and high-level behaviours simultaneously using a model of the system.

An observation made clear by the literature is that the assumptions about the vehicle dynamics, modelling approaches and DOF considered in UAS controller design differ widely. With the applications and capabilities of UAS only set to expand, it appears that developments towards effective modelling strategies and formalities for describing UAS are still immature. The opportunity to propose a novel mechanism for formally describing the dynamics of variable UAS topologies is evident and represents a path to enhancing model-based control and simulation. The use of symbolic variables in the representation of UAS topologies allows them to be expressed as a sequence abstract components. Considering the aggregate reaction between each symbolic component allows the behaviour of the system to be characterised without parameterisation or loss of generality. This presents some interesting opportunities for enhancing autonomy through model-based FCS design whilst maintaining a level of generality to make the approach applicable for more complex UAV configurations. Developments into this symbolic modelling framework are described in the following chapter.

Table 2.2: *An in-exhaustive comparison of the breath of methods applied to FCS design and their utility as an aspect of autonomy within modern UAS. The UAS vehicle model, approach to dynamic modelling and degrees of freedom are also given with reference to the FCS's control objective. The table of symbols of this are given in Table 2.1 [37, 49, 168, 169, 278].*

Approach Name	**Vehicle Dynamics**	**Modelling Approach**	**Control Objective**	**Considered DOF**	**Control Locality**	**Autonomy Level**
Classic/PID Methods						
PID with Back-Stepping [32, 62]	Quadcopter	Newtonian	Al	6	L	Low
Adaptive key-frame manoeuvring [154, 157]	Quadcopter	Newtonian	Al,At,T	6	L	Medium
Gain-Scheduled/fuzzy PID [132]	Hybrid	Newtonian	At	6	L	Low
LQR/LQG Control						
PID vs. LQR [15, 32]	Quadcopter	Euler-Lagrange	At	3	L	Low
LQR stabilisation/trajectory tracking [83]	Quadcopter	Newtonian	Al,At,T	6	L	Medium
UKF Enhanced hover LQR [118]	Helicopter	Newtonian	Al,At	6	L	Low
Model-Predictive Control						
Linear MPC vs PID Tracking/Stabilisation [152]	Quadcopter	Euler-Lagrange	Al,At,T	6	L	Medium
Multi-Model Linear MPC [121]	Helicopter	Newtonian	Al,At	6	L	Low
NMPC Formation Flight [43, 221]	3D Dublin's Car	-	T,F,CA	5	C,De	High
Explicit Closed-Loop NMPC [151]	Helicopter	Newtonian	Al,At,T	6	L	Medium
Distributed Linear MPC [98]	Single Integrator	-	T,F,CA	3	Di	High

Approach Name	Vehicle Dynamics	Modelling Approach	Control Objective	Considered DOF	Control Locality	Autonomy Level
Backstepping						
Adaptive Constraints via Back-stepping [79]	Fixed-Wing	Newtonian	Al,At	6	L	Low
Backstepping and Parameter Adaptation [198]	3D Dublin's Car	-	T	5	L	Medium
Artificial Fields and Backstepping [253]	Quadcopter	Newtonian	Al,At,T,F	6	C,De	High
Fuzzy/Gain Scheduling						
Adaptive Gain-Scheduling / Sliding Mode [214, 265]	Quadcopter	Newtonian	Al,At	6	L	Low
Robust Gain-Scheduling [247]	Fixed-wing	Newtonian	Al	1	L	Low
Adaptive Model, Gain-Scheduled PID [116]	Quadcopter	Newtonian	At	6	L	Low
Biologically Inspired						
Learned Altitude Control via NN [90]	Helicopter	Newtonian	Al	6	L	Low
Adaptive Backstepping and NN [145]	Fixed-wing	Newtonian	Al,At	6	L	Low
Evolutionary GA planning [173, 181]	Single Integrator	-	T	3	L,C	Medium
Parameter Learning by Adaptive NN [146]	Helicopter	-	Al,At,T	5	L	Medium
NN based stabilsation and tuning [273]	Quadcopter	Euler-Lagrange	Al,At,T	6	L	Low
Other						
Formation Constraint Optimisation [24]	Fixed-wing	-	T,F	3	C,De	High
Connectivity Robustness [232]	Single Integrator	-	T,F	3	C,De	High
Information-based learning for dual control [10]	Single Integrator	MDP	T	2	C,De	Medium

Chapter 3

Symbolic Modelling & Control

In the literature, the generation of dynamic behaviours is typically achieved through the application of Newtonian physics or Euler-Lagrange. Here it is shown how these principles may be abstracted to form a generalised approach to describing *unmanned aerial systems* (UAS) using symbolic approach to model synthesis (see Figure 3.1). High fidelity models of an UAS or *unmanned aerial vehicles* (UAVs) flight characteristics is vital in the design of effective control algorithms, as often the behaviour is complex and non-linear. As a result, it is often necessary to apply linearisation methods (e.g. small perturbation theory, feedback linearisation) or assume known operating conditions to reduce the system before conventional control techniques can be applied. In Section 3.1 the Newtonian rigid-body representation is introduced as the basis for describing primitive UAS components. It is then shown in Section 3.2, how these components may be generalised to described their interactions with other component groups and sub-assemblies. By attributing these influences symbolically to the overall system behaviour, the approach is shown able to generate parametric descriptions of several known UAS configurations as shown in Figure 3.1. This is demonstrated through the formation of their non-linear equations of motion that characterise the system.

The approach is initially applied to an F450 quadcopter where it is shown how key control parameters can be defined symbolically for the purpose of *linear quadratic regulation* (LQR) and *model predictive control* (MPC). Demonstration on more complex systems can then be seen in the definition of a parameter-less tricopter and deltacopter systems.

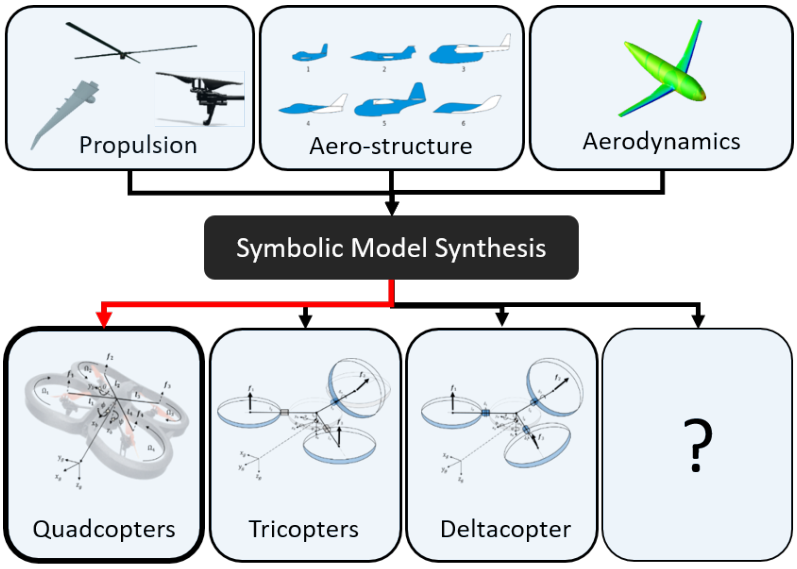


Figure 3.1: *The symbolic modelling approach to describing various unmanned aerial systems (UAS) where the interactions between distinct groups (e.g. propulsion, aerodynamics..) are defined as symbolic components and force centres. This generic approach is used in the assembly of the characteristic dynamics of several popular UAS configurations as part of these works.*

3.1 Rigid Body Analysis

The airframe of an *micro aerial vehicle* (MAV) is assumed able move to freely in 3D Cartesian space; with six *degrees of freedom* (DOF) describing the translations and rotations of the body in the NED coordinate axes. We define it's linear and angular body axis rates to be $\boldsymbol{v} = [u, v, w]^T$ and $\boldsymbol{\omega} = [p, q, r]^T$ respectively. The airframe mass and inertia tensor are given the symbolic parameters m_b and $\mathbf{I}_b \in \mathbb{R}^{3\times3}$ respectively. Newtons equations describing the motion of a generic rigid-body can then be written:

$$m_b(\dot{\boldsymbol{v}} + \boldsymbol{\omega} \times \boldsymbol{v}) = \boldsymbol{f} \tag{3.1}$$

$$\mathbf{I}_b\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\mathbf{I}_b\boldsymbol{\omega}) = \boldsymbol{\tau} \tag{3.2}$$

In Equation (3.1) the linear and angular body accelerations are denoted $\dot{\boldsymbol{v}}$ and $\dot{\boldsymbol{\omega}}$ respectively. The vectors $\boldsymbol{f}$ and $\boldsymbol{\tau}$ are then parameters describing the forces and torques acting on the body due to external influences [64]. To be able to express the dynamics of a rigid body as a 6DOF system more conveniently, it may be rewritten as a matrix expression of the form shown in Equation (3.3). We define the coefficient matrices using the relationship $\boldsymbol{x} \times \boldsymbol{y} = -\boldsymbol{y} \times \boldsymbol{x}$. Here $S(\boldsymbol{x})$ denotes the skew-symmetric operation of

vector $\boldsymbol{x}$.

$$\begin{bmatrix} m_b\mathbf{I}_{3x3} & 0_{3x3} \\ 0_{3x3} & \mathbf{I}_b \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{v}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} + \begin{bmatrix} m_b S(\boldsymbol{\omega}) & 0_{3\times3} \\ 0_{3\times3} & -S(\mathbf{I}_b\boldsymbol{\omega}) \end{bmatrix} \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{\tau} \end{bmatrix} \tag{3.3}$$

Which may also be expressed as:

$$\mathbf{M}\ddot{\boldsymbol{x}} + \mathbf{C}\dot{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{\tau} \end{bmatrix} = \boldsymbol{\nu} \tag{3.4}$$

Here the matrices $\mathbf{M}$ and $\mathbf{C}$ parameterise the body's inertial, Coriolis and centripetal contributions respectively in terms of a local generalised coordinate vector $\dot{\boldsymbol{x}} = [\boldsymbol{v}, \boldsymbol{\omega}]^T$. Using these equations it is then possible to describe the acceleration of the body i in the frame of i via some rearrangement in Equation 3.4 [64]:

$$\ddot{\boldsymbol{x}} = \mathbf{M}^{-1}\left(\boldsymbol{\nu} - \mathbf{C}\dot{\boldsymbol{x}}\right) \tag{3.5}$$

The assumption of a rigid-body fuselage is ubiquitous within the UAV community; and is typically sufficient in the modelling of smaller systems such as MAVs where the effects of aeroelasticity are minimal. By assuming the motion of the UAV is measured relative to it's local coordinate axes, we are able to construct a local representation of the fuselage motion by assigning the relationship $\dot{\boldsymbol{x}} = [u, v, w, p, q, r]^T = [\dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi}]^T$.

3.2 Attributing Influences

The inertial behaviour of aircraft fuselage has been defined symbolically. The forces and torques contributing to dynamical behaviour of the UAV from further sources may grouped in accordance to their frames of reference. Influences acting within the inertial frame and body frame influences are then parameterised by the symbolic vectors $\boldsymbol{\nu}_g \in \mathbb{R}^{6\times1}$ and $\boldsymbol{\nu}_b \in \mathbb{R}^{6\times1}$ respectively. The relationship between the two may then be defined by the transformation between the fixed inertial frame O_g, and the airframe body O_b. This may be expressed in terms of the body's Euler rotations ψ, θ and ψ about the x, y

and z inertial axes respectively:

$$\mathbf{R}^x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \tag{3.6}$$

$$\mathbf{R}^y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \tag{3.7}$$

$$\mathbf{R}^z(\psi) = \begin{bmatrix} \cos(\psi) & -sin(\psi) & 0 \\ \sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.8}$$

In this thesis the pose of a given aircraft is described using the standard aerospace convention [74, 75]. Here, the pose of the fuselage is defined by the rotation sequence $\mathbf{R}(\phi, \theta, \psi) = \mathbf{R}^x(\psi)\mathbf{R}^y(\theta)\mathbf{R}^z(\phi)$ resulting in Expression (3.9):

$$\mathbf{R}(\phi, \theta, \psi) = \begin{bmatrix} c(\theta)c(\psi) & c(\theta)s(\psi) & -s(\theta) \\ s(\phi)s(\theta)c(\psi) - c(\phi)s(\psi) & c(\phi)c(\psi) + s(\phi)s(\theta)s(\psi) & s(\phi)c(\theta) \\ s(\phi)s(\psi) + c(\phi)s(\theta)c(\psi) & c(\phi)s(\theta)s(\psi) - s(\phi)c(\psi) & c(\phi)c(\theta) \end{bmatrix} \tag{3.9}$$

Here $c(\theta)$ and $s(\theta)$ represent the sin and cos operations of angle θ respectively. For clarity, $\mathbf{R}_b^g = \mathbf{R}(\phi, \theta, \psi) \in SO(3)$ defines a special orthonormal group with properties such that $\mathbf{I} = (\mathbf{R}_g^b)^T\mathbf{R}_g^b$ and $\mathbf{R}_b^g = (\mathbf{R}_g^b)^T$. The term $\tilde{\mathbf{R}}_g^b = diag(\mathbf{R}_g^b, \mathbf{R}_g^b)$ is also defined to represent the diagonal concatenation of $\mathbf{R}_g^b$. The net forces and torques acting within the body axes can be expressed as the sum of the forces, grouped as contributions from specific UAS design parameters in Equation (3.10):

$$\boldsymbol{\nu} = \boldsymbol{\nu}_{prop} + \boldsymbol{\nu}_{gyro} + \tilde{\mathbf{R}}_b^g \left(\boldsymbol{\nu}_{aero} + \boldsymbol{\nu}_{grav}\right) \tag{3.10}$$

Here $\boldsymbol{\nu}_{prop}$, $\boldsymbol{\nu}_{gyro}$ and $\boldsymbol{\nu}_{aero}$ represent configurational parameters describing the propulsion system and aerodynamic properties of the UAS. Contributions due to drag are neglected in these works and so it is assumed that $\boldsymbol{\nu}_{aero} = \mathbf{0}_{6\times1}$ in the calculation of $\boldsymbol{\nu}$. The gravitational force acting on the airframe $\boldsymbol{\nu}_{grav}$ is introduced as constant acceleration in

Figure 3.2: *a) The F450 quadcopter platform and its axis-aligned coordinate axes used to provide numeric context for the symbolic modelling process. b) An exploded view of the F450's components contributing to the inertial parameters of the body and rotor assemblies.*

the inertial frame, parameterised by it's mass m_b in Equation (3.11).

$$\boldsymbol{\nu}_{grav} = m_b \begin{bmatrix} 0 \\ 0 \\ g \\ \mathbf{0}_{3\times 1} \end{bmatrix} \tag{3.11}$$

From Equations (3.5) and (3.10) it's clear we are able to define UAS configurations by specifying symbolic definitions for it's method of propulsion $\boldsymbol{\nu}_{prop}$ and it's aerodynamic description $\boldsymbol{\nu}_{aero}$.

3.3 Rotor Dynamics

The term "multicopter" is given to an MAV configuration that is composed of multiple propulsive groups using rotors as their means of propulsion. These groups typically consist of a series of electro-mechanical components; such as an *electronic speed controller* (ESC), brush-less motor and the rotor blade assembly. Occasionally this may include supporting actuators for blade manipulation (see Section (3.6)). In these works, the term "nacelle" is used to refer to static elements of these propulsive groups as a sub-assembly within the MAV's airframe that includes the supporting arm (gantry) separating it from the central fuselage.

In the literature, *blade element theory* is frequently cited as the method used to define a relationship between the rotor's rotational speed Ω_i and the associated thrust $\boldsymbol{f}_{r_i}$. This relationship is commonly assumed to be that $\boldsymbol{f}_{r_i} \propto \Omega^2$ around the hover condition [19, 34, 64, 175]. This relationship is known to be an approximation of the complex

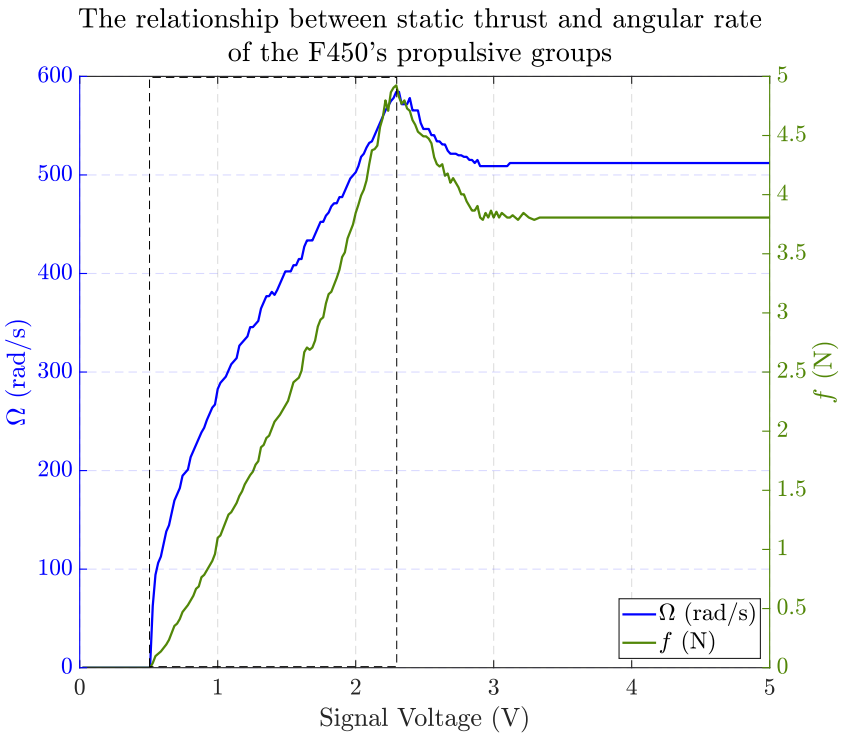


Figure 3.3: *An analysis of the rotational speed* Ω *and the associated static thrust* $\boldsymbol{f}$ *across the servo* pulse-width modulation *(PWM) set-point range taken from the F450 quadcopter. The region frequently cited as proportional is indicated.*

aerodynamics interactions of the blade and so a experimental verification is necessary. Figures 3.2(a) and 3.2(b) present the F450 quadcopter; the system used to experimentally validate the rotor model and provide the numeric parameterisation necessary for controller evaluation. The results from an examination of the F450's static and dynamic thrust properties are shown in Figures 3.3 and 3.4. Here, this assumption can be seen to be valid if the system is assumed to be operating within the indicated operational window. In this case, this assumption is viable for representing a quadcopter operating around the *straight and level* (S&L) condition in still air as is consistent with the literature [64].

It is implied by this assumption that the aircraft is in a near-hover scenario; such that the air is static with a mean velocity of zero. The blade itself is modelled as a symmetrical rigid-body with mass m_{r_i} and interia $\mathbf{I}_{r_i}$ given by Expression (3.12). It is assumed to be connected with it's associated nacelle at it's centre of gravity $\boldsymbol{p}_{r,cg}$. In other works, this assumption is challenged by investigating phenomena such as blade-flapping in [188,273], but is not considered in this thesis. The aerodynamic effects and disturbance

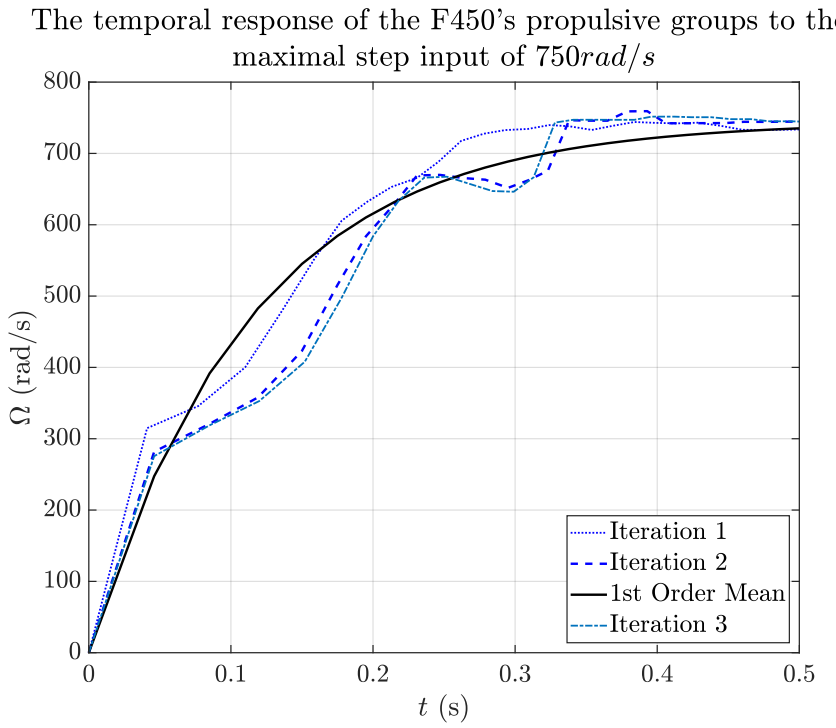


Figure 3.4: *The temporal response of the F450's propulsion group to a step input of* $750rad/s$ *which represents the largest possible control input. A first order response can be seen fitted to the mean result of three samples.*

rejection capabilities of non-planar nacelle configurations is also investigated in [182].

$$\mathbf{I}_{r_i} = \begin{bmatrix} I_r^{xx} & 0 & 0 \\ 0 & I_r^{yy} & 0 \\ 0 & 0 & I_r^{zz} \end{bmatrix} \tag{3.12}$$

The rotor thrust $\boldsymbol{f}_{r_i}$ is characterised in local axes by it's thrust coefficient, cross-sectional area, radius, rotational speed and the ambient air density as C_t, a, r, Ω_i and ρ respectively. The resulting force in the frame of rotor assembly can be written as a function of the rotor geometry and ambient aerodynamic properties as seen in Equation (3.13).

$$\boldsymbol{f}_{r_i} = \begin{bmatrix} f_{r_i}^x \\ f_{r_i}^y \\ f_{r_i}^z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \cdot \rho C_t a \left(r\Omega_i\right)^2 = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \cdot k_p \Omega_i^2 \tag{3.13}$$

The aerodynamic coefficients seen in Equation (3.13) are grouped under the thrust constant k_p for convenience. The aerodynamic force resisting the motion of blade i, generates a reaction torque $\boldsymbol{\tau}_{r_i}$ at the motors hub referred to as the *hub force* [32, 167]. The magnitude of the force is proportional to the rotors angular velocity $\Omega_{r_{1:4}}$ similar to

the relationship seen in Equation (3.13). This torque, again expressed in the frame of the rotor assembly, is written in terms of a known hub force coefficient C_h, a rotation direction parameter κ_i and the constants defining the blades geometry in (3.14).

$$\boldsymbol{\tau}_{r_i} = \begin{bmatrix} \tau_{r_i}^x \\ \tau_{r_i}^y \\ \tau_{r_i}^z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \kappa_i \end{bmatrix} \cdot \rho C_h a (r\Omega_i)^2 = \begin{bmatrix} 0 \\ 0 \\ \kappa_i \end{bmatrix} \cdot k_h \Omega_i^2 \tag{3.14}$$

The forces $\boldsymbol{f}_{r_i}, \boldsymbol{\tau}_{r_i}$ are assumed to be specified in the coordinates of rotor assembly i. In addition to the static thrust properties, the propulsion groups used in multicopter systems have an associated rise-time t_s to achieve the set-point angular velocity Ω_i. This relationship was found experimentally to be of first order for propulsive groups of the F450 as shown in Figure 3.4. The associated continuous-time transfer function representing this relationship can then be seen in Equation (3.15).

$$R(s) = \frac{k_t}{1 + t_s s} = \frac{0.0529}{1 + 0.108s} \tag{3.15}$$

Conventionally the propulsive groups used by multicopters are homogeneous in that their thrust characteristics are assumed to be identical for nacelles $i = 1 : n$. The presented behaviour is later used in the validation of the proposed control strategies of a F450 quadcopter.

3.4 Quadcopter Dynamics

The quadcopter MAV has become a popular platform for research in aerial robotics due to their comparative design simplicity, as well as their capabilities in a wide range of environments. Interest in the use of multi-rotors, as a more general field, has seen a significant rise in use cases such as search and rescue, ordinance surveying, wild-life protection and aerial cinematography in recent years. The mass distribution of the quadcopter airframe is known to be symmetric (i.e. $\mathbf{I}_b = diag(\mathbf{I})$) with it's body axes aligned with the Euler rotational axes. Each group generates a thrust vector $\boldsymbol{f}_{r_{1:4}}$ that acts at distance l from the CG $\boldsymbol{p}_{cg}$ (see Figure 3.5). Quadcopters are characterised by their four identical nacelles and rotor assemblies in fixed alignment with the body XY plane. Due to the symmetric nature of the quadcopter, $\boldsymbol{p}_{cg}$ is assumed aligned with the body axis origin.

The rotor frame is said to be aligned with the frame of the nacelle arm, whose x-axis

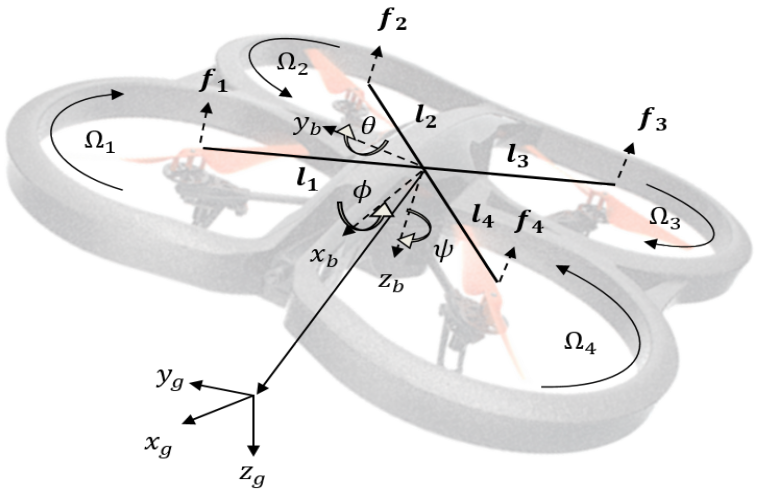


Figure 3.5: *A description of the body and inertial axis notations applied to an ARdrone quadcopter as an Xflyer configuration. The four rotor rotational rates $\Omega_{1:4}$ in relation to the thrust of each rotor $\boldsymbol{f}_{1:4}$ at distance l from the airframe CG.*

extends along the length of arm i. The transformation between the frame of rotor i and nacelle i can be written simply as $\mathbf{R}_{n_i} = \mathbf{I}_{3\times3}$ such that $\boldsymbol{f}_{n_i} = \mathbf{I}_{3\times3} \cdot \boldsymbol{f}_{r_i} \ \therefore \ \boldsymbol{f}_{r_i} \equiv \boldsymbol{f}_{n_i}$. The forces acting in the frame of rotor i can therefore be expressed in body axes via the transformation seen in Equation (3.16). Here λ_i defines the configuration angle between the x body axis and the x axis of nacelle i:

$$\mathbf{R}_{\lambda_i} = \begin{bmatrix} cos(\lambda_i) & -sin(\lambda_i) & 0 \\ sin(\lambda_i) & cos(\lambda_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.16}$$

The forces generated by each rotor act at a defined arm distance l from the origin $\boldsymbol{p}_{cg}$. Given Equation (3.16), the resulting influences of rotor i can be expressed as Equations (3.17) and (3.18).

$$\boldsymbol{f}_{prop} = \begin{bmatrix} f^x_{prop} \\ f^y_{prop} \\ f^z_{prop} \end{bmatrix} = \sum_{i=1}^{4} \left(\mathbf{R}_{\lambda_i} \boldsymbol{f}_{r_i} \right) \tag{3.17}$$

$$\boldsymbol{\tau}_{prop} = \begin{bmatrix} \tau^x_{prop} \\ \tau^y_{prop} \\ \tau^z_{prop} \end{bmatrix} = \sum_{i=1}^{4} \mathbf{R}_{\lambda_i} \left(\boldsymbol{\tau}_{r_i} + \boldsymbol{l} \times \boldsymbol{f}_{r_i} \right) \tag{3.18}$$

The resultant thrust induced force $\boldsymbol{f}$ and torque $\boldsymbol{\tau}$ are seen above expressed as the sum of the contributions from the differential rotor thrusts and hub forces of rotors 1 to 4.

MAV systems with physically rotating components also contribute angular momentum to the system under the Newtonian approach. We model this contribution as a source of gyroscopic precession $\boldsymbol{\tau}_{gyro} =\in \mathbb{R}^{3\times 1}$. For a quadcopter style MAV with four identical rotors rotating in the same plane, the torque induced by the angular momentum of the four identical rotors can then be expressed as Equation (3.19).

$$\boldsymbol{\tau}_{gyro} = \begin{bmatrix} \tau^x_{gyro} \\ \tau^y_{gyro} \\ \tau^z_{gyro} \end{bmatrix} = \mathbf{I}_r \sum_{i=1}^{4} \left(\omega \times \begin{bmatrix} 0 \\ 0 \\ \kappa_i \end{bmatrix} \Omega_i \right) \tag{3.19}$$

The contributions of the MAV's propulsion system can now be expressed in terms of their symbolic vectors $\boldsymbol{\nu}_{prop} = [\boldsymbol{f}_{prop}, \boldsymbol{\tau}_{prop}]^T$ and $\boldsymbol{\nu}_{gyro} = [0, 0, 0, \boldsymbol{\tau}_{gyro}]^T$. Here $\boldsymbol{l} = [l, 0, 0]^T$ defines a vector scaled by the arm length l aligned with the body axis x vector. Representing the quadcopter plant through a sequence of transformations $\mathbf{R}_\lambda$ allows the position of each nacelle and rotor to be generalised. The configuration vector $\lambda_i \in \{\lambda_1, \lambda_2, ...\lambda_n\}$ describing each nacelle position in the body axes is dependant on the chosen control configuration for most multicopters. Conventionally, this is either *axis-aligned* or orientated as an *X-flyer* [101, 273].

3.4.1 Axis-Aligned Configuration

The axis-aligned nacelle arrangement was popular initially due to it's comparative simplicity over the Xflyer [64]. Under this convention, the nacelles are positioned in alignment with the body axis vectors as seen in Figure 3.6. With each nacelle position aligned with the body axis vectors, the resultant torques act exclusively about the x, y and z axes. The axis-aligned plant representation is then defined by assigning their respective configuration angles $\lambda_1 = 0\,rad$, $\lambda_2 = \frac{\pi}{2}\,rad$, $\lambda_3 = \pi\,rad$ and $\lambda_4 = \frac{3}{2}\pi\,rad$ in Equations (3.16)-(3.18). Substitution of the resulting symbolic forces $\boldsymbol{\nu}_{prop} = [\boldsymbol{f}_{prop}, \boldsymbol{\tau}_{prop}]^T$ and $\boldsymbol{\nu}_{gyro} = [0, 0, 0, \boldsymbol{\tau}_{gyro}]^T$ into Equation (3.10) yields the non-linear equations of motion

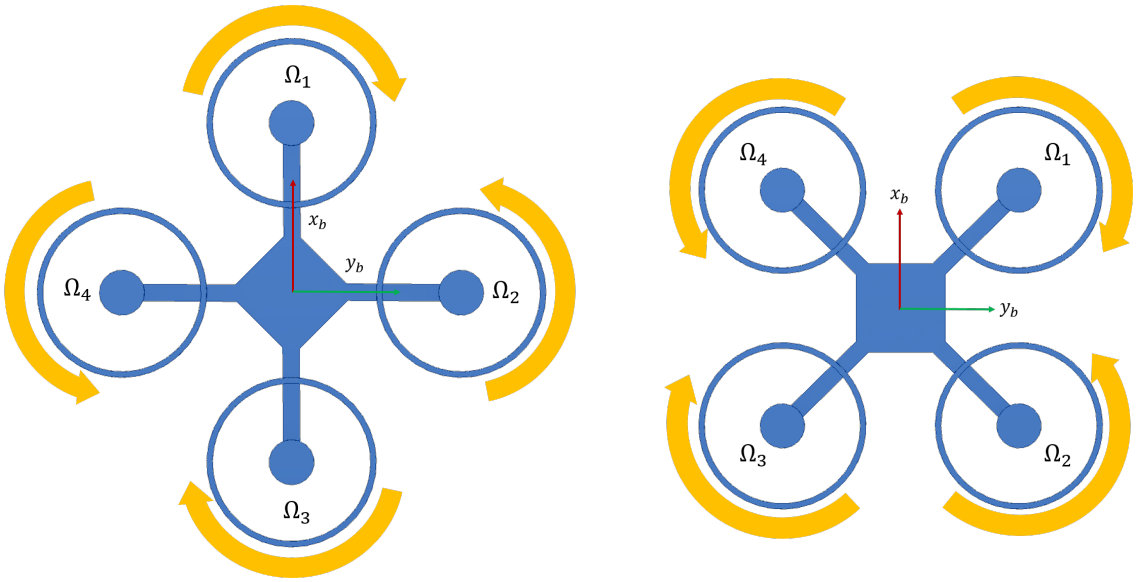


Figure 3.6: *A depiction of the* axis-aligned *and* X-flyer *control configurations for a quadcopter MAV, their respective nacelle positions and rotation directions [101, 273].*

of an *axis-aligned* quadcopter configuration shown in Equation (3.20).

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \ddot{\varphi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \dot{y}\,\dot{\psi} - \dot{z}\,\dot{\theta} - g\,\sin(\theta) \\ \dot{z}\,\dot{\varphi} - \dot{x}\,\dot{\psi} + g\,\cos(\theta)\,\sin(\varphi) \\ \dot{x}\,\dot{\theta} - \dot{y}\,\dot{\varphi} + g\,\cos(\varphi)\,\cos(\theta) - \frac{k_p}{m_b}(\Omega_1{}^2 + \Omega_2{}^2 + \Omega_3{}^2 + \Omega_4{}^2) \\ \frac{\mathbf{I}_b^{yy} - \mathbf{I}_b^{zz}}{\mathbf{I}_b^{xx}}\,\dot{\theta}\,\dot{\psi} + \frac{\mathbf{I}_r^{xx}}{\mathbf{I}_b^{xx}}\dot{\theta}(\Omega_1 - \Omega_2 + \Omega_3 - \Omega_4) - \frac{k_p\,l}{\mathbf{I}_b^{xx}}(\Omega_2{}^2 - \Omega_4{}^2) \\ \frac{\mathbf{I}_b^{zz} - \mathbf{I}_b^{xx}}{\mathbf{I}_b^{yy}}\,\dot{\varphi}\,\dot{\psi} - \frac{\mathbf{I}_r^{yy}}{\mathbf{I}_b^{yy}}\dot{\varphi}(\Omega_1 - \Omega_2 + \Omega_3 - \Omega_4) + \frac{k_p\,l}{\mathbf{I}_b^{yy}}(\Omega_1^2 - \Omega_3^2) \\ \frac{\mathbf{I}_b^{xx} - \mathbf{I}_b^{yy}}{\mathbf{I}_b^{zz}}\,\dot{\varphi}\,\dot{\theta} - \frac{k_h}{\mathbf{I}_b^{zz}}(\Omega_1{}^2 - \Omega_2{}^2 + \Omega_3{}^2 - \Omega_4{}^2) \end{bmatrix} \tag{3.20}$$

The dynamic behaviour of the axis-aligned quadcopter can be seen expressed in it's body axes as a function of it's Euler states $\boldsymbol{x}$, it's rotor speeds $\Omega_{r,1:4}$, it's design parameters and gravity.

3.4.2 X-flyer Configuration

The "X-flyer" configuration; defined by the alternative positioning of it's nacelles has recently become favoured in the MAV community. This is because, this configuration displaces the Quadcopter's nacelles to prevent obscuring the view of on-board sensors or gimbal assemblies in applications such as cinematography or *first-person-view* (FPV) racing [101, 273]. The representation of the X-flyer thrust centres in the body axes, can be attained through the substitution of the configuration angles $\lambda_1 = \frac{1}{4}\pi\,rad$, $\lambda_2 = \frac{3}{4}\pi\,rad$, $\lambda_3 = \frac{5}{4}\pi\,rad$ and $\lambda_4 = \frac{7}{4}\pi\,rad$ into Equations (3.16)-(3.18).

The resultant influences of the X-flyer's propulsion system can then be represented

by the symbolic input vectors $\boldsymbol{\nu}_{prop} = [\boldsymbol{f}_{prop}, \boldsymbol{\tau}_{prop}]^T$ and $\boldsymbol{\nu}_{gyro} = [0, 0, 0, \boldsymbol{\tau}_{gyro}]^T$. Evaluating the propulsion contributions under this convention yields the non-linear equations of the motion of a generic "X-flyer" quadcopter seen in Equation (3.21).

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \ddot{\varphi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \dot{y}\,\dot{\psi} - \dot{z}\,\dot{\theta} - g\,\sin(\theta) \\ \dot{z}\,\dot{\varphi} - \dot{x}\,\dot{\psi} + g\,\cos(\theta)\,\sin(\varphi) \\ \dot{x}\,\dot{\theta} - \dot{y}\,\dot{\varphi} + g\,\cos(\varphi)\,\cos(\theta) - \frac{k_p}{m_b}(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ \frac{\mathbf{I}_b^{yy} - \mathbf{I}_b^{zz}}{\mathbf{I}_b^{xx}}\,\dot{\theta}\,\dot{\psi} + \frac{\mathbf{I}_r^{xx}}{\mathbf{I}_b^{xx}}\dot{\theta}(\Omega_1 - \Omega_2 + \Omega_3 - \Omega_4) - \frac{\sqrt{2}\,k_p\,l}{2\,\mathbf{I}_b^{xx}}({\Omega_1}^2 + {\Omega_2}^2 - {\Omega_3}^2 - {\Omega_4}^2) \\ \frac{\mathbf{I}_b^{zz} - \mathbf{I}_b^{xx}}{\mathbf{I}_b^{yy}}\,\dot{\varphi}\,\dot{\psi} - \frac{\mathbf{I}_r^{yy}}{\mathbf{I}_b^{yy}}\dot{\varphi}(\Omega_1 - \Omega_2 + \Omega_3 - \Omega_4) + \frac{\sqrt{2}\,k_p\,l}{2\,\mathbf{I}_b^{yy}}({\Omega_1}^2 - {\Omega_2}^2 - {\Omega_3}^2 + {\Omega_4}^2) \\ \frac{\mathbf{I}_b^{xx} - \mathbf{I}_b^{yy}}{\mathbf{I}_b^{zz}}\,\dot{\varphi}\,\dot{\theta} - \frac{k_h}{\mathbf{I}_b^{zz}}({\Omega_1}^2 - {\Omega_2}^2 + {\Omega_3}^2 - {\Omega_4}^2) \end{bmatrix} \tag{3.21}$$

Expressions (3.21) and (3.20) define the continuous dynamical progression of each quadcopter system over a defined sample period Δt. In both cases the generalised systems are described algebraically without linearisation or parameterisation.

3.4.3 Controller Design

To allow the application of linear control theory, small perturbation theory can be applied to the non-linear system descriptions seen in Equations (3.20) and (3.21) in order to generate a linearised symbolic representation of the model. In associated works [64] this is demonstrated by parameterising the model presented in Equation (3.20) with the numerics of an "F450" quadcopter MAV. The MAV's state space description (3.22) is presented as follows:

$$\dot{\boldsymbol{x}} = \mathbf{A}\boldsymbol{x} + \mathbf{B}\boldsymbol{u} + \boldsymbol{w} \tag{3.22}$$

$$\boldsymbol{y} = \mathbf{C}\boldsymbol{x} + \mathbf{D}\boldsymbol{u} \tag{3.23}$$

Here, $\boldsymbol{x} = [x, y, z, \psi, \theta, \phi, \dot{x}, \dot{y}, \dot{z}, \dot{\psi}, \dot{\theta}, \dot{\phi}]^T$ defines the systems state at time t. The systems input vector is defined as squared velocities of the four rotors $\boldsymbol{u} = [\Omega_1^2, \Omega_2^2, \Omega_3^2, \Omega_4^2]^T$. The "aligned" quadcopter plant matrix $\mathbf{A}$ and input matrix $\mathbf{B}$, are given in Equations (3.25) and (3.24) respectively. The observation matrix is denoted $\mathbf{C} = \mathbf{I}_{12x12}$, feed-forward matrix $\mathbf{D} = \mathbf{0}_{12x4}$. Finally, the term $\boldsymbol{w} = \mathcal{N}(\mathbf{0}_{12\times1},\, diag\{\mathbf{I}_{3\times3}\sigma_p^2, \mathbf{I}_{3\times3}\sigma_\Theta^2, \mathbf{I}_{3\times3}\sigma_v^2, \mathbf{I}_{3\times3}\sigma_\omega^2\}^T)$ is introduced as a source of state noise.

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{k_p}{m_b} & -\frac{k_p}{m_b} & -\frac{k_p}{m_b} & -\frac{k_p}{m_b} \\ \frac{\mathbf{I}_r^{xx}}{2\,\mathbf{I}_b^{xx}\,\Omega_1}\,\dot{\theta} & \left(-\frac{\mathbf{I}_r^{xx}}{2\,\mathbf{I}_b^{xx}\,\Omega_2}\right)\,\dot{\theta} - \frac{k_p\,l}{\mathbf{I}_b^{xx}} & \frac{\mathbf{I}_r^{xx}}{2\,\mathbf{I}_b^{xx}\,\Omega_3}\,\dot{\theta} & \left(-\frac{\mathbf{I}_r^{xx}}{2\,\mathbf{I}_b^{xx}\,\Omega_4}\right)\,\dot{\theta} + \frac{k_p\,l}{\mathbf{I}_b^{xx}} \\ \left(-\frac{\mathbf{I}_r^{yy}}{2\,\mathbf{I}_b^{yy}\,\Omega_1}\right)\,\dot{\varphi} + \frac{k_p\,l}{\mathbf{I}_b^{yy}} & \frac{\mathbf{I}_r^{yy}}{2\,\mathbf{I}_b^{yy}\,\Omega_2}\,\dot{\varphi} & \left(-\frac{\mathbf{I}_r^{yy}}{2\,\mathbf{I}_b^{yy}\,\Omega_3}\right)\,\dot{\varphi} - \frac{k_p\,l}{\mathbf{I}_b^{yy}} & \frac{\mathbf{I}_r^{yy}}{2\,\mathbf{I}_b^{yy}\,\Omega_4}\,\dot{\varphi} \\ -\frac{k_h}{\mathbf{I}_b^{zz}} & \frac{k_h}{\mathbf{I}_b^{zz}} & -\frac{k_h}{\mathbf{I}_b^{zz}} & \frac{k_h}{\mathbf{I}_b^{zz}} \end{bmatrix} \tag{3.24}$$

Here the hover condition is selected as the operating point for which a linear stability controller can be designed. In such conditions, it can be assumed that the airframe is near S&L. It may be assumed that the initial attitude angles $\phi, \theta \approx 0$ and axis rates are small; $\dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi} \approx 0$. For a quadcopter, it may also be assumed that the rotors speeds are near equal; $\Omega_1 \approx \Omega_2 \approx \Omega_3 \approx \Omega_4$. Equally, by retaining the symbolic definitions of $\mathbf{A}$ and $\mathbf{B}$, the state-space representation may be redefined dynamically upon new measurements of the axis rates and rotor set-point speeds.

$$
\mathbf{A} = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & -g\cos(\theta) & 0 & 0 & \dot{\psi} & -\dot{\theta} & 0 & -\dot{z} & \dot{y} \\
0 & 0 & 0 & g\cos(\varphi)\cos(\theta) & -g\sin(\varphi)\sin(\theta) & 0 & -\dot{\psi} & 0 & \dot{\varphi} & \dot{z} & 0 & -\dot{x} \\
0 & 0 & 0 & -g\cos(\theta)\sin(\varphi) & -g\cos(\varphi)\sin(\theta) & 0 & \dot{\theta} & -\dot{\varphi} & 0 & -\dot{y} & \dot{x} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\mathbf{I}_b^{yy}-\mathbf{I}_b^{zz}}{\mathbf{I}_b^{xx}}\,\dot{\psi} + \frac{\mathbf{I}_r^{xx}\,\Omega_1-\mathbf{I}_r^{xx}\,\Omega_2+\mathbf{I}_r^{xx}\,\Omega_3-\mathbf{I}_r^{xx}\,\Omega_4}{\mathbf{I}_b^{xx}} & \frac{\mathbf{I}_b^{yy}-\mathbf{I}_b^{zz}}{\mathbf{I}_b^{xx}}\,\dot{\theta} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\mathbf{I}_b^{zz}-\mathbf{I}_b^{xx}}{\mathbf{I}_b^{yy}}\,\dot{\psi} - \frac{\mathbf{I}_r^{yy}\,\Omega_1-\mathbf{I}_r^{yy}\,\Omega_2+\mathbf{I}_r^{yy}\,\Omega_3-\mathbf{I}_r^{yy}\,\Omega_4}{\mathbf{I}_b^{yy}} & 0 & \frac{\mathbf{I}_b^{zz}-\mathbf{I}_b^{xx}}{\mathbf{I}_b^{yy}}\,\dot{\varphi} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\mathbf{I}_b^{xx}-\mathbf{I}_b^{yy}}{\mathbf{I}_b^{zz}}\,\dot{\theta} & \frac{\mathbf{I}_b^{xx}-\mathbf{I}_b^{yy}}{\mathbf{I}_b^{zz}}\,\dot{\varphi} & 0
\end{bmatrix} \tag{3.25}
$$

3.4.4 Linear Quadratic Regulation

The LQR controller is a classical approach to control often cited in the context of UAV control. The LQR cost function presented in Equation (3.26) parameterises the state error $\boldsymbol{e}_k = \boldsymbol{r}_k - \boldsymbol{x}_k$, input magnitude $\boldsymbol{u}_k$ and their penalisation matrices $\mathbf{Q}$ and $\mathbf{R}$ respectively. Here we do not penalise the terminal state.

$$V_k = min \int \left[\boldsymbol{e}_k^T \mathbf{Q} \boldsymbol{e}_k + \boldsymbol{u}_k^T \mathbf{R} \boldsymbol{u}_k\right] \Delta t \tag{3.26}$$

For the solution to the optimal control problem to be positive (i.e $V_k \geq 0$) it implies that $\mathbf{Q}$ and $\mathbf{R}$ must be positive definite. The performance of the controller V_k is optimised through the selection of $\boldsymbol{u}$ subject to the systems linear dynamics seen in Equation (3.22) and the state constraints defined by the *matrix Riccati equation*:

$$\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} + \mathbf{Q} = 0 \tag{3.27}$$

The solution $\mathbf{P}$ to Equation (3.27) then defines the feedback matrix:

$$\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}. \tag{3.28}$$

The resulting state feedback is applied to the system in the form:

$$\boldsymbol{u}_k = -\mathbf{K}\boldsymbol{e}_k \tag{3.29}$$

The feedback gain $\mathbf{K}$ is then used to instigate a response proportional to the error feedback $\boldsymbol{e}_k$ in the MAVs dynamic expressions in Equation (3.20).

3.4.5 Model Predictive Control

MPC is another form of optimal control that utilises the predicted performance of the system over a defined horizon, in order to compute the optimal instantaneous control inputs. Using the linearised model seen in Equation (3.22), we are able to formulate a prediction at time t_k of how the systems output will evolve over a defined time horizon. This is achieved by predicting the evolution of the systems states and actuator contributions over a defined horizon h_i [155]. The state prediction matrices $\mathbf{F}$ and input prediction

matrices $\mathbf{G}$ are formulated in Equation (3.30):

$$\mathbf{F} = \begin{bmatrix} \mathbf{A} \\ \mathbf{A}^2 \\ \mathbf{A}^3 \\ : \\ \mathbf{A}^{h_i} \end{bmatrix}, \mathbf{G} = \begin{bmatrix} \mathbf{B} & 0 & \dots & 0 \\ \mathbf{AB} & \mathbf{B} & \dots & : \\ : & : & \ddots & : \\ \mathbf{A}^{h_{i-1}}\mathbf{B} & \mathbf{A}^{h_{i-2}}\mathbf{B} & \cdots & \mathbf{B} \end{bmatrix} \tag{3.30}$$

The resultant system outputs at each discrete horizon step can similarly be written as $\tilde{\boldsymbol{z}}_k$. Here the notation $\tilde{\boldsymbol{x}}$ denotes the horizon-concatenated vector of $\boldsymbol{x}$:

$$\tilde{\boldsymbol{z}}_k = \tilde{\mathbf{C}} \left[\mathbf{F}\boldsymbol{x}_k + \mathbf{G}\tilde{\boldsymbol{u}}_k \right] \tag{3.31}$$

We assume that at a given reference point $\boldsymbol{r}_k$ there exists a steady state $\boldsymbol{x}_k$ with an associated steady-state input $\boldsymbol{u}_k$. It is then possible to define the input required to maintain the reference through the dynamic relation seen in Equation (3.32):

$$\begin{bmatrix} \mathbf{I}_{12\times 12} - \mathbf{A} & -\mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{u}_k \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{12x1} \\ \boldsymbol{r}_k \end{bmatrix} \tag{3.32}$$

Solving Equation (3.32) for a given reference output $\boldsymbol{r}_k$ allows the steady state $\boldsymbol{x}_k$ and input deviation $\boldsymbol{u}_k$ to be determined. Calculation of this value allows the absolute input to be defined as a difference between the reference control and the control signal generated by the controller. The predicted error over the horizon is defined as $\tilde{e}_k = \tilde{r}_k - \tilde{z}_k$. Similar to Section 3.4.4, penalisation of the tracking error, control weightings and terminal error are introduced through the weighting matrices $\mathbf{Q}$, $\mathbf{R}$ and $\mathbf{N}$ respectively. The associated penalisations over the complete horizon are are therefore represented by the concatenated matrices $\tilde{\mathbf{Q}}$ and $\tilde{\mathbf{R}}$.

$$\tilde{\mathbf{Q}} = \begin{bmatrix} \mathbf{Q} & 0 & \dots & 0 \\ 0 & \mathbf{Q} & \dots & : \\ : & : & \ddots & : \\ 0 & \dots & \dots & \mathbf{N} \end{bmatrix}, \tilde{\mathbf{R}} = \begin{bmatrix} \mathbf{R} & 0 & \dots & 0 \\ 0 & \mathbf{R} & \dots & : \\ : & : & \ddots & : \\ 0 & \dots & \dots & \mathbf{R} \end{bmatrix} \tag{3.33}$$

To be able to determine the optimal control inputs under the regime of linear MPC, the problem must be formulated as a convex cost function. This cost function is derived in terms of the total predicted error $\tilde{e}_k$ over each discrete horizon step, the aforemen-

tioned penalisations $\mathbf{Q}, \mathbf{R}$ and $\mathbf{N}$ and the exerted control efforts $\boldsymbol{u}_k$ as seen in Equation (3.34) [139, 155]:

$$V_k = \sum_{k=1}^{h_i} \left(||(\boldsymbol{r}_k - \boldsymbol{z}_k)||^2 \mathbf{Q} + ||\boldsymbol{u}_k||^2 \mathbf{R} \right) + ||(\boldsymbol{r}_k - \boldsymbol{z}_k)||^2 \mathbf{N} \tag{3.34}$$

Here $||\boldsymbol{u}_k||$ is used to describe the Euclidean norm of the input. Substituting the horizon prediction matrix expressions allows Equation (3.34) to be redefined as quadratic coefficients of the control input $\boldsymbol{u}_k$. The terminal cost matrix $\mathbf{N}$ can be seen neglected, with no value assigned to a terminal angular rate:

$$V_k = \tilde{\mathbf{u}}_k^T [\Theta^T \tilde{\mathbf{Q}} \Theta + \tilde{\mathbf{R}}] \tilde{\mathbf{u}}_k - 2 \tilde{\mathbf{u}}_k^T \Theta^T \tilde{\mathbf{Q}} \tilde{\mathbf{e}}_k + \tilde{\mathbf{e}}_k^T \tilde{\mathbf{Q}} \tilde{\mathbf{e}}_k \tag{3.35}$$

where:

$$\Theta = \tilde{\mathbf{C}} \mathbf{G} \tag{3.36}$$

If we define $\mathbf{H} = \Theta^T \tilde{\mathbf{Q}} \Theta + \tilde{\mathbf{R}}$ and $\mathbf{G} = -2\Theta^T \tilde{\mathbf{Q}} \tilde{\mathbf{e}}_k$, the cost function takes the quadratic form:

$$V_k = \tilde{\mathbf{u}}_k^T \mathbf{H} \tilde{\mathbf{u}}_k + \tilde{\mathbf{u}}_k^T \mathbf{G} + \tilde{\mathbf{e}}_k^T \tilde{\mathbf{Q}} \tilde{\mathbf{e}}_k \tag{3.37}$$

The optimal control sequence then occurs where equation (3.37) is minimal, subject to the MAV's dynamics. This optimisation operation was computed directly using the Matlab® function *quadprog*. The first of the optimal control inputs is then selected at t_k and applied to the non-linear plant model seen in Expression (3.20). One of the principle advantages of MPC based control is the inclusion of constraints. A series of design constraints were therefore introduced to represent the performance limits of the quadcopter aircraft:

$$-1 rad/s \geq \phi, \theta \leq 1 rad/s \tag{3.38}$$

$$0 rad/s \geq \Omega_{1:4} \leq 580 rad/s \tag{3.39}$$

A regime was applied to ensure the MAV remains within a defined set of maximum deflection rates to aid in maintaining stability (3.38). The above input conditions (3.39) were then selected to represent the physical limits of the actuators and enforce operation within the linear region identified in Figure 3.3.

Symbol	Value	Units	Symbol	Value	Units
r	1.268×10^{-1}	m	m_b	1.224	kg
a	1.400×10^{-3}	m^2	$\mathbf{I}_b^{xx}$	0.279×10^{-1}	kgm^2
ρ	1.225	kgm^{-3}	$\mathbf{I}_b^{yy}$	0.549×10^{-1}	kgm^2
C_t	0.531	–	$\mathbf{I}_b^{zz}$	0.281×10^{-1}	kgm^2
C_h	4.243×10^{-3}	–	$\mathbf{I}_r^{xx}$	9.880×10^{-6}	kgm^2
g	9.807	ms^{-2}	$\mathbf{I}_r^{yy}$	9.692×10^{-5}	kgm^2
l	3.223×10^{-1}	m	$\mathbf{I}_r^{zz}$	9.059×10^{-5}	kgm^2

Table 3.1: *The design parameters representing the F450 Quadcopter orientated in a 'aligned' control configuration.*

3.4.6 Performance Evaluation

In the evaluation of the effectiveness of the two presented control techniques it becomes necessary to parameterise the symbolic quadcopter descriptions in order to generate numeric expressions equivalent to Equation (3.20) and (3.22). This was achieved by substituting the parameters seen in Table 3.1 resembling the F450 quadcopter in a S&L condition. The designed MPC algorithm is evaluated by observing the systems transient response to a step input. The results are then directly compared to the response of the LQR, subject to the same inputs. Both preliminary attitude controllers are designed to obtain a reference roll, pitch and yaw rate $\dot{\phi}, \dot{\theta}$ and $\dot{\psi}$, of the MAV respectively.

Initially, the tracking error and input weighting matrices $\mathbf{Q}$ and $\mathbf{R}$ of the LQR controller were used to formulate a comparison between the two controllers. Each LQR gain was then tuned heuristically until a critically damped step response was observed. The relative settling times, overshoot and sensitivity to noise could then be compared in Figure 3.7. The step response of both the LQR and MPC control regimes are shown in Figure 3.7. Both controllers can be seen to effectively stabilise the system about the desired axis rate of $0.5 rad/s$. Some minor steady-state error can be seen in the LQR output as a result of the Gaussian noise added to the system via the $\boldsymbol{w}_k$ term in Equation (3.22). In the literature, integral action (or integral feedback) is often cited as an established technique for removing steady state error under LQR and LQG control regimes. This is however beyond the scope of these works; wherein linear control is demonstrated on the proposed symbolic model formulation. As expected the formulated MPC controller can also be seen to achieve the desired attitude rates in the presented conditions. From the traces it was then possible to observe the mean settling times of the two controllers over a one hundred Monte-Carlo iterations.

As seen in Table 3.2, the LQR and MPC algorithms demonstrate similar performance in the presence of the Gaussian noise signal. The MPC controller is however shown

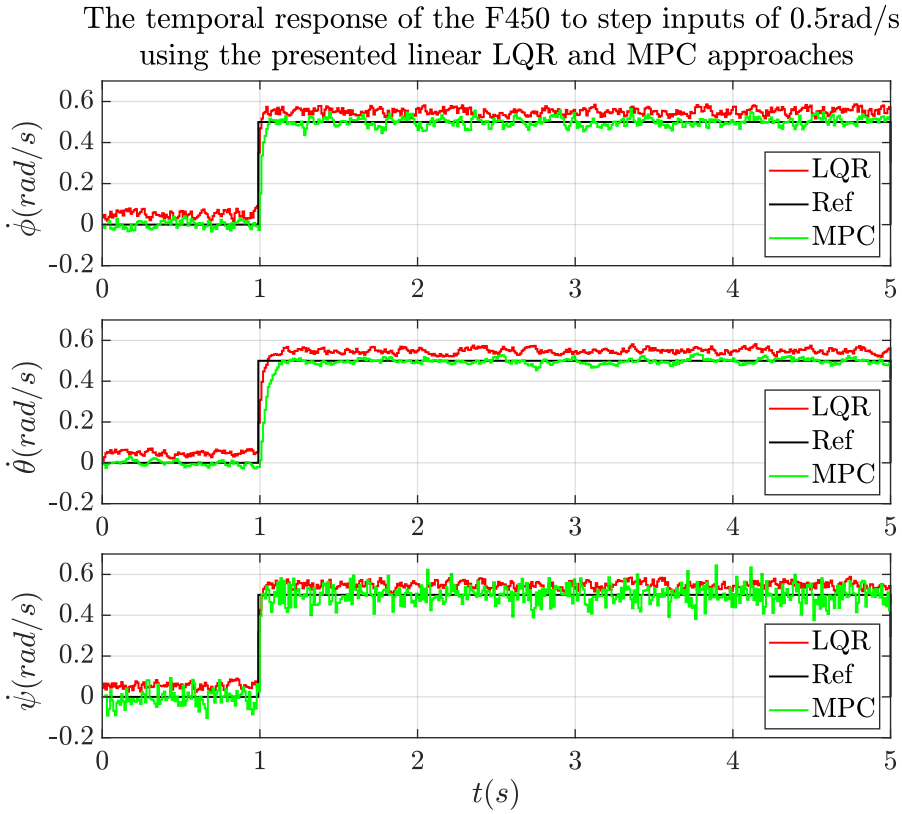


Figure 3.7: *The F450's response to a step input of 0.5rad/s about each rotational axis using the presented linear MPC and LQR control regimes. Here both methods are shown to achieve the desired reference signal in the presence of additional corruption signals. Here the MPC approach is shown to track the reference more tightly with lower steady-state error.*

Controller	Roll axis $t(s)$	Pitch axis $t(s)$	Yaw axis $t(s)$
LQR	0.100	0.220	0.110
MPC	0.090	0.200	0.050

Table 3.2: *A comparison of the temporal response of the linear LQR/MPC controllers with respect to mean settling times following an initial one hundred Monte-Carlo iterations. The LQR and MPC controllers are shown to behave similarly, with the MPC producing marginally lower settling times. This is expected as both methods are variants on linear optimal control.*

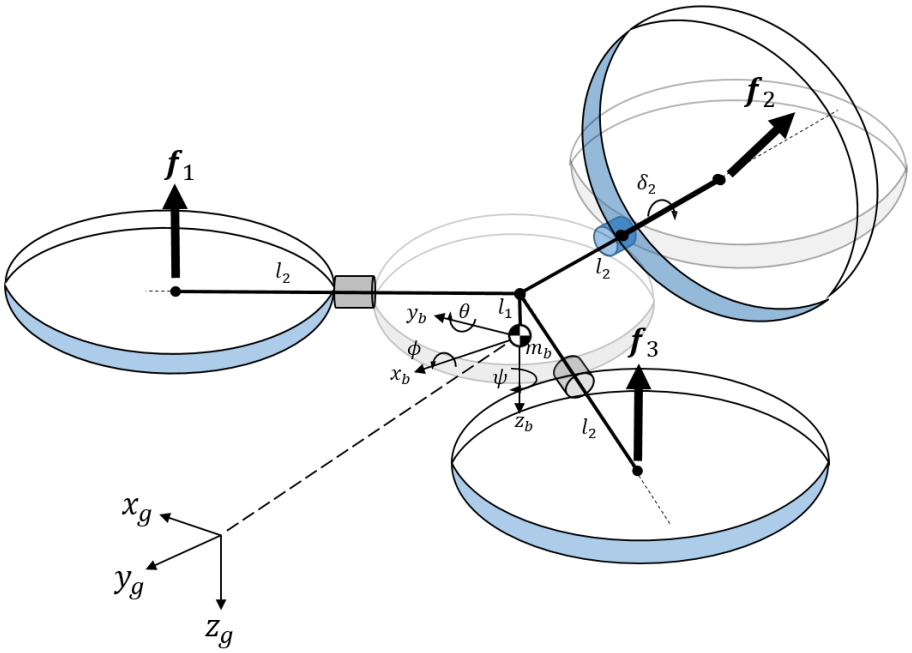


Figure 3.8: *The propulsion convention of a Tricopter style UAS. Nacelles one and three are mounted statically to the fuselage while nacelle two is able to be vectored relative to $-\boldsymbol{x}_b$ using a mounted actuator (shown in blue).*

to yield a settling time marginally slower than the LQR, with reduced overshoot or steady state error. The optimal inputs of the MPC are also computed by anticipating the behaviour of the system over a defined horizon, considering the limitations of the physical system which the LQR cannot do directly. This allows the algorithm to plan the inputs around the possibility of the VTOL system reaching actuator saturation or a limit on the physical output.

3.5 Tricopter Dynamics

By extending the symbolic method presented in Section 3.4, it is possible to formulate the dynamic descriptions of a wide range of UAS topologies. In this section we examine the *Tricopter* configuration which utilises a three propulsive group model with and additional actuator. Similar to the quadcopter, the tricopter is symmetric in the body axes. Each rotor is positioned at vertical and horizontal distances $\boldsymbol{l}_1$ and $\boldsymbol{l}_2$ from the centre of gravity $\boldsymbol{p}_{cg}$ respectively as shown in Figure 3.8. It can therefore assumed that the inertia matrix $\mathbf{I_b}$ is symmetrical with mass distribution of the rigid fuselage structure assumed fixed.

3.5.1 Thrust Vectoring

One of the defining properties of the tricopter is the servo-actuated mechanism used to drive rotor 2 through angle δ_2. The collective thrust from rotors 1 and 3, in addition to the vectored thrust of rotor 2 allow the UAS to manoeuvre in 3D space. The rotor defection δ_2 input is necessary for balancing the net angular momentum from the three rotor speeds $\Omega_{1:3}$ with a horizontal thrust component. This thrust component is the projection of $\boldsymbol{f}_2$ on the body axes XY plane. This results in a higher degree of yaw authority as this projection allows for a maximum available torque higher than conventional quadcopter systems.

With the rotor assumption presented in Section 3.3, the forces and torques acting within the frame of rotor i are characterised as $\boldsymbol{f}_i$ and $\boldsymbol{\tau}_i$ respectively. Using the principles introduced in Section 3.4, we can define a symbolic expression relating the force in the frame of the rotor to the acceleration of the CG by procedurally describing their transformation to the body axes. The propulsive groups of rotors 1 and 3 are assumed to be fixed and aligned with the nacelle initially. Expressing their rotation relative to the nacelle axes naturally takes the form $\mathbf{R}_{n_1} = \mathbf{R}_{n_3} = \mathbf{I}_{3\times3}$. Nacelle 2 however has a second control input that rotates the rotor through δ_2. The rotation is said to occur about $\boldsymbol{l}_2$ where $\boldsymbol{l}_2 \parallel -\boldsymbol{x}_b$ as seen in Figure 3.8. We may define the position of the rotors deflection δ_2 relative to the nacelle by defining the transformation $\mathbf{R}_\delta$:

$$\mathbf{R}_{n_2} = \mathbf{R}_{\delta_2} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\delta_2) & -sin(\delta_2) \\ 0 & sin(\delta_2) & cos(\delta_2) \end{bmatrix} \tag{3.40}$$

With the transformation of the rotor relative to the nacelle coordinate frame expressed in Equation (3.40), the forces and torques acting in nacelle coordinates can be written:

$$\boldsymbol{f}_{n_i} = \begin{bmatrix} f^x_{n_i} \\ f^y_{n_i} \\ f^z_{n_i} \end{bmatrix} = \mathbf{R}_{n_i}\boldsymbol{f}_{r_i} \tag{3.41}$$

$$\boldsymbol{\tau}_{n_i} = \begin{bmatrix} \tau^x_{n_i} \\ \tau^y_{n_i} \\ \tau^z_{n_i} \end{bmatrix} = \mathbf{R}_{n_i}\boldsymbol{\tau}_{r_i} \tag{3.42}$$

In considering the forces induced by rotor 2, Equations (3.41) and (3.42) describe the change in body axis projections of $\boldsymbol{f}_{r_i}$ and $\boldsymbol{\tau}_{r_i}$ as a result of the deflection δ.

The configuration matrix introduced Equation (3.16) is again invoked to describe the orientation of the nacelle frame relative to the body axes. The transform describes the clockwise rotation though the three nacelle configuration angles $\lambda_1 = \frac{1}{3}\pi\, rad, \lambda_2 = \pi\, rad$ and $\lambda_3 = \frac{5}{3}\pi\, rad$. The expression profiling the propulsive components of the complete tricopter configuration can then be seen written as Equations (3.43) and (3.44).

$$\begin{aligned} \boldsymbol{f}_{prop} &= \mathbf{R}_{\lambda_1}\mathbf{R}_{n_1}\boldsymbol{f}_{r_1} + \mathbf{R}_{\lambda_2}\mathbf{R}_{\delta_2}\boldsymbol{f}_{r_2} + \mathbf{R}_{\lambda_3}\mathbf{R}_{n_3}\boldsymbol{f}_{r_3} \\ &= \sum_{i=1}^{3} (\mathbf{R}_{\lambda_i}\boldsymbol{f}_{n_i}) \end{aligned} \tag{3.43}$$

$$\begin{aligned} \boldsymbol{\tau}_{prop} &= \mathbf{R}_{\lambda_1}(\boldsymbol{\tau}_{n_1} + (\boldsymbol{l}_1 + \boldsymbol{l}_2) \times \mathbf{R}_{n_1}\boldsymbol{f}_{r_1}) \\ &\quad + \mathbf{R}_{\lambda_2}(\boldsymbol{\tau}_{n_2} + (\boldsymbol{l}_1 + \boldsymbol{l}_2) \times \mathbf{R}_{\delta_2}\boldsymbol{f}_{r_2}) \\ &\quad + \mathbf{R}_{\lambda_3}(\boldsymbol{\tau}_{n_3} + (\boldsymbol{l}_1 + \boldsymbol{l}_2) \times \mathbf{R}_{n_3}\boldsymbol{f}_{r_3}) \\ &= \sum_{i=1}^{3} (\mathbf{R}_{\lambda_i}(\boldsymbol{\tau}_{n_i} + (\boldsymbol{l}_1 + \boldsymbol{l}_2) \times \boldsymbol{f}_{n_i})) \end{aligned} \tag{3.44}$$

The rotational components of the tricopter also introduce a gyroscopic torque as a result of their perturbation from their level position. The gyroscopic torques of each nacelle may be expressed in the body axis as Equation (3.45):

$$\boldsymbol{\tau}_{gyro} = \begin{bmatrix} \tau^x_{gyro} \\ \tau^y_{gyro} \\ \tau^z_{gyro} \end{bmatrix} = \mathbf{I}_r \sum_{i=1}^{3} (\boldsymbol{\omega} \times \boldsymbol{\omega}_i) \tag{3.45}$$

where;

$$\boldsymbol{\omega}_1 = \mathbf{R}_{\lambda_1} \begin{bmatrix} 0 \\ 0 \\ \kappa_1 \end{bmatrix} \Omega_{r_1} \tag{3.46}$$

$$\boldsymbol{\omega}_2 = \mathbf{R}_{\lambda_2}\mathbf{R}_{\delta_2} \begin{bmatrix} 0 \\ 0 \\ \kappa_2 \end{bmatrix} \Omega_{r_2} \tag{3.47}$$

$$\boldsymbol{\omega}_3 = \mathbf{R}_{\lambda_3} \begin{bmatrix} 0 \\ 0 \\ \kappa_3 \end{bmatrix} \Omega_{r_3} \tag{3.48}$$

Here, Equations (3.46) to (3.48) express the rotational speed of rotor Ω_{r_i} in relation to

the body axis rates ω. Summation of the gyroscopic and propulsive influences allows us to define the symbolic input vectors $\boldsymbol{\nu}_{prop} = [\boldsymbol{f}_{prop}, \boldsymbol{\tau}_{prop}]^T$ and $\boldsymbol{\nu}_{gyro} = [0, 0, 0, \boldsymbol{\tau}_{gyro}]^T$ as the body axis representation of the propulsive mechanism of the tricopter. If it is assumed that fuselage again behaves like a rigid body, substituting these symbolic vectors into Equation (3.1) yields the generalised non-linear equations of motion for the tricopter style MAV written as Equation (3.49):

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \ddot{\varphi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \dot{y}\,\dot{\psi} - \dot{z}\,\dot{\theta} - g\,\sin(\theta) \\ \dot{z}\,\dot{\varphi} - \dot{x}\,\dot{\psi} + g\,\cos(\theta)\,\sin(\varphi) - \frac{k_p}{m_b}(\Omega_2^2\,\sin(\delta_2)) \\ \dot{x}\,\dot{\theta} - \dot{y}\,\dot{\varphi} + g\,\cos(\varphi)\,\cos(\theta) - \frac{k_p}{m_b}(\Omega_1^2 + {\Omega_3}^2 + {\Omega_2}^2\,\cos(\delta_2)) \\ \frac{\mathbf{I}_b^{yy} - \mathbf{I}_b^{zz}}{\mathbf{I}_b^{xx}}\,\dot{\theta}\,\dot{\psi} + \frac{\mathbf{I}_r^{xx}}{\mathbf{I}_b^{xx}}(\Omega_1 + \Omega_3 - \Omega_2\,\cos(\delta_2))\,\dot{\theta} + \frac{\mathbf{I}_r^{xx}}{\mathbf{I}_b^{xx}}(\Omega_2\,\sin(\delta_2))\,\dot{\psi} - \frac{\sqrt{3}\,k_p\,l}{2\mathbf{I}_b^{xx}}({\Omega_1}^2 - {\Omega_3}^2) \\ \frac{\mathbf{I}_b^{zz} - \mathbf{I}_b^{xx}}{\mathbf{I}_b^{yy}}\dot{\varphi}\,\dot{\psi} + \frac{\mathbf{I}_r^{yy}}{\mathbf{I}_b^{yy}}(\Omega_1 + \Omega_3 - \Omega_2\,\cos(\delta_2))\,\dot{\varphi} + \frac{1}{\mathbf{I}_b^{yy}}(k_h\,\sin(\delta_2) - k_p\,l\,\cos(\delta_2))\,{\Omega_2}^2 + \frac{k_p\,l}{2\mathbf{I}_b^{yy}}({\Omega_1}^2 + {\Omega_3}^2) \\ \frac{\mathbf{I}_b^{xx} - \mathbf{I}_b^{yy}}{\mathbf{I}_b^{zz}}\,\dot{\varphi}\,\dot{\theta} - \frac{\mathbf{I}_r^{zz}}{\mathbf{I}_b^{zz}}\,(\Omega_2\,\sin(\delta_2))\,\dot{\varphi} - \frac{k_h}{\mathbf{I}_b^{zz}}({\Omega_1}^2 + {\Omega_3}^2 - {\Omega_2}^2\cos(\delta_2)) - \frac{k_p\,l}{\mathbf{I}_b^{zz}}({\Omega_2}^2\sin(\delta_2)) \end{bmatrix} \tag{3.49}$$

With the non-linear description of a tricopter style MAV defined, control methodologies similar to that seen in Section (3.4.3) may also be applied. Investigation into appropriate control methodologies for the tricopter style model will be subject of future work.

3.6 Deltacopter Dynamics

The *delta* style UAS, or *deltacopter*, is similar to the tricopter configuration introduced in the previous section. The deltacopter is well established in the literature as an example of MAV configurations that builds on the underacted nature of quadcopter and helicopter style aircraft, to enable more sophisticated pose control [110, 153, 208]. The deltacopter is an over-actuated MAV configuration that allows full control over the aircraft's position and orientation in 3D space. This is achieved by the independent actuation nacelles i through a given tilt angle δ_i as seen in Figure (3.9).

Similar to the tricopter configuration introduced in Section 3.5, the positions of the three nacelles are defined radially to $\boldsymbol{p}_{cg}$ at vertical and horizontal distances $\boldsymbol{l}_1$ and $\boldsymbol{l}_2$ respectively. The exception however is that each rotor can be deflected through angle δ_i by an actuator aligned with each nacelle axis $\boldsymbol{x}_{n_i}$ (see Figure 3.9). The system description again builds on the definitions of $\boldsymbol{f}_{r_i}$ and $\boldsymbol{\tau}_{r_i}$ initially introduced in Section 3.3. The projections of rotor i on nacelle i may be defined by generalising Equation (3.40) to rotor

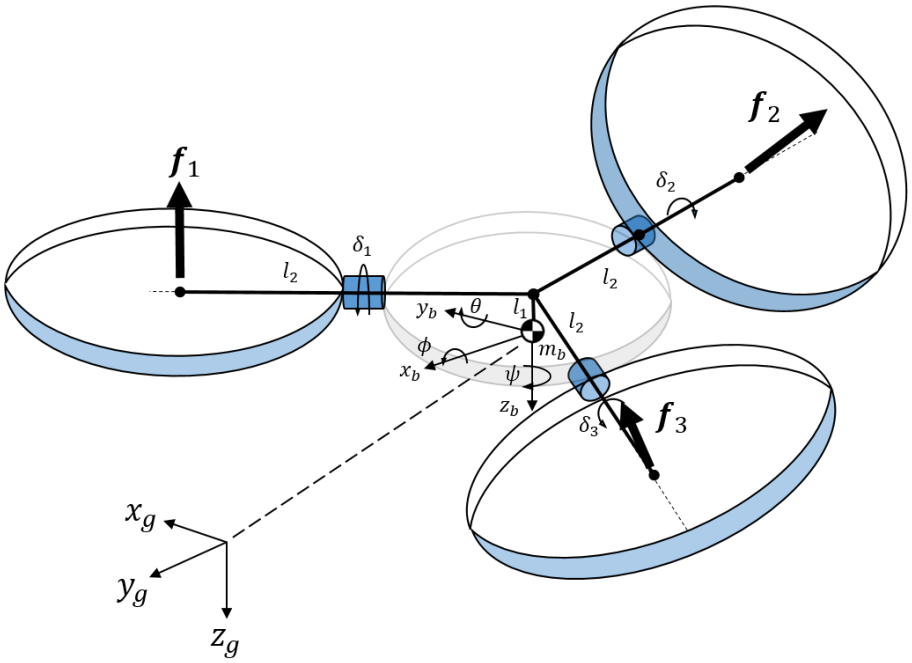


Figure 3.9: *A depiction of a Deltacopter style UAS, with it's three vectored nacelles. The thrust projection from each rotor in the body axes is controlled via a dedicated servo; actuating the nacelle about it's local x-axis (shown in blue).*

i as shown in Equation (3.50):

$$\mathbf{R}_{n_i} = \mathbf{R}_{\delta_i} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\delta_i) & -sin(\delta_i) \\ 0 & sin(\delta_i) & cos(\delta_i) \end{bmatrix} \tag{3.50}$$

The resulting forces acting with the frame of nacelle i must then be expressed in the body axes by invoking the configuration transform $\mathbf{R}_{\lambda_i}$ introduced in Equation (3.16). Given that the nacelles are aligned with the tricopter, the configuration angles λ_i are equivalent.

In these works the symbolic approach has been demonstrated on the conventional three rotor deltacopter [153]. The procedure can be extended trivially to define four rotor deltacopter variants as seen in [180, 208] by introducing a fourth rotor and adopting the configuration vector given in Section 3.4. The expression relating the body axis forces and torques can then be written as the sum of the contributions from each vectoring nacelle

as seen in Equations (3.51) and (3.52).

$$\boldsymbol{f}_{prop} = \begin{bmatrix} f^x_{prop} \\ f^y_{prop} \\ f^z_{prop} \end{bmatrix} = \sum_{i=1}^{3} \left(\mathbf{R}_{\lambda_i} \mathbf{R}_{\delta_i} \boldsymbol{f}_{r_i} \right) \tag{3.51}$$

$$\boldsymbol{\tau}_{prop} = \begin{bmatrix} \tau^x_{prop} \\ \tau^y_{prop} \\ \tau^z_{prop} \end{bmatrix} = \sum_{i=1}^{3} \mathbf{R}_{\lambda_i} \left(\mathbf{R}_{\delta_i} \boldsymbol{\tau}_{r_i} + (\boldsymbol{l}_1 + \boldsymbol{l}_2) \times \mathbf{R}_{\delta_i} \boldsymbol{f}_{r_i} \right) \tag{3.52}$$

The tilt angle transformation $\mathbf{R}_{\delta_i}$ defines the projection of the rotor thrust $\boldsymbol{f}_i$ onto the nacelle frame. The configuration matrix $\mathbf{R}_{\lambda_i}$ given in Equation (3.16) again describes the rotation from each nacelle frame to the body axes. The propulsive characteristics of the deltacopter can therefore be expressed analytically as $\boldsymbol{f}_{prop}$ and $\boldsymbol{\tau}_{prop}$ [153, 180]. The actuation of the rotors through the tilt angle δ_i induces a reaction torque on the airframe similar to vectored nacelle in Section 3.5. Rewriting Equation (3.45) to describe the reaction torque induced by the motion of each rotor yields Equation (3.53):

$$\boldsymbol{\tau}_{gyro} = \begin{bmatrix} \tau^x_{gyro} \\ \tau^y_{gyro} \\ \tau^z_{gyro} \end{bmatrix} = \mathbf{I}_r \sum_{i=1}^{3} \left[\boldsymbol{\omega} \times \left(\mathbf{R}_{\lambda_i} \mathbf{R}_{\delta_i} \begin{bmatrix} 0 \\ 0 \\ \kappa_i \end{bmatrix} \Omega_{r_i} \right) \right] \tag{3.53}$$

With the contributions from each nacelle defined within the body axes, the resulting symbolic vectors $\boldsymbol{\nu}_{prop} = [\boldsymbol{f}_{prop}, \boldsymbol{\tau}_{prop}]^T$ and $\boldsymbol{\nu}_{gyro} = [0, 0, 0, \boldsymbol{\tau}_{gyro}]^T$ now represent the propulsive characteristics of a deltacopter MAV. By introducing the rigid-body fuselage assumptions, the non-linear dynamics defining the body axis accelerations of a deltacopter MAV are then presented as Equation (3.54).

The thrust interactions from the three vectoring nacelles can clearly be seen as a function of their respective deflection angles δ_i in Equation (3.54). The body axis torques are then presented in terms of the contributions from the thrust of the rotor k_p, their aerodynamic resistance k_h and the gyroscopic interaction induced by the deflection of rotor i from it's current instantaneous orientation.

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \ddot{\varphi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \dot{y}\,\dot{\psi} - \dot{z}\,\dot{\theta} - g\,\sin(\theta) - \frac{\sqrt{3}\,k_p}{2\,m_b}(\Omega_1{}^2\,\sin(\delta_1) - \Omega_3{}^2\,\sin(\delta_3)) \\ \dot{z}\,\dot{\varphi} - \dot{x}\,\dot{\psi} + g\,\cos(\theta)\,\sin(\varphi) + \frac{k_p}{2\,m_b}(\Omega_1{}^2\sin(\delta_1) + \Omega_3{}^2\,\sin(\delta_3) - m_b\,\Omega_2{}^2\sin(\delta_2)) \\ \dot{x}\,\dot{\theta} - \dot{y}\,\dot{\varphi} + g\,\cos(\varphi)\,\cos(\theta) - \frac{k_p}{m_b}\left(\Omega_1{}^2\,\cos(\delta_1) + \Omega_2{}^2\,\cos(\delta_2) + \Omega_3{}^2\,\cos(\delta_3)\right) \\ \frac{\mathbf{I}_b^{yy} - \mathbf{I}_b^{zz}}{\mathbf{I}_b^{xx}}\,\dot{\theta}\,\dot{\psi} + \frac{\mathbf{I}_r^{xx}}{\mathbf{I}_b^{xx}}(\Omega_1\,\cos(\delta_1) - \Omega_2\,\cos(\delta_2) + \Omega_3\,\cos(\delta_3))\dot{\theta} + \frac{\mathbf{I}_r^{xx}}{2\,\mathbf{I}_b^{xx}}(\Omega_1\,\sin(\delta_1) + 2\,\Omega_2\,\sin(\delta_2) + \Omega_3\,\sin(\delta_3))\,\dot{\psi} \\ -\frac{\sqrt{3}}{2\,\mathbf{I}_b^{xx}}\left(k_h\left(\sin(\delta_1)\,\Omega_1{}^2 - \sin(\delta_3)\,\Omega_3{}^2\right) + k_p\,l\left(\Omega_1{}^2\,\cos(\delta_1) - \Omega_3{}^2\,\cos(\delta_3)\right)\right) \\ \frac{\mathbf{I}_b^{zz} - \mathbf{I}_b^{xx}}{\mathbf{I}_b^{yy}}\,\dot{\varphi}\dot{\psi} - \frac{\mathbf{I}_r^{yy}}{\mathbf{I}_b^{yy}}(\Omega_1\,\cos(\delta_1) - \Omega_2\,\cos(\delta_2) + \Omega_3\,\cos(\delta_3))\dot{\varphi} + \frac{\sqrt{3}\,\mathbf{I}_r^{yy}}{2\,\mathbf{I}_b^{yy}}\,(\Omega_1\,\sin(\delta_1) - \Omega_3\,\sin(\delta_3))\,\dot{\psi} \\ +\frac{1}{2\,\mathbf{I}_b^{yy}}\left(k_h\,\left(\Omega_1{}^2\,\sin(\delta_1) + 2\,\Omega_2{}^2\,\sin(\delta_2) + \Omega_3{}^2\,\sin(\delta_3)\right) + k_p\,l\,\left(\Omega_1{}^2\,\cos(\delta_1) - 2\,\Omega_2{}^2\,\cos(\delta_2) + \Omega_3{}^2\,\cos(\delta_3)\right)\right) \\ \frac{\mathbf{I}_b^{xx} - \mathbf{I}_b^{yy}}{\mathbf{I}_b^{zz}}\,\dot{\varphi}\dot{\theta} - \frac{\mathbf{I}_r^{zz}}{2\,\mathbf{I}_b^{zz}}(\Omega_1\,\sin(\delta_1) + 2\,\Omega_2\,\sin(\delta_2) + \Omega_3\,\sin(\delta_3))\,\dot{\varphi} + \frac{\sqrt{3}\,\mathbf{I}_r^{zz}}{2\,\mathbf{I}_b^{zz}}(\Omega_3\,\sin(\delta_3) - \Omega_1\,\sin(\delta_1))\,\dot{\theta} \\ +\frac{1}{\mathbf{I}_b^{zz}}\,\left(k_h\,\left(\Omega_2{}^2\,\cos(\delta_2) - \Omega_1{}^2\cos(\delta_1) - \Omega_3{}^2\,\cos(\delta_3)\right) + k_p\,l\,\left(\Omega_1{}^2\,\sin(\delta_1) + \Omega_2{}^2\,\sin(\delta_2) + \Omega_3{}^2\,\sin(\delta_3)\right)\right) \end{bmatrix} \tag{3.54}$$

3.7 Conclusions

In this chapter the key concepts and principles for the dynamic representation of multi-copter style UAS are introduced. A symbolic framework for the analysis and control of MAVs is then proposed. By deriving several popular MAV configurations from first principles, it is shown how parameter-less representations of each systems may be defined for the purposes of control and stability analysis. It is also demonstrated how the approach can be used as a basis for preliminary control design by symbolically defining key model parameters [64].

The presented "aligned" quadcopter model is used as a basis for a preliminary control investigation in which two linear techniques are demonstrated. To contextualise the analysis to a specific MAV system, experimental data gathered from a real F540 quadcopter is introduced. A comparison of the model's performance under the LQR and MPC control approaches is presented; in which control over the vehicles attitude rates is demonstrated in noisy conditions. The MPC controller is shown able to track the desired trajectory more tightly whilst adhering to state and input constraints.

Using the principles presented, the dynamic behaviour of the quadcopter is formulated and examined for the purpose of control analysis. It is then shown how the symbolic definitions of the tricopter and deltacopter MAV configurations can be defined similarly. Here the non-linear body axis representations can be seen presented as opportunities for future work in MAV control design. Previously in Chapter 2 the kinematic representation of objects with arbitrary dynamics is introduced for simulation in OpenMAS [66]. The presented symbolic approach to generating parameter-less dynamic descriptor functions is used to define UAV configurations for the simulation of multi-agent systems in later chapters.

The generalised dynamic description of a tricopter and deltacopter MAV are derived without parameterisation or reduction. While the aerodynamic interactions are neglected in these works, their inclusion would be a clear next step toward synthesising higher fidelity models for the purpose of control design. It is shown how more complex systems such as the deltacopter, with greater degrees of freedom, may be also be modelled effectively through the use of symbolic representation. The principles introduced here are also used to provide context for Chapter 4, where they are extended further in the formation of the novel *Polycopter* MAV configuration presented as part of these works.

Chapter 4

The Polycopter

In this chapter several of the concepts introduced in Chapter 3 are extended in the proposal of a new type of unmanned aerial system. This system, referred to as the *Polycopter*, is able to actively deflect a set of nacelle sub-assemblies positioned radially around it's body through two *degrees of freedom*(DOF). The Polycopteris named according to the versatility gained by combining ideal thrust vector control with active stability augmentation due to the systems agility to control its *centre of gravity*(CG). With each nacelle actuated through 2DOF the system is naturally over-actuated. This presents some interesting opportunities, as well as challenges, as strategies for exploitative control design are investigated.

4.1 Background

The notion of "morphing" aerial systems is a concept that has been gaining momentum in the last decade. This is partly due to the rising interest in reconfigurable and dynamic aerial systems as new applications for unmanned systems are introduced. Traditionally, morphing systems are defined to have two or more operational conditions. Transition between these conditions may be subtle, which may act to "trim" the aircraft. In other cases the transition may be more significant, as a transition is made through discrete propulsive mechanisms. In the literature, there are two distinct groups of morphing systems namely; i) *static* - systems varying a collective thrust by reorientation of a static airframe or ii) *dynamic* - systems that use additional actuators to manipulate local propulsive groups.

Examples of effective pose control using static rotor topologies can be found in [36, 71, 100], where the use of more rotors is often necessary. Hybrid *unmanned aerial systems* (UAS) and *unmanned aerial vehicles* (UAVs) are more commonly adopting principles of

aerial morphism; in systems such as Google's "Project Wing" which aims to combine the VTOL capabilities of conventional multicopters with the operational range of fixed-wing aircraft [236]. In the last decade, there have been several demonstrations of the potential of dynamic morphism. Several of the designs stem for the principles of active augmentation of the rotor disk as presented in Sections 3.5 and 3.6. Examples of complex pose control using the "delta" style configuration can be found in [153, 180, 208]. Here the ability for the aircraft to freely orientate it's fuselage in 3D space is shown by augmenting the attitude of their radial nacelles through one *degree of freedom* (DOF).

More recent derivatives of these systems can be see to deflect the rotor disk through 2DOF to achieve further authority over the pose of the aircraft's airframe. The additional DOF enhances the authority over the fuselage attitude by means of thrust vectoring. This allows the translational mechanics of the system to be decoupled from the pose of the aircraft and allow it to be maintained through various manoeuvres. The use of a dual-axis disk tilting mechanism can be seen in [28, 31, 78, 158, 177, 213, 218, 280]. A more recent example of this concept being extended further, to include design topologies with a greater number of actuated rotor assemblies, can be found in [28]. The concept introduced in [19], is most relevant to the works of this thesis by introducing the concept of vehicle limb transformation. The proposed quadcopter system, is able to augment the radial position of it's nacelles in the body XY plane by way of servo mechanism whilst augmenting the length of each nacelle arm using a prismatic joint. While the investigation is preliminary, the aircraft is shown to augment the position of it's centre of thrust to facilitate thrust compensation in the event of component failure.

Another avenue of investigation in aerial morphism is the concept of multi-linked systems. Through the introduction of systems such as "Dragon" in [274] and the system proposed by Zhao et al. in [275], new territory has seen increasing interest from the aerial robotics community. Here, it is demonstrated that active aerial morphism can be used to achieve unparalleled levels of aerial dexterity and even achieve complex tasks such as grasping. This presents a number of challenges from a control perspective, as it implies an intrinsic understanding of the system's dynamics, which are often highly non-linear, configuration-dependent and time-variant. As a result, often sophisticated control regimes and control allocation strategies are typically required for meaningful control over the dexterous system UAS [274, 275].

4.2 Motivation

The growing body of research into morphing body and unconventional UAS topologies has shown there are numerous opportunities yet to be explored. Systems that are highly manoeuvrable and adaptable have immense potential in scenarios that demand they operate in the presence of component failure, confined spaces or hazardous changing environments. Many of these challenges are typical in applications such as search and rescue, exploration and surveillance operations, just to name a few. A common theme between morphing UAS is versatility. This may apply to the physical capabilities of the UAS; for grasping in [275], enhanced pose control [110,274] or performance optimisation in different conditions [19, 236].

4.2.1 Aerial Morphology

Conventional multicopters, typically consist of several propulsive groups situated in a common plane. While this simplifies the dynamics of the aircraft, systems like the deltacopter and the dual tilt-axis systems introduced in Section 4.1 demonstrate that there are numerous advantages for systems that are able to vector the thrust independently of the main bodies orientation. In the works of Otsuka et al. in [182], it is also suggested that there is a relationship between the relative pitch (or cant) of a rotor disk and the resulting pitch up moment exerted on the body for a given *angle of attack* (AoA). This suggests that there is an inherent advantage for systems with active control over the cant angle of a given rotor disk so that it's AoA can be modified with respect to incoming airflow. This is highly useful for the purposes of "trimming" aircraft to a new flight condition, but also in the precise manipulation of the lift (and drag) generated at a given rotor speed [182].

Not unlike conventional rotary-wing systems, multicopters face numerous challenges due to the "down wash" created by their propulsive groups (see Figure 4.1). Turbulence induced by the momentum exchange from the rotors with the ambient airflow has the potential to create complex and potentially hazardous disruptions to it's flight path and to that of nearby vehicles. In addition, poor management of the airflow can often lead to a distinct loss of stability, power and aerodynamic efficiency of the rotors [241,242,266]. The challenges here are compounded for systems with multiple rotors, within close proximity to the ground or ceiling (in the *ground/ceiling effect*) or to other vehicles, where flow re-circulation becomes an issue [241, 242].

The concept of *active airflow management* has yet to be explored within the field of UAS technology as control authority is often inadequate. This presents a number

Figure 4.1: *The Eurocopter AS350 engaging in a conventional landing flare manoeuvre [92].*

of opportunities for systems that are able to dynamically manipulate their flow field for enhanced stability, disturbance rejection and control in complex scenarios [182]. An example of this type of control is most notably observed in the "flaring" of helicopters as seen Figure 4.1. Here, in order to reduce the approach velocity towards contact with the ground, an aggressive pitch input is used to manipulate the centre of pressure of the propulsion system. As a result, the fuselage is inclined and the pilot is able to symmetrically distribute disturbances below the vehicle as it descends [266].

In light of these challenges, unmanned systems with thrust vectoring or *gimballed* rotor assemblies are favourable in enhancing the control resolution where manoeuvrability is limited. In addition to this, systems with greater authority over the attitude of the fuselage, as well as the centre of pressure, present an opportunity to *design* the down-wash generated by the propulsion system for more efficient and robust trajectory control [241, 242, 266].

4.2.2 Biomimicry

The level of precise control of the aerodynamic forces generated by lifting bodies is widely observed in nature and in the complex anatomy of avians as seen in Figure 4.2. Avians are capable of adjusting the relative attitude and mean chord length of each lifting body in order to vector the associated aerodynamic forces. The geometry of the wings, with some exceptions, is both proportional to mass of the avian and the lifting capacity. The precision however originates from the morphology of their wings and tail as control surfaces in order to generate, or bleed off, excess lift (drag) and modify their approach trajectory

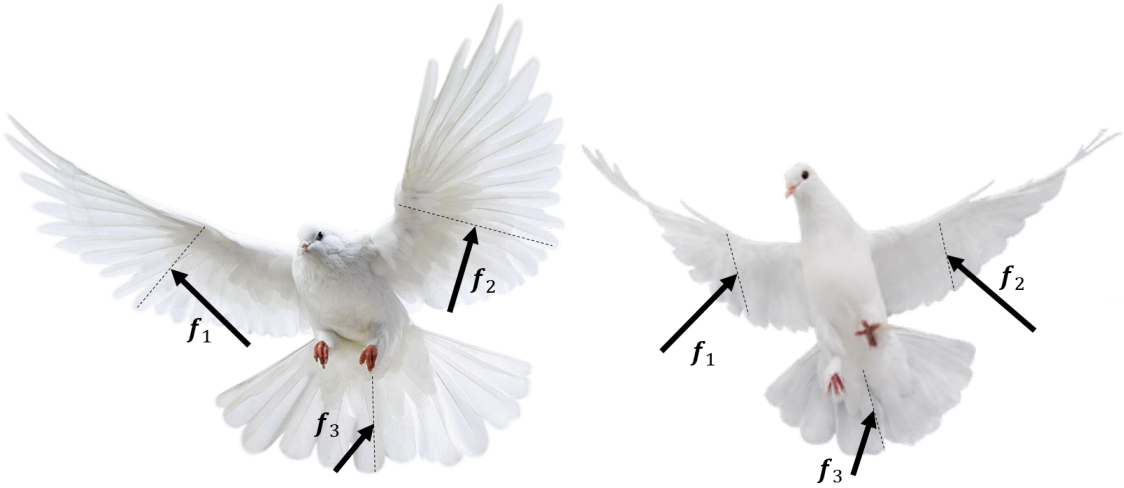


Figure 4.2: *A depiction of a dove's ability to actively adjust the dihedral (anhedral) angle of it's lift (drag) generating surfaces independently of their body. Their precise control (trimming) of their respective aerodynamic moments is achieved by adjusting the effective* angle of attack *(AoA) and chord length of their wings [89, 182, 245].*

and speed. Further information on the aerodynamic properties of avian wings can be found in [239, 245].

This level of dexterity is vital where precise control of the avian's approach angle is necessary such as hunting, perching or landing. Manoeuvring and stability are both critically important for avians across their various kinematic gaits. Their natural ability to adjust the dihedral (anhedral) angles of their wings during flight allows them to transition for stability in gliding conditions, but then also facilitate aggressive turns through acute control over their CG and lifting forces [239, 245].

The potential for robotic systems to be able to better emulate the morphology of avian flight is clear. *micro aerial vehicles* (MAVs) that are able to dynamically adjust their flight characteristics represent a number of opportunities for contribution to the field of aerial robotics. In the following sections a novel morphing MAV configuration is presented referred to as the "Polycopter". This configuration is defined by a series of actuated nacelle assemblies to mimic the versatility of avian control surfaces in complex manoeuvres. The manipulation of these nacelles effectively allows active control over the centre of gravity of the airframe in relation to the centres of lift of it's three radial rotor assemblies.

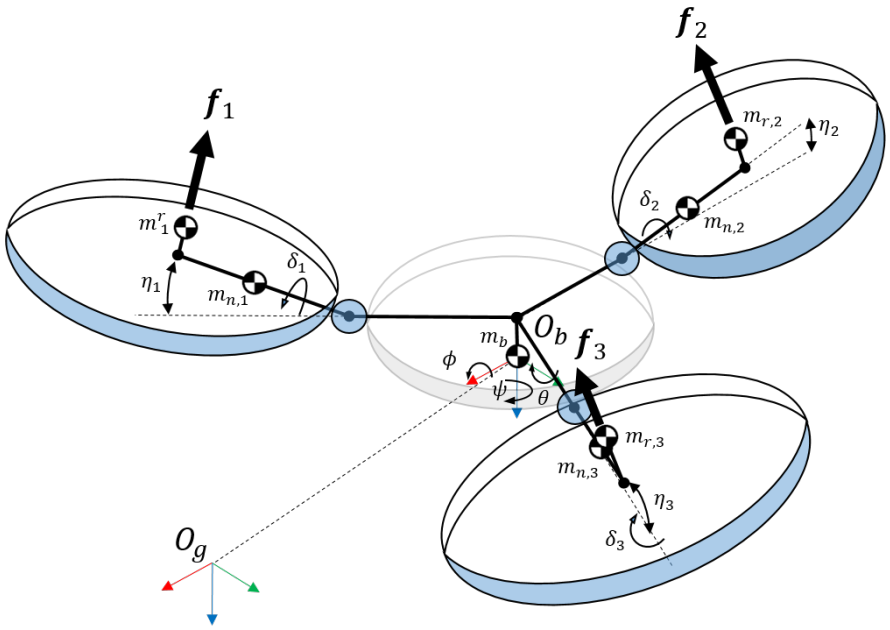


Figure 4.3: *A depiction of the proposed Polycopterconcept as a logical extension of the Tricopter and Deltacopter configurations. The relative pitch (δ_i) and roll (η_i) is actively adjusted using a servo assembly positioned at the base of the nacelle. As a result, the thrust $\boldsymbol{f}_i$ from rotor i is vectored independently of the body orientation.*

4.3 Overview

The name "Polycopter" UAS stems from the versatility gained by combining active control over the system's centre of gravity and complete thrust vector authority. This is achieved by taking advantage of the Polycopter's morphing airframe, which is defined by a central body and a set of actively driven nacelle sub-assemblies. In addition to the tilt angle δ_i, analogous to the deflection angle presented in Sections 3.5 and 3.6, the Polycopteris able to pitch each nacelle assembly through angle η_i. The pose of each nacelle is actively maintained by a 2DOF actuator positioned at the nacelle joint as seen in Figure 4.3.

The fact that the Polycopteris able to define the time variant configuration its nacelles independently provides three principle advantages, namely; i) their instantaneous pose dictates the mass distribution of the airframe and therefore its flight stability characteristics, ii) Their active vectoring relative to the body acts to decouple the translational and rotational forces from the pose of the central body and iii) the nacelle thrust is commanded by a combination of three actuators presenting an opportunity for higher precision control and trimming.

Together these characteristics present a unique opportunity for unparalleled versatility in conventionally difficult scenarios and environments. A classical example of a challenging scenario that is initially introduced in Section 4.2.1 is manoeuvring within the

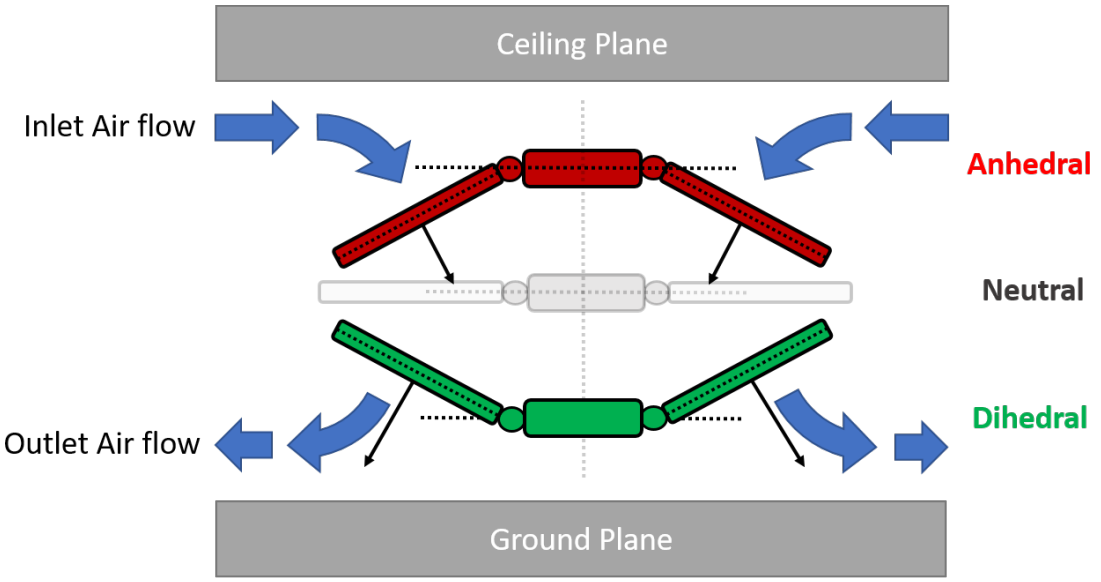


Figure 4.4: *The concept of* active airflow management *used to adjust the flow field around the rotor disks without moving the centre of pressure or fuselage orientation in 3D space. Utilising this premise, the versacopter can reorient its nacelles for aerodynamic optimisation in conventionally complex scenarios such as the ceiling/ground effect.*

ground(ceiling) effect. Due to the static nature of conventional UAS designs, re-circulation is often a factor leading to instability, unreliability and the need for more intelligent control design. It is however shown in Figure 4.4, how the Polycopter's unique range of motion can be used to optimise airflow above, or beneath, the airframe in such conditions without modifying the body pose. This is achieved by optimising the nacelle configuration in accordance to the desired trajectory and some higher-level objective emphasising *air-flow management*, stability or failure tolerance.

The same nacelle motion providing universal authority over the thrust vector from rotor i relative to the body, the Polycopteris able to mimic a number of properties discussed in Section 4.2.1 and 4.2.2 by transitioning between multiple dynamic gaits. Due to the over-actuated nature of the system, numerous opportunities exist for the design of controllers optimising nacelle configurations for various flight performance metrics (such as speed, precision or efficiency) can be devised. These are however beyond the scope of this thesis.

In the follow sections, the dynamics of the Polycoptersystem are investigated under two initial assumptions. A component-based overview of the airframe is presented in Figure 4.5 in which their frames of reference and symbolic parameters are presented. The mass of the central body m_b is assumed positioned at the body axis origin O_b where it's mass distribution is described by $\boldsymbol{I}_b$. The position of the coordinate system O_{n_i} defining nacelle joint i is positioned as vertical and horizontal distances l_1 and l_2 from O_b

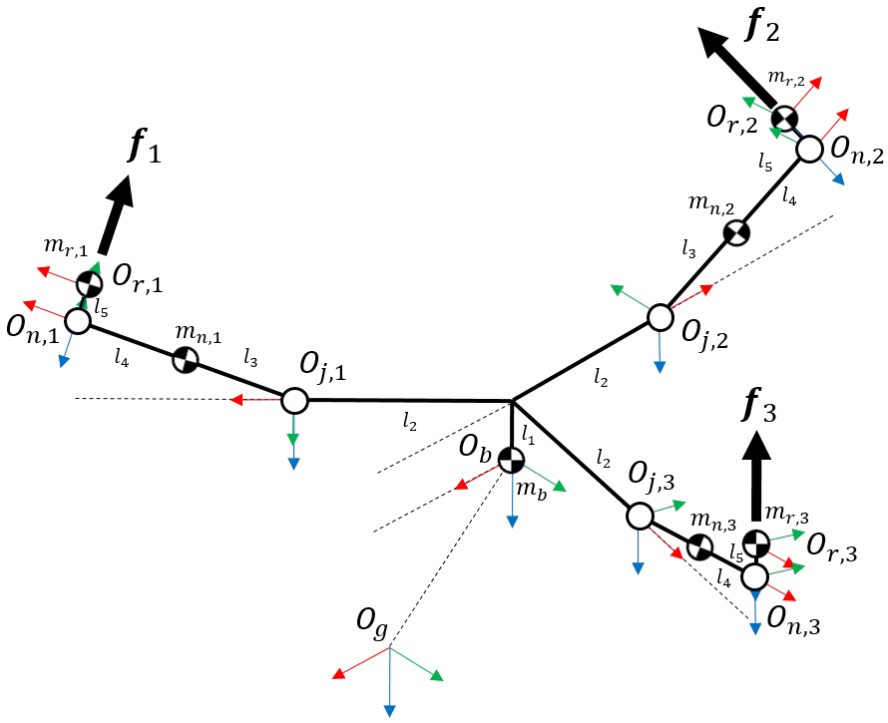


Figure 4.5: *The component frames of reference that defined the PolycopterUAS, in which the orientation or the body O_b, nacelles O_{n_i} and rotors O_{r_i} are described. The geometric positions of the component masses m_b, m_{n_i} and m_{r_i} is shown in relation to the body axis origin O_b.*

respectively. The mass of the nacelle assembly m_n is positioned at horizontal distance l_3 from O_{n_i} with it's mass distribution described by $\boldsymbol{I}_n$. The position of the rotor frame O_{r_i} is at horizontal and vertical distances l_4 and l_5 from the nacelle centre of mass m_n. The mass of rotor m_r is assumed aligned with the origin of O_{r_i}, with inertia tensor $\boldsymbol{I}_r$. The distance l_5 defines the vertical offset of the rotor plane from the nacelle mass due to the electro-mechanical propulsive components introduced in Section 3.3.

4.4 Assuming Small Nacelle Deflections

In this section, the dynamics of the Polycopterare presented for the first time. Due to the morphing nature of the airframe and nacelle assemblies, the inertial interactions of the nacelles present a challenge from a control perspective. Initially, a derivation of the system is proposed that assumes that the perturbations made to the nacelle angles η_i and δ_i are small, such that $\mathbf{I}_b$ can be said to be static and time-invariant. Under this assumption, the dynamics of the vehicle are synonymous with the dual tilt-axis systems discussed in the literature (see Section 4.2). This assumption is examined further in Section 4.5.

The dynamics of the airframe are defined by the force interactions between each of the components as a result of the nacelle deflections and changing airframe geometry.

The gyroscopic effects of perturbing rotor assembly i, are modelled by considering the rotor disks as sources of momentum as presented in Chapter 3. In the following sections, it is shown how by considering the interactions between the various components of the airframe, a simplified dynamic description of the Polycoptermay be formed by examining the forces exerted by rotor i on the body O_b.

4.4.1 Rotor Interactions

In the literature, the assumption is frequently made that the airframe mass is defined by m_b. The mass of the rotational components of each nacelle is assumed to be negligible with respect to the airframe (i.e. $m_r \ll m_b$). Building on the initial rotor model introduced in Section 3.3, the rotational rate of the rotor ω_{r_i} in frame O_{r_i} is defined exclusively about the local $\hat{z}$ axis in Equation (4.1):

$$\boldsymbol{\omega}_{r_i} = \begin{bmatrix} \omega_{r_i}^x \\ \omega_{r_i}^y \\ \omega_{r_i}^z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \kappa_i \end{bmatrix} \Omega_{r_i} \tag{4.1}$$

The aerodynamic forces $\boldsymbol{f}_{r_i}, \boldsymbol{\tau}_{r_i}$ acting within O_{r_i}, induced by the shape of rotor i, are expressed proportional to it's angular rate ω_{r_i} in Equations (4.2) to (4.3).

$$\boldsymbol{f}_{r_i} = \begin{bmatrix} f_{r_i}^x \\ f_{r_i}^y \\ f_{r_i}^z \end{bmatrix} = k_p \boldsymbol{\omega}_{r_i}^2 \tag{4.2}$$

$$\boldsymbol{\tau}_{r_i} = \begin{bmatrix} \tau_{r_i}^x \\ \tau_{r_i}^y \\ \tau_{r_i}^z \end{bmatrix} = k_h \boldsymbol{\omega}_{r_i}^2 \tag{4.3}$$

4.4.2 Nacelle Interactions

The rotor is modelled as a component within nacelle assembly i with reference axes O_{r_i}. For simplicity, it is assumed that the axes of the nacelle O_{n_i} are aligned with the rotor such that $\mathbf{R}_n^r = \mathbf{I}_{3\times 3}$. The angular velocity of the rotor i is then expressed in O_{n_i} by the

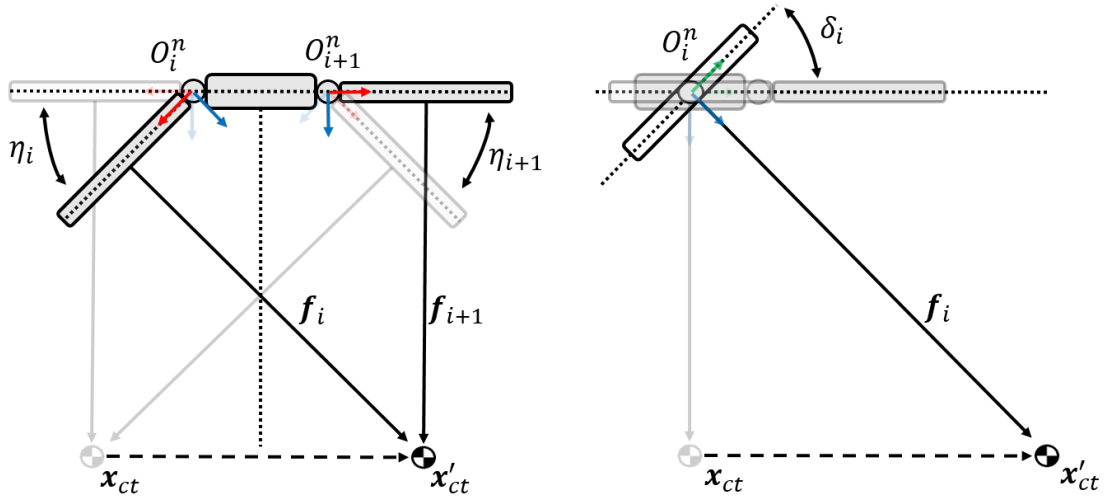


Figure 4.6: *Left) The position of the centre of thrust $\boldsymbol{x}'_{ct}$ as a result of the nacelle deflections η_i and η_j. Right) The movement of the centre of thrust $\boldsymbol{x}_{ct}$ to $\boldsymbol{x}'_{ct}$ as a result of nacelle deflection δ_i . Collectively, these unique motion characteristics allow the emulation of that seen in Figure 4.2.*

transform (4.4).

$$\boldsymbol{\omega}_{n_i} = \begin{bmatrix} \omega_{n_i}^x \\ \omega_{n_i}^y \\ \omega_{n_i}^z \end{bmatrix} = \mathbf{R}_{n_i}^r \boldsymbol{\omega}_{j_i} \tag{4.4}$$

The forces $\boldsymbol{f}_{n_i}$ and torques $\boldsymbol{\tau}_{n_i}$ acting within O_{n_i} as a result of the propulsive forces are then written as Equations (4.5) and (4.6):

$$\boldsymbol{f}_{n_i} = \begin{bmatrix} f_{n_i}^x \\ f_{n_i}^y \\ f_{n_i}^z \end{bmatrix} = \mathbf{R}_{n_i}^r \boldsymbol{f}_{r_i} \tag{4.5}$$

$$\boldsymbol{\tau}_{n_i} = \begin{bmatrix} \tau_{n_i}^x \\ \tau_{n_i}^y \\ \tau_{n_i}^z \end{bmatrix} = \mathbf{R}_{n_i}^r \boldsymbol{\tau}_{r_i} + (l_3 + l_4 + l_5) \times \boldsymbol{f}_{n_i} \tag{4.6}$$

Here $\boldsymbol{l}_3 = [l_3, 0, 0]^T$, $\boldsymbol{l}_4 = [l_4, 0, 0]^T$ are geometric parameters describing the horizontal length of nacelle i and $\boldsymbol{l}_5 = [0, 0, -l_5]^T$ defines the vertical offset of frame O_{r_i} from O_{n_i}.

4.4.3 Joint Interactions

The motion of nacelle i is measured relative to frame O_{j_i}; assigned to define the actuation axes of servo i. The servo positioned at joint i actuates nacelle i through deflections η_i and δ_i in order to modify the projection of $\boldsymbol{f}_i$ as seen in Figure 4.6. The corresponding

transformation of the nacelle $\mathbf{R}^n_{j_i}$ as a result of the deflection angles, assuming that the actuators are fast-acting, is expressed as Equation (4.7).

$$\mathbf{R}^n_{j_i} = \mathbf{R}^y(\eta_i)\mathbf{R}^x(\delta_i) = \begin{bmatrix} \cos(\eta_i) & 0 & \sin(\eta_i) \\ 0 & 1 & 0 \\ -\sin(\eta_i) & 0 & \cos(\eta_i) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\delta_i) & -\sin(\delta_i) \\ 0 & \sin(\delta_i) & \cos(\delta_i) \end{bmatrix} \tag{4.7}$$

Given the instantaneous position of nacelle i, the rotational rate of the rotor is defined in the axes of the nacelle joint O_{j_i} in (4.8):

$$\boldsymbol{\omega}_{j_i} = \mathbf{R}^n_{j_i}\boldsymbol{\omega}_{n_i} \tag{4.8}$$

The resultant forces $\boldsymbol{f}_{j_i}$ and torques $\boldsymbol{\tau}_{j_i}$ acting at nacelle joint O_{j_i} are defined in Equations (4.9) and (4.10).

$$\boldsymbol{f}_{j_i} = \mathbf{R}^n_j \boldsymbol{f}_{n_i} \tag{4.9}$$

$$\boldsymbol{\tau}_{j_i} = \mathbf{R}^n_j \boldsymbol{\tau}_{n_i} \tag{4.10}$$

As seen in Figure 4.6, O_{n_i} and O_{j_i} are co-situated at the joint and so $\boldsymbol{\tau}_{j_i}$ is a direct transform of $\boldsymbol{\tau}_{n_i}$ from the nacelle O_{n_i}.

4.4.4 Body Interactions

Finally, before the interaction of each jointed nacelle may be considered, their contributions must be transformed to the body axes O_b. In Chapter 3, a configuration angle λ_i is used to describe the radial position of joint i in relation to the body axes O_b. This transformation is expressed as a vector rotation around the body z-axis as seen in Equation (4.11):

$$\mathbf{R}^j_{b_i} = \mathbf{R}^z(\lambda_i) = \begin{bmatrix} \cos(\lambda_i) & -\sin(\lambda_i) & 0 \\ \sin(\lambda_i) & \cos(\lambda_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.11}$$

The position of each nacelle joint is assumed defined by the constant configuration angle λ_i; describing it's angular position relative to the body x axis. An example of a morphing MAV where the lateral position of nacelle i is actuated can be found in [19]. Similar to the systems introduced in Section 3.5 and 3.6, radial position of nacelle joints $1:3$ are constant and are equidistributed about the body z axis as $\lambda_1 = \frac{1}{3}\pi\, rad, \lambda_2 = \pi\, rad$ and

$\lambda_3 = \frac{5}{3}\pi\, rad.$

With the constant configuration transform $\mathbf{R}_{b_i}^{j}$ defined, the net forces and torques due to the propulsive components of the nacelles may be written as Equations (4.12) and (4.13):

$$\boldsymbol{f}_{prop} = \sum_{i=1}^{3} \left(\mathbf{R}_{b_i}^{j}\, \boldsymbol{f}_i^{j} \right) \tag{4.12}$$

$$\boldsymbol{\tau}_{prop} = \sum_{i=1}^{3} \mathbf{R}_{b_i}^{j} \left(\boldsymbol{\tau}_i^{j} + (\boldsymbol{l}_1 + \boldsymbol{l}_2) \times \boldsymbol{f}_i^{j} \right) \tag{4.13}$$

Here $\boldsymbol{l}_1 = [0, 0, -l_1]^T$ and $\boldsymbol{l}_2 = [l_2, 0, 0]^T$ define the vertical and horizontal offset of O_{n_i} from O_b respectively; where O_b is assumed aligned with the centre of mass m_b of the central body. The torque induced on the airframe as a result of modifying the orientation of nacelles 1 to 3 is therefore also written as Equation (4.14):

$$\boldsymbol{\tau}_{gyro} = \mathbf{I}_r \sum_{i=1}^{3} \left(\boldsymbol{\omega}_b \times (\mathbf{R}_{b_i}^{j}\, \boldsymbol{\omega}_{n_i}) \right) \tag{4.14}$$

The resulting generalised forces, assuming small nacelle deflections may then be expressed as the profile vectors $\boldsymbol{\nu}_{prop} = [\boldsymbol{f}_{prop}^{T}, \boldsymbol{\tau}^{T}]^T$ and $\boldsymbol{\nu}_{gyro} = [0, 0, 0, \boldsymbol{\tau}_{gyro}^{T}]^T$ as demonstrated in Chapter 3. The body of the fuselage is assumed to behave as a rigid-body under the influence of $\boldsymbol{\nu}_{prop}$ and $\boldsymbol{\nu}_{gyro}$. Invoking Equation (3.10) here in Equation (4.15), the net influences acting on the body are defined in terms of the symbolic sources:

$$\boldsymbol{\nu} = \boldsymbol{\nu}_{prop} + \boldsymbol{\nu}_{gyro} + \tilde{\mathbf{R}}_{b}^{g} \left(\boldsymbol{\nu}_{aero} + \boldsymbol{\nu}_{grav} \right) \tag{4.15}$$

In this preliminary investigation, the forces due to the aerodynamics of the airframe and rotors are neglected under the assumption that the vehicle is operating near to a hover state. If it is assumed that the nacelle deflections are small(η_i and δ_i are near zero) then $\mathbf{I}_b$ may be considered constant. The resulting equations of motion of the Polycoptersystem are then defined by the interaction of the forces within O_b as seen in Equations (4.16) and (4.17):

$$\begin{bmatrix} m_b \mathbf{I}_{3x3} & 0_{3x3} \\ 0_{3x3} & \mathbf{I}_b \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{v}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} + \begin{bmatrix} m_b S(\boldsymbol{\omega}) & 0_{3\times3} \\ 0_{3\times3} & -S(\mathbf{I}_b \boldsymbol{\omega}) \end{bmatrix} \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{\omega} \end{bmatrix} = \boldsymbol{\nu} \tag{4.16}$$

Which may also be expressed as:

$$\ddot{\boldsymbol{x}} = \mathbf{M}^{-1}\left(\boldsymbol{\nu} - \mathbf{C}\dot{\boldsymbol{x}}\right) \tag{4.17}$$

The accelerations of the body $\ddot{\boldsymbol{x}}$ can then be seen defined in terms of the Coriolis and centripetal forces $\mathbf{C}$, inertial coefficients $\mathbf{M}$ and symbolic vector $\boldsymbol{\nu}$ profiling the propulsion system of the Polycopter. The motion of the airframe is defined within the body axes O_b. If it is assumed, that the axes of rotation of airframe are aligned with Euler axes of rotation the relation can be defined $\boldsymbol{v} = [u, v, w]^T = [\dot{x}, \dot{y}, \dot{z}]^T$ and $\boldsymbol{\omega} = [\dot{\phi}, \dot{\theta}, \dot{\psi}]^T$. We can then define non-linear behaviour of the Polycoptersystem to be Equation (4.18):

$$\ddot{\boldsymbol{x}} = d(\dot{\boldsymbol{x}}, \boldsymbol{x}, \boldsymbol{u}) \tag{4.18}$$

Here, the system state is defined to be the kinematic states of the airframe $\boldsymbol{x} = [x, y, z, \phi, \theta, \psi]$ with inputs $\boldsymbol{u} = [\delta_1, \delta_2, \delta_3, \eta_1, \eta_2, \eta_3, \Omega_1, \Omega_2, \Omega_3]^T$ defining the sequence of nacelle deflections $\eta_{1:3}$, $\delta_{1:3}$ and rotor speeds $\Omega_{1:3}$.

4.5 Assuming Massive Nacelles

In Section 4.4 the Polycopteris initially introduced under the assumption of small nacelle deflection angles. While this assumption is sufficient in describing the inertial interactions of the assembly in a near-hover scenario, it becomes unrepresentative as the nacelle deflections move away from *straight and level* (S&L) (i.e $\eta_i, \delta_i > 5°$) and the effects of greater nacelle deflections must be considered.

To consider the changing inertia of the airframe, as a result of the nacelle deflections, is to represent the true morphing nature of the Polycopter. Each nacelle assembly is assumed to be identical as discussed in Section 4.3, whose pose relative to the central body is time-variant and actuated by a torque applied at joint O_{j_i} as seen in Figure 4.6. The representation of the inertial interaction of the vectoring nacelles is important to the modelling of the Polycopter, as it is here that many novel behaviours are introduced (see Section 4.2).

In the following sections the author presents a parallel formulation of the Polycopterdynamics, generalised to include the inertial interactions of the actuated nacelle assemblies. This is presented in the form of a component analysis, where a *recursive newton-euler* (RNE) method is introduced to procedurally describe the interactions between each airframe component.

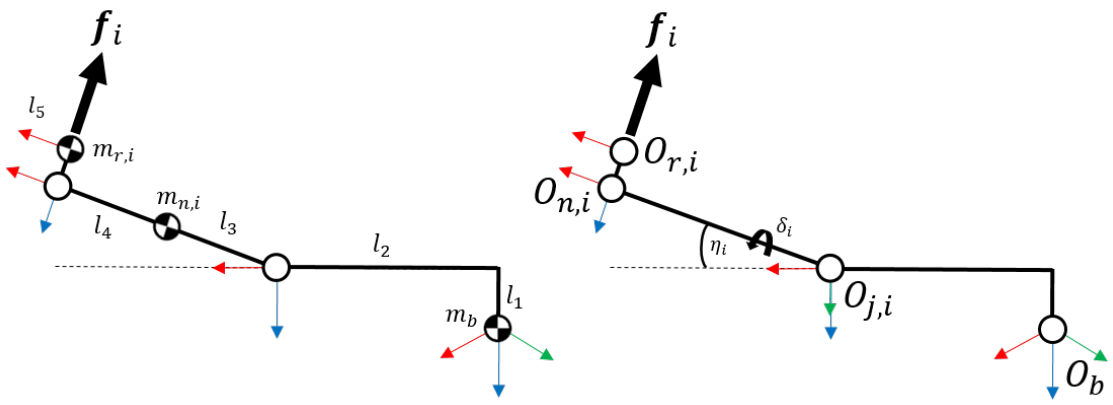


Figure 4.7: *left) The geometric positions of the component masses m_b, m_{n_i} and m_{r_i} are shown. right) A depiction of the Polycoptercoordinate axes where the orientation of O_b, O_{j_i}, O_{n_i} and O_{r_i} are defined as the component frames of the body, joint i nacelle i and rotor i respectively.*

4.5.1 Recursive Newton-Euler Method

The Polycopterairframe is composed of a central body and three actuated nacelle sub-assemblies; through which the propulsive forces are vectored. Each sub-assembly consists of i) a nacelle structure, with mass m_{n_i} and inertia $\mathbf{I}_{n_i}$ ii) the static electro-mechanical components such as the speed controller and brush-less motor stator and iii) the actuated rotor assembly O_{r_i} with mass m_r. The series of transformations defining the relationship between the body frame O_b and the rotor frame O_{r_i} are described in Figure 4.7 with reference to their inertial and geometric parameters of the 15DOF system.

It is assumed that servo mechanism i actuates the pose of nacelle i as seen in Figures 4.6 and 4.7 by applying a torque $\tau_{n_i} = [\tau^x, \tau^y, 0]^T$ about joint axes O_{j_i}. Each nacelle dynamically interacts with the central body of the fuselage through joint i and contributes to acceleration of the airframe.

This contribution is calculated within O_b by considering each nacelle as a serial-link system or *kinematic chain*. The overall expression for dynamic interactions of the nacelles (and their associated propulsive groups) is written as their sum in Equation (4.19) [99,222].

$$\boldsymbol{\nu} = [\boldsymbol{f}_b, \boldsymbol{\tau}_b, \boldsymbol{\tau}_n^x, \boldsymbol{\tau}_n^y, \boldsymbol{\tau}_r]^T = \boldsymbol{\nu}_b + \boldsymbol{\nu}_{n_1} + \boldsymbol{\nu}_{n_2} + \boldsymbol{\nu}_{n_3} + \boldsymbol{\nu}_{ext} \tag{4.19}$$

Here, $\boldsymbol{\nu} \in \mathbb{R}^{15\times 1}$ defines the symbolic forces and torques acting with respect to each of the system's degree of freedom. Here $\boldsymbol{\tau}_b = [\tau_b^x, \tau_b^y, \tau_b^z]^T$ and $\boldsymbol{f}_b = [f_b^x, f_b^y, f_b^z]^T$ define the net body axis torques and forces respectively. The contributions from each sub-assembly, being the body and nacelles $n_i \in n_1, ..n_3$ are then defined as $\boldsymbol{\nu}_b$ and $\boldsymbol{\nu}_{n_i}$ respectively. The vectors defining the torques acting about the x and y axes of the nacelle joint are defined

Property	Symbol	Body	Joint	Nacelle	Rotor
Mass (kg)	m	m_b	0	m_n	m_r
Inertia (kgm^2)	$\mathbf{I}$	$\mathbf{I}_b$	$\mathbf{I}_{3x3}$	$\mathbf{I}_n$	$\mathbf{I}_r$
Rotation (rad)	$\mathbf{R}$	$\mathbf{I}_{3\times3}$	$\mathbf{R}_j^b$	$\mathbf{R}_n^j$	$\mathbf{R}_r^n$
Joint offset (m)	$\boldsymbol{r}_j^i$	$[0,0,0]^T$	$\mathbf{R}_{\lambda_i}(\boldsymbol{l}_1+\boldsymbol{l}_2)$	$\boldsymbol{l}_3+\boldsymbol{l}_4$	$\boldsymbol{l}_5$
CG offset (m)	$\boldsymbol{r}_c^i$	$[0,0,0]^T$	$[0,0,0]^T$	$-\boldsymbol{l}_4$	$[0,0,0]^T$

Table 4.1: *The geometric and inertial constant properties of the Polycopterassociated with each component's coordinate frame. Here $\boldsymbol{r}_j^i$ defines the vector separation between frame i and j respectively.*

as $\boldsymbol{\tau}_n^x = [\tau_{n_1}^x, \tau_{n_2}^x, \tau_{n_3}^x]^T$ and $\boldsymbol{\tau}_n^y = [\tau_{n_1}^y, \tau_{n_2}^y, \tau_{n_3}^y]^T$ respectively. The torque acting within the frame of the rotor $O_{r,i}$ are defined in the vector $\boldsymbol{\tau}_r = [\tau_{r_1}, \tau_{r_2}, \tau_{r_3}]^T$. Finally, $\boldsymbol{\nu}_{ext} \in \mathbb{R}^{15x1}$ is used to characterise external sources of disturbance or noise.

The *recursive Newton-Euler* (RNE) method, traditionally applied in the context of serial manipulators, is introduced here to sequentially define the kinematic and dynamic parameters of the nacelle assemblies. Each component is represented by a component frame (see Figure 4.5); in which the geometry and inertial parameters of that component are defined. These parameters are explicitly stated in Table 4.1.

4.5.2 Joint Interactions

To begin defining a description of the Polycopterairframe, the problem is posed from the perspective of the body O_b. Previously, the influences on the acceleration of the airframe have been attributed to propulsive and gyroscopic sources; as $\boldsymbol{\nu}_{prop}$ and $\boldsymbol{\nu}_{gyro}$ respectively (see Section 4.4.4). Under the RNE convention however, these influences may be considered simultaneously in the symbolic representation of the force vector of nacelle i as $\boldsymbol{\nu}_{n_i}$.

The position of nacelle joint O_{j_i} is defined relatively to O_b and is assumed to be fixed. While the mass of the joint is assumed to be considered within m_b, the kinematic definition of the joint as a result of the motion of the body must therefore be defined.

The transformation between O_b and O_{j_i} as $\mathbf{R}^b_{j_i}$ is written in Equation (4.20):

$$\mathbf{R}^b_{j_i} = \mathbf{R}^z(\lambda_i) = \begin{bmatrix} \cos(\lambda_i) & -\sin(\lambda_i) & 0 \\ \sin(\lambda_i) & \cos(\lambda_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.20}$$

$$\boldsymbol{\omega}_{j_i} = \mathbf{R}^b_{j_i}\boldsymbol{\omega}_b \tag{4.21}$$

$$\dot{\boldsymbol{\omega}}_{j_i} = \mathbf{R}^b_{j_i}\dot{\boldsymbol{\omega}}_b \tag{4.22}$$

$$\dot{\boldsymbol{p}}_{j_i} = \mathbf{R}^b_{j_i}\dot{\boldsymbol{p}}_b + \boldsymbol{\omega}_{j_i} \times \boldsymbol{r}^b_j \tag{4.23}$$

$$\ddot{\boldsymbol{p}}^c_{j_i} = \mathbf{R}^b_{j_i}\ddot{\boldsymbol{p}}_b + \boldsymbol{\omega}_{j_i} \times \boldsymbol{r}^j_c + \boldsymbol{\omega}_{j_i} \times (\boldsymbol{\omega}_{j_i} \times \boldsymbol{r}^j_c) \tag{4.24}$$

Here $\boldsymbol{\omega}_{j_i}$ and $\dot{\boldsymbol{\omega}}_{j_i}$ define the angular velocity and acceleration of joint O_{j_i}, $\dot{\boldsymbol{p}}_{j_i}$ and $\ddot{\boldsymbol{p}}_{j_i}$ are it's linear velocity and acceleration respectively. The separation between between the body and joint is also defined be the vector $\boldsymbol{r}^b_j$. With the kinematics of the joint O_{j_i} in relation to the central body O_b defined, the kinematic parameters of O_{j_i} are related to the torques and forces enacted at O_b by the fixed transform:

$$\boldsymbol{f}_b = \mathbf{R}^{j_i}_b \boldsymbol{f}_{j_i} \tag{4.25}$$

$$\boldsymbol{\tau}_b = -\boldsymbol{f}_b \times \boldsymbol{r}^j_b + \mathbf{R}^{j_i}_b \boldsymbol{\tau}_{j_i} \tag{4.26}$$

Here, the resultant torque and forces acting on O_b are defined in vectors $\boldsymbol{f}_b$ and $\boldsymbol{\tau}_b$ respectively; in terms of the fuselage geometric separation of joint O_{j_i} from O_b as $\boldsymbol{r}^b_j$.

4.5.3 Nacelle Interactions

With the kinematics of joint i defined, their relationship with the nacelle kinematics can be similarly defined. The nacelle is modelled as a rigid-body with mass m_n and inertia $\mathbf{I}_n$; connected with the body of the Polycopterat O_{j_i}. Given the nature of the 2DOF servo mechanism, the transformation of O_{n_i} relative to O_{j_i} may be written in terms of the current nacelle deflection parameters δ_i and η_i as seen in Equation (4.27):

$$\begin{aligned} \mathbf{R}^{j_i}_{n_i} &= \mathbf{R}^y(\eta_i)\mathbf{R}^x(\delta_i) \\ &= \begin{bmatrix} \cos(\eta_i) & 0 & \sin(\eta_i) \\ 0 & 1 & 0 \\ -\sin(\eta_i) & 0 & \cos(\eta_i) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\delta_i) & -\sin(\delta_i) \\ 0 & \sin(\delta_i) & \cos(\delta_i) \end{bmatrix} \end{aligned} \tag{4.27}$$

Given the new orientation of the nacelle $\mathbf{R}_{n_i}^{j_i}$, the definition of the linear and angular axis rates of O_{n_i} and mass m_n may be defined in Equations (4.28)-(4.32).

$$\boldsymbol{\omega}_{n_i} = \mathbf{R}_{n_i}^{j_i} \left(\boldsymbol{\omega}_{j_i} + \begin{bmatrix} \dot{\delta}_i \\ \dot{\eta}_i \\ 0 \end{bmatrix} \right) \tag{4.28}$$

$$\dot{\boldsymbol{\omega}}_{n_i} = \mathbf{R}_{n_i}^{j_i} \left(\dot{\boldsymbol{\omega}}_{j_i} + \begin{bmatrix} \ddot{\delta}_i \\ \ddot{\eta}_i \\ 0 \end{bmatrix} + \boldsymbol{\omega}_{n_i} \times \begin{bmatrix} \dot{\delta}_i \\ \dot{\eta}_i \\ 0 \end{bmatrix} \right) \tag{4.29}$$

$$\dot{\boldsymbol{p}}_{n_i} = \mathbf{R}_{n_i}^{j_i} \dot{\boldsymbol{p}}_{j_i} + \boldsymbol{\omega}_{n_i} \times \boldsymbol{r}_n^j \tag{4.30}$$

$$\ddot{\boldsymbol{p}}_{n_i} = \mathbf{R}_{n_i}^{j_i} \ddot{\boldsymbol{p}}_{j_i} + \boldsymbol{\omega}_{n_i} \times \boldsymbol{r}_n^j + \boldsymbol{\omega}_{n_i} \times (\boldsymbol{\omega}_{n_i} \times \boldsymbol{r}_n^j) \tag{4.31}$$

$$\ddot{\boldsymbol{p}}_{n_i}^c = \ddot{\boldsymbol{p}}_{n_i} + \dot{\boldsymbol{\omega}}_{n_i} \times \boldsymbol{r}_c^n + \boldsymbol{\omega}_{n_i} \times (\boldsymbol{\omega}_{n_i} \times \boldsymbol{r}_c^n) \tag{4.32}$$

Here $\boldsymbol{\omega}_{n_i}$ and $\dot{\boldsymbol{\omega}}_{n_i}$ define the angular velocity and acceleration of O_{n_i}; $\dot{\boldsymbol{p}}_{n_i}$ and $\ddot{\boldsymbol{p}}_{n_i}$ it's linear velocity and acceleration; and $\ddot{\boldsymbol{p}}_{n_i}^c$ the linear acceleration of m_n. The vector $\boldsymbol{r}_n^j$ defines the separation between O_{n_i} and O_{j_i}; $\boldsymbol{r}_c^j$ is the separation between O_{n_i} and the position of m_n.

$$\boldsymbol{f}_{j_i} = \mathbf{R}_{j_i}^{n_i} \boldsymbol{f}_n + m_j \ddot{\boldsymbol{p}}_c^n \tag{4.33}$$

$$\begin{aligned} \boldsymbol{\tau}_{j_i} = & -\boldsymbol{f}_{j_i} \times (\boldsymbol{r}_j^n + \boldsymbol{r}_c^n) \\ & + \mathbf{R}_{j_i}^{n_i} \boldsymbol{\tau}_{n_i} \\ & + (\mathbf{R}_{j_i}^{n_i} \boldsymbol{f}_{n_i}) \times \boldsymbol{r}_c^n \end{aligned} \tag{4.34}$$

Here, the forces and torques acting at the joint are defined as $\boldsymbol{f}_{j_i}$ and $\boldsymbol{\tau}_{j_i}$ respectively. The resulting influences are then due to the torques $\boldsymbol{\tau}_{n_i}$ and force $\boldsymbol{f}_{n_i}$ transmitted from nacelle i. The joint is assumed statically mounted to the body of the Polycopterso that it's mass is considered within m_b and $\mathbf{I}_b$. As O_{n_i} and O_{j_i} are co-situated $\boldsymbol{r}_n^j \equiv [0,0,0]^T$ resulting in Expression (4.34) [99, 222].

4.5.4 Rotor Interactions

Finally, rotor i is modelled as a tertiary rigid-body within the nacelle assembly. The propulsive force generated by the rotor is determined by the rate of the rotation of rigid-rotor i, which is proportional to the torque exerted on rotor i by the nacelle motor. The rotation of O_{r_i} is parameterised by γ_i, with it's velocity and acceleration defined as $\dot{\gamma}_i$ and $\ddot{\gamma}_i$ respectively. The kinematic relationship between frame O_{r_i} from O_{n_i} is defined by

Equations (4.35)-(4.40):

$$\mathbf{R}_{r_i}^{n_i} = \begin{bmatrix} \cos(\gamma_i) & -\sin(\gamma_i) & 0 \\ \sin(\gamma_i) & \cos(\gamma_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.35}$$

$$\boldsymbol{\omega}_{r_i} = \mathbf{R}_{r_i}^{n_i}(\boldsymbol{\omega}_{n_i} + \dot{\gamma}_i \hat{\boldsymbol{z}}_i) \tag{4.36}$$

$$\dot{\boldsymbol{\omega}}_{r_i} = \mathbf{R}_{r_i}^{n_i}(\dot{\boldsymbol{\omega}}_{n_i} + \ddot{\gamma}_i \hat{\boldsymbol{z}}_i + (\boldsymbol{\omega}_{n_i} \times \dot{\gamma}_{r_i})) \tag{4.37}$$

$$\dot{\boldsymbol{p}}_{r_i} = \mathbf{R}_{r_i}^{n_i} \dot{\boldsymbol{p}}_{n_i} + \boldsymbol{\omega}_{r_i} \times \boldsymbol{r}_r^n \tag{4.38}$$

$$\ddot{\boldsymbol{p}}_{r_i} = \mathbf{R}_{r_i}^{n_i} \ddot{\boldsymbol{p}}_{n_i} + \boldsymbol{\omega}_{r_i} \times \boldsymbol{r}_n^r + \boldsymbol{\omega}_{r_i} \times (\boldsymbol{\omega}_{r_i} \times \boldsymbol{r}_n^r) \tag{4.39}$$

$$\ddot{\boldsymbol{p}}_{r_i}^c = \ddot{\boldsymbol{p}}_{r_i} + \dot{\boldsymbol{\omega}}_{r_i} \times \boldsymbol{r}_r^n + \boldsymbol{\omega}_{r_i} \times (\boldsymbol{\omega}_{r_i} \times \boldsymbol{r}_c^r) \tag{4.40}$$

Here $\boldsymbol{\omega}_{r_i}$ and $\dot{\boldsymbol{\omega}}_{r_i}$ define the angular velocity and acceleration of O_{r_i} respectively; $\dot{\boldsymbol{p}}_{r_i}$ and $\ddot{p}_{r_i}$ it's linear velocity and acceleration; and $\ddot{\boldsymbol{p}}_{r_i}^c$ the linear acceleration of the centre of mass. The vector $\boldsymbol{r}_r^n$ defines the position of O_{r_i} in O_{n_i} and $\boldsymbol{r}_c^r$ defines the position of m_r in O_{r_i}. The unit vector $\hat{\boldsymbol{z}}_{r_i}$ is then the rotor actuation axis defined in O_{r_i} [222].

The forces $\boldsymbol{f}_{n_i}$ and torques $\boldsymbol{\tau}_{n_i}$ acting within frame O_{n_i} can be expressed in Equations (4.41) and (4.42). Here the influences of rotor i ($\boldsymbol{f}_{r_i}$ and $\boldsymbol{\tau}_{r_i}$) are expressed within O_{n_i} via the transformation $\mathbf{R}_{n_i}^{r_i}$.

$$\boldsymbol{f}_{n_i} = \mathbf{R}_{n_i}^{r_i} \boldsymbol{f}_{r_i} + m_n \ddot{\boldsymbol{p}}_c^n \tag{4.41}$$

$$\begin{aligned} \boldsymbol{\tau}_{n_i} = & -\boldsymbol{f}_{n_i} \times (\boldsymbol{r}_n^j + \boldsymbol{r}_c^n)) \\ & + \mathbf{R}_{n_i}^{r_i} \boldsymbol{\tau}_{r_i} \\ & + (\mathbf{R}_{n_i}^{r_i} \boldsymbol{f}_{r_i}) \times \boldsymbol{r}_c^r \\ & + \mathbf{I}_n \dot{\boldsymbol{\omega}}_{n_i} + \boldsymbol{\omega}_{n_i} \times (\mathbf{I}_n \boldsymbol{\omega}_{n_i}) \end{aligned} \tag{4.42}$$

The position of rotor i within O_{n_i} is defined as $\boldsymbol{r}_r^n$. $\boldsymbol{r}_c^n$ defines the CG position of nacelle i within O_{n_i}. $\mathbf{I}_n$ and m_n are the nacelle inertial parameters [99, 222].

4.5.5 Rotor Dynamics

Similar to the rotor model introduced in Chapter 3, rotor i is assumed to rotate about it's local z axis, with it's mass m_r positioned at origin O_{r_i} with symmetrical mass distribution described by $\mathbf{I}_r$. In addition to the inertial and Coriolis forces induced by the motion of O_{r_i}, the aerodynamic forces produced by the form of rotor i are shown proportional to it's angular rate $\boldsymbol{\omega}_{r_i} = [0, 0, \dot{\gamma}_i]$ [19, 64, 175]. Incorporating the rotor assumptions introduced

previously, the force and corresponding reaction (hub) torque may be written proportional to the rotational rate of the rotor body in Equations (4.43) and (4.44):

$$\boldsymbol{f}_{r_i} = \begin{bmatrix} f_{r_i}^x \\ f_{r_i}^y \\ f_{r_i}^z \end{bmatrix} = m_r \ddot{\boldsymbol{p}}_{c_i}^r + \cdot k_p \boldsymbol{\omega}_{r_i}^2 \tag{4.43}$$

$$\boldsymbol{\tau}_{r_i} = \begin{bmatrix} \tau_{r_i}^x \\ \tau_{r_i}^y \\ \tau_{r_i}^z \end{bmatrix} = -\boldsymbol{f}_{r_i} \times \boldsymbol{r}_r^n + \mathbf{I}_r \dot{\boldsymbol{\omega}}_{r_i} + \boldsymbol{\omega}_{r_i} \times (\mathbf{I}_r \boldsymbol{\omega}_{r_i}) + k_h \boldsymbol{\omega}_{r_i}^2 \tag{4.44}$$

Here, k_h and k_p again define the hub force and thrust constants of the rotor assembly respectively. The vector $\boldsymbol{r}_r^n$ defines the separation between the centre of the rotor and the nacelle joint. The parameters $\boldsymbol{\omega}$ and $\dot{\boldsymbol{\omega}}$ and $\ddot{\boldsymbol{p}}_{c_i}^r$ then define the angular velocity and acceleration of O_{r_i}, and the linear acceleration of rotor CG respectively.

Together, Equations (4.43) and (4.44) define the forces acting within the frame of the final link of the nacelle kinematic chain (or "effector"). The aerodynamic load, in addition to the inertial forces the rotor exerts on the nacelle, may then be propagated backward to define the interaction of the nacelle with the body of Polycopter.

4.5.6 Formalisation

The interaction between each of the Polycoptersub-assemblies is now characterised relative to the body origin O_b. Unlike the assumption presented in Section 4.4, the state of the system $\boldsymbol{x}_k$ is defined by the instantaneous 6DOF pose of the body, of the nacelles, and their associated rotor state at time t_k. The system inputs $\boldsymbol{u}_k$ are then a concatenated vector of the servo and rotor torques as $\boldsymbol{\tau}_\delta$ and $\boldsymbol{\tau}_\eta$, and $\boldsymbol{\tau}_\gamma$ respectively:

$$\begin{aligned} \boldsymbol{x}_k &= [\boldsymbol{p}^T, \boldsymbol{\Theta}^T, \boldsymbol{\delta}^T, \boldsymbol{\eta}^T, \boldsymbol{\gamma}^T]^T \\ &= [x, y, z, \phi, \theta, \psi, \delta_1, \delta_2, \delta_3, \eta_1, \eta_2, \eta_3, \gamma_1, \gamma_2, \gamma_3]^T \end{aligned} \tag{4.45}$$

$$\begin{aligned} \boldsymbol{u}_k &= [\boldsymbol{\tau}_\delta^T, \boldsymbol{\tau}_\eta^T, \boldsymbol{\tau}_\gamma^T]^T \\ &= [\tau_{\delta_1}, \tau_{\delta_2}, \tau_{\delta_3}, \tau_{\eta_1}, \tau_{\eta_2}, \tau_{\eta_3}, \tau_{\gamma_1}, \tau_{\gamma_2}, \tau_{\gamma_3}]^T \end{aligned} \tag{4.46}$$

Here $\boldsymbol{p}$ and $\boldsymbol{\Theta}$ define the instantaneous position and pose of the Polycopterin the body frame; $\boldsymbol{\delta}$ and $\boldsymbol{\eta}$ are vectors defining the nacelle roll and pitch deflections respectively; and $\boldsymbol{\gamma}$ defines the angular position of the rotors.

To be able to express the evolution of the Polycopter's state $\boldsymbol{x}$ over time, it must

be written in a form that is easily integrated in OpenMAS [66]. This function is referred to as the "dynamic descriptor" function d, which characterises the system change across it's "n"DOF over time. The relationship between $\boldsymbol{x}_k$ and $\boldsymbol{u}_k$ and the state update $\ddot{x}_k$ for the Polycoptermust then be expressed in the form of Equation (4.47):

$$\ddot{\boldsymbol{x}}_k = d(\dot{\boldsymbol{x}}_k, \boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{w}_k) \tag{4.47}$$

Here $\boldsymbol{w}_k = \mathcal{N}(\mathbf{0}_{15\times1}, \mathbf{I}_{15\times15}\boldsymbol{\sigma})$ defines the state noise vector, where $\boldsymbol{\sigma} = [\sigma_p^2, \sigma_\Theta^2, \sigma_\delta^2, \sigma_\eta^2, \sigma_\gamma^2]^T$ denotes the standard deviation vector of the state-specific noise signal. To be able to define Equation (4.47), Equation (4.19) must be expressed in terms of the system accelerations $\ddot{\boldsymbol{x}}_k$ in each DOF. This is achieved by expressing the system's equations of motion in the general form of Equation (4.48) [222]:

$$\mathbf{M}(\boldsymbol{x})\ddot{\boldsymbol{x}} + \mathbf{C}(\boldsymbol{x}, \dot{\boldsymbol{x}})\dot{\boldsymbol{x}} + \boldsymbol{g}(\boldsymbol{x}) + \mathbf{J}(\boldsymbol{u}) = \boldsymbol{\nu} \tag{4.48}$$

Here, $\mathbf{M}$, $\mathbf{C}$ and $\mathbf{R}$ are the system's inertial, Coriolis and centripetal, and potential coefficient matrices respectively. This expression is synonymous to the general form used to describe the rigid-body characteristics in Section 3.1. Expressing the system in terms of the accelerations in each DOF, Equation (4.48) is rewritten in the form of Equation (4.49):

$$\ddot{\boldsymbol{x}} = \mathbf{M}(\boldsymbol{x})^{-1}\left(\boldsymbol{\nu} - (\mathbf{C}(\boldsymbol{x}, \dot{\boldsymbol{x}})\dot{\boldsymbol{x}} + \boldsymbol{g}(\boldsymbol{x}) + \mathbf{J}(\boldsymbol{u}))\right) \tag{4.49}$$

Here $\mathbf{M}$ is a symmetric positive-definite matrix defined by the partial derivative of the system definition in Equation (4.19) with respect to the system's accelerations $\dot{\boldsymbol{x}}$ across each DOF. Similarly, $\mathbf{J}(\boldsymbol{u})$ defines the system's input Jacobian, defined by considering the partial derivatives with respect to each of the system's inputs $\boldsymbol{u}$. Finally, the gravitation vector $\boldsymbol{g}(\boldsymbol{x})$ is defined by the system's gravitational coefficients, which is dependant on the instantaneous configuration of the system.

The forces acting on the Polycopterand nacelle assemblies due to the Coriolis and centripetal forces are characterised by it's *Coriolis matrix* $\mathbf{C} = \mathbf{C}(\dot{\boldsymbol{x}}, \boldsymbol{x})$. In classical mechanics this matrix its components are attributed to the Coriolis and centripetal forces and are expressed as coefficients of the form $\dot{\boldsymbol{x}}_i\dot{\boldsymbol{x}}_j$ and $\dot{\boldsymbol{x}}_i^2$ respectively. $\mathbf{C} = \mathbf{C}(\dot{\boldsymbol{x}}, \boldsymbol{x})$ is defined by it's relation to the system's mass matrix $\mathbf{M} = \mathbf{M}(\boldsymbol{x})$ via it's corresponding *Christoffel Symbol* [16]. The relationship between element $\mathbf{C}_{ij}(\dot{\boldsymbol{x}}, \boldsymbol{x})$ and the corresponding

Christoffel symbol Γ_{ijk} is given in Equation (4.50):

$$\mathbf{C}_{ij}(\boldsymbol{x}, \dot{\boldsymbol{x}}) = \sum_{k=1}^{n} \Gamma_{ijk} \dot{\boldsymbol{x}}_k = \frac{1}{2} \sum_{k=1}^{n} \left(\frac{\delta \mathbf{M}_{ij}}{\delta \boldsymbol{x}_k} + \frac{\delta \mathbf{M}_{ik}}{\delta \boldsymbol{x}_j} - \frac{\delta \mathbf{M}_{kj}}{\delta \boldsymbol{x}_i} \right) \dot{\boldsymbol{x}}_k \tag{4.50}$$

With each of the terms in Equations (4.48) and (4.49) defined, the dynamic model of the Polycopteris given in a form that can be integrated easily. Equation (4.47 now captures the time-variant pose of the body, as well as the instantaneous configuration of each nacelle so that the morphology of the system may be characterised over time. Following the proposition of the Polycoptermodel, a preliminary investigation into viable control techniques can now be undertaken with the model in it's current form. This is however the focus of future work towards the design of effective control mechanisms for the Polycopterstyle UAS.

4.6 Conclusions

In this chapter, the novel concept of the *Polycopter* is proposed for the first time. Using the principles of symbolic modelling presented in Chapter 3, two formulations of the vehicle's dynamics are presented under two assumed conditions i) small nacelle deflection angles and ii) massive nacelles.

The chapter begins by introducing the motivation behind morphing aerial systems and their advantages over traditional unmanned systems. From the associated literature it is shown, morphing (hybrid) systems have the potential for extended flight time, enhanced manoeuvrability and failure tolerance by being able adapt to new operating conditions. In addition, aircraft with heightened control over their lifting capabilities present numerous opportunities for bio-mimetic systems where the emulation of multiple kinematic gaits is novel. This heightened level of control over the aerodynamic forces creates the opportunity to design the flight characteristics of the vehicle in flight.

This motivation is used to provide context for the design of the Polycopter; which utilises actuated nacelle assemblies to vector the thrust produced by it's three rotors. Each nacelle is actuated through 2DOF by a linkage positioned on the central body to create a morphing airframe. Unlike previous systems, the position of these joints gives the Polycopterunique authority over the centre of mass of the airframe and it's stability properties as the effective anhedral and dihedral angles are modified. Initially, a formulation of the system is proposed under the assumption that the nacelle deflections are small; such that the inertia of the system remains near constant. This formulation

is shown to be principally similar to many of the dual tilt-axis UAS proposed in the literature, which typically assume that only the rotor disk is deflected . It is clear however from the first formulation that the morphology of the system is not captured effectively under this assumption, despite demonstrating the thrust vectoring capabilities of the system.

By re-examining the dynamics of the Polycopter, an alternative formulation of the system is proposed by modelling each nacelle sub-assembly as a sequence of rigid-bodies in a kinematic chain. Here it is shown how the interactions between each of the components may be considered in sequence through the an RNE based analysis. As a result, the dynamic interactions within each component frame is defined in order to characterise the effect of the thrust and orientation of the nacelle on the accelerations of the overall airframe. This formulation of the system is effectively able to describe the state evolution of the Polycopteras a 15DOF system by describing the instantaneous transformation of it's three nacelles.

The presented models are shown to provide parallel descriptions of the Polycopterconcept under two different assumptions and associated control regimes. The system is distinctly over-actuated and presents a number of opportunities in terms of control strategy design and adaptive control. In the literature, feedback linearisation has been applied in the context of over-actuated UAS and will be the subject of future work. Further investigation into control regimes that are able to take advantage of the versatility of this UAS configuration will also be considered moving forwards.

Chapter 5

Literature Review - Collision Avoidance

Collision avoidance is a fundamental aspect of coordination in systems defined by multiple kinematic agents. *multi-agent systems* (MAS) operating as a collective, or swarm, are generally capable of reaching higher levels of performance, reliability and redundancy when compared to their individual agent members. However enhancing the level of autonomy in such systems so that an acceptable level of safety can be assured is a complex task. The emphasis on safety in this area can be seen comparable to that of conventional manned operations and is a challenge that limits the technology as a whole.

This is because amongst the difficulties in coordinating systems composed of multiple physical agents, there are significant challenges in communication, path planning and collision avoidance. These challenges are particularly prevalent in systems composed of *micro aerial vehicles* (MAVs) and other *unmanned aerial systems* (UASs) derivatives, as they are also subject to regulation if they are to operate inside public airspace.

5.1 A Modern Airspace

Autonomous collision avoidance in the air-traffic domain is a field of research that has seen increasing interest over the last three decades. This has primarily been fuelled by increasing conventional air-traffic, and thus the need for automation, but now also by the integration of unmanned services. The *National Air Traffic System (NATS)* of America and their program *NextGen*, in addition *EUROCONTROL* and their initiative *Single European Sky* are part of the reform attempting to both increase unilateral capacity, and heighten support for new technologies [73, 80, 161, 192]. Other research bodies, such as

the National Aeronautical and Space Agency (NASA) [217] with their software *Automatic Ground Collision Avoidance System (GCAS)* in conjunction with the United States Air-Force Research Laboratory (AFRL) [160] and the Defence Advanced Research Projects Agency (DARPA) [4, 178] are some of those also sponsoring research in this field.

One of the principle functions of conventional *air traffic management* (ATM) services is to provide real-time avoidance information from a designated ground station in order to maintain safe separation between flight paths. This can be seen as collision avoidance at a supervisory level; synonymous to a centralised avoidance regime. Ultimately this serves to reduce the chance of a collision scenario occurring initially, as information provided by the ATM alongside other systems such as *traffic collision avoidance system* (TCAS) is then used to support the pilot's own knowledge of the vehicle's safety [11, 109].

Today, unmanned systems are heavily reliant on information communicated to them about their surroundings, especially in scenarios when there is a collision possibility. Unilateral use of existing protocols such as *automatic dependent surveillance broadcast* (ADS-B) are being assessed as a platform for cross compatibility in an increasingly diverse and automated airspace [53, 73, 107, 126, 178, 217]. Such initiatives aim to make telemetry data widely available to neighbouring aircraft for purposes such as conflict detection and resolution and will be a requirement in the United States from 2020 [53]. Ensuring backward compatibility with systems without such capabilities, or scenarios where communication is denied, still pose significant design challenges to upcoming systems [45, 109]. Legislation supporting the viability of the free-flight concept can be found as extensions of the *visual flight rules* (VFR) in [77, 128, 130, 174, 204]

5.2 Cooperation and Non-Cooperation

Quantification of an unmanned system's ability to cooperate in the event of a collision scenario is becoming a critical part of its integration into the modern airspace [63, 93, 141, 184, 271]. The Federal Aviation Authority (FAA) states in their right-of-way rule (14 CFR 91.113) that any flight-worthy aircraft must be able to detect a conflicting trajectory craft and instigate a corrective manoeuvre once the separation becomes less than $150m$ [178]. In [141] a *target level of safety* (TLOS), synonymous to the *equivalent level of safety* (ELOS) in [178], is proposed as a measure of a system's ability to minimise risk of collision with respect to the equivalent human pilot. Capability is described through the hierarchical layers seen in Figure 5.1, with systems that are capable of interacting at all levels being considered equally capable to an equivalent manned aircraft [11, 93].

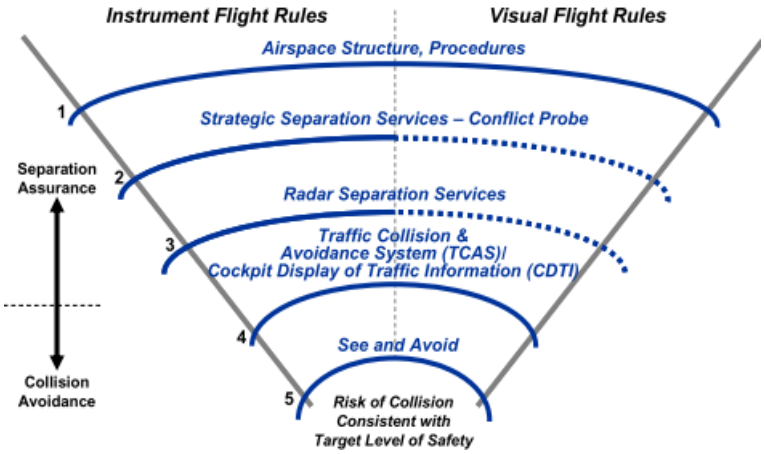


Figure 5.1: *The target levels of safety (TLOS) for unmanned systems operating in public airspace [141]. While the majority of conflicts are handled at TLOS level 1 and 2 (airspace structural design), SDA mechanisms typically operate between TLOS 4 and 5 at the point of visual confirmation. This is focus of this thesis.*

It is clear from Figure 5.1, that UAS operating in environments with civil aircraft must have systems in place to engage in collision avoidance not only by means of obeying the rules of the air, but act reactively to plan and orchestrate avoidance manoeuvres as a last resort. Requirements of this kind are often grouped into two distinct areas i) *co-operative*; responding to obstacle trajectory data communicated directly between aircraft (or ATM) or ii) *non-cooperative*; where obstacle trajectory data must be sourced from an appropriate system on board the aircraft [109]. Modern aircraft operating in civilian airspace must be capable of both cooperative and non-cooperative avoidance before a maximal TLOS can be achieved [141, 142].

Examples of the types of sensors currently being explored for increasing autonomy in manned and unmanned systems can be seen in Table 5.1. The complexity of the problem becomes more obvious when the possibility of transponding and non-transponding obstacles is also considered [56, 93, 109]. This essentially discerns between obstacles that actively transmit their trajectory data for the purpose of avoidance and those where the data must be determined by a local sensor system. There are several potential scenarios where obstacles are non-transponding; the obstacle is incapable, the systems are operating in a communication denied environment, or the obstacle is adversarial (see Section 5.3).

At the maximal TLOS, modern UAS are assumed able to assemble a description of an obstacle's trajectory that is adequate for collision avoidance, by combining communicated data when possible with data acquired locally. In Table 5.1, methods currently being used to provide this information are given with respect to their modality, the type of information provided and the processing required to attain a viable obstacle trajectory.

Protocol	Cooperation	Modality	Information	Object Trajectory
TCAS/ACAS	Cooperative	Active	Range & Altitude	Derived
ADS-B	Cooperative	Active	Position, Altitude & Velocity	Provided
MMW RADAR	Non-Cooperative	Active	Range & Bearing	Derived
SAR	Non-Cooperative	Active	Range & Bearing	Derived
Acoustic	Non-Cooperative	Active	Azimuth & Elevation	Derived
Laser/LIDAR	Non-Cooperative	Active	Range, Azimuth & Elevation	Derived
Optical	Non-Cooperative	Passive	Azimuth & Elevation	Derived
Thermal/Infrared	Non-Cooperative	Passive	Azimuth & Elevation	Derived

Table 5.1: *Classifications of modern sensor systems used in* sense detect and avoid *(SDA) operations and their modality as cooperative and non-cooperative mechanisms. The nature of the measurements made by the device determines whether the object's trajectory is sensed directly or must be derived computationally [53, 56, 77, 93, 107, 109, 178, 197, 212].*

The need for encompassing both non-transponding and transponding obstacle types becomes more apparent when considering the applications of UASs with respect to traditional aircraft; where obstacles such as trees and buildings must be considered alongside conventional air-traffic. It is therefore clear from Table 5.1 that a combination of sensing mechanisms is required to ensure an adequate level of integration with modern air-traffic communication systems and protocols. The need for parallel cooperative and non-cooperative systems is also seen in the literature [5, 55, 93, 109, 141, 143, 225]. Restrictions imposed on the *maximum take off weight* (MTOW) of a UAS may mean however, that hardware size and weight ultimately determine if the TLOS is achievable for a given UAS classification.

In considering collision avoidance in a MAS composed of collaborating UASs, an analogy can be drawn to the requirements of assuring safety in large scale automated ATMs [5, 47, 123, 137, 225]. This is because many of the challenges facing collaborative aerial systems are synonymous with traditional ATM and conflict resolution protocols [142, 211]. These protocols and algorithms are irrespective of the agent's classification or function in the airspace. Cooperative collision avoidance in the context of MASs can therefore be seen analogous to correction information received by a unanimous ATM system. It therefore stands to reason that for smaller scale systems to attain their maximum TLOS, they must have the innate ability to *sense, detect and avoid* (SDA).

5.3 Adversarial Attacks

In the consideration of both cooperative and non-cooperative obstacles, the potential for adversarial obstacles must be discussed. An adversarial attack is defined an a malicious act to disrupt or inhibit the function of the MAS. An adversarial obstacle however, is defined in these works as an object with an objective of inducing a collision between itself and the agent.

Adversarial behaviour in collision avoidance typically occurs in robotics and aerospace scenarios where the obstacle is an interceptor (i.e a UAV or missile) [105, 113, 150, 278]. In the literature, adversarial obstacles (UAVs) have been used to form the basis for *adversarial control*. Examples of this can be seen in in [49, 150, 186, 191]. In [186] the robotic herding of UAVs is demonstrated through specific placement of adversarial agents. The result is shown to effectively restrict the motion of a second team of agents which are seek to avoid collision. In the woks of Zengin and Dogan in [269] a probabilistic approach is used to model an adversarial scenario. A cost function is proposed to quantify placement

of UAVs around the target in formation in terms of threat from an adversarial obstacles. The cost function is used to create a cooperation strategy that both minimises the total threat exposure of the UAV team and the mean deviation in target separation throughout the presented pursuit scenario.

Another example for adversarial agent consideration is presented by representing an unknown hazard that seeks to inhibit the function of the system as an adversarial agent. In the works of Schwager and Kumar in [216], these obstacle are defined as unknown threats to the system with an uncertainty in their risk(effect) and must be avoided as the environment is explored. In [46] and [264], both authors focus on maintaining network connectivity and performance in the presence of adversarial agents. This is presented in the form of an adversarial game with an alternating-play algorithm. This aims to maximise the connectivity of the networked system whilst mitigating the adversarial agents attempts to block connection and cause information loss.

Adversarial obstacles represent another challenge to the reliability of collision avoidance systems for modern UAS. This is primarily because their behaviour may act against many of the assumptions made in Section 5.2. Depending on the complexity of the agent, identification of an adversary from obstacles that are simply non-cooperative presents another set of challenges. While consideration of adversarial agents is a component contributing to the dependable safety of a systems, it is not addressed as part of these works.

5.4 Sense, Detect and Avoid

The principle of sense, detect and avoid has seen substantial interest in the last decade as a means of ensuring an acceptable TLOS in upcoming autonomous systems. Similarly, more recent efforts to increase the safety of manned operations have sought to combine modern sensing technology and SDA principles, with existing flight deck systems (such as TCAS and ADS-B) [107]. Many of these approaches to automating collision avoidance using information communicated, or acquired locally, are cited in the literature under the principles of *free-flight* [136, 137, 162, 185, 226]. This ideology, albeit conceived in the nineties, has promising grounds in the future of a diverse and automated airspace [73]. The need for alternative systems for handling increasing air traffic is cited in [128]. Feasibility studies for free-flight base air-traffic topologies are historically based on terrestrial air traffic in [56, 106, 128, 185, 204]. However with the increasing sophistication of civilian and military unmanned systems, the free-flight concept is re-examined as a means for developing automated mechanisms for handling collision avoidance in the context of a

diverse airspace in [77, 93, 104, 107, 123, 141, 178, 185, 192].

For the majority of conventional airspace operations SDA algorithms aim to assist in the generation of emergency avoidance trajectories, in the event that TLOS 1-4 (Figure 5.1) have failed to prevent the conflict. Principally, SDA algorithms are fundamental to replicating a pilot's ability to interpret an evolving scenario and respond to unforeseen behaviour [93]. This is often referred to as *reflexive* avoidance in the literature [63, 225]. This is critical in situations where the behaviour of conflicting aircraft may contradict communicated information, received telemetry, or established protocols. It is this premise that not only makes the SDA algorithms desirable for reflexive collision prevention, but also for their independence on external sources.

Today collision avoidance in unmanned systems is predominantly achieved at the level of mission design and path planning; by designating airspace occupancy (synonymous to TLOS one and two) [141]. Control over the airspace however, particularly at ground level, may mean that this is unfeasible. This is especially apparent in the use of smaller systems such as MAVs; which today, are being designed for use in a vast array of environments from search and rescue to elaborate light displays [63, 209].

In the literature, a system's capacity for SDA based avoidance is limited by the reliability and accuracy of the data that can be acquired by local hardware. In scenarios such as search and rescue, or collaborative systems, autonomous MAVs are now required to manoeuvre through unknown environments and respond to situational changes in real-time. This implies that obstacle trajectory data must also be acquired in real time; filtered and fused with data from other sensors to best estimate the states of identified obstacles. Fundamental to this is the computational power to process the obstacle's trajectory and compute it's own resolution trajectory accordingly [95, 209, 212]. Hybrid systems often utilise communicated data as a method of providing *dead reckoning* for obstacle trajectory estimations. These methods seek to benefit from the apparent strengths of both cooperative and non-cooperative based SDA approaches [5, 141].

Given the dependence of SDA approaches on local hardware, such systems are often cited in conjunction with higher-level path planning algorithms capable of avoidance at TLOS three and four [96, 126]. Hierarchical approaches may present a method of encoding predictability through procedural collision avoidance whilst simultaneously enhancing a system's capability to respond dynamically to changing environments as the risk of collision increases. It is likely that, if proven successful in the UAS industry, many developments in the field of SDA technology will be directly applicable to the next generation of manned aircraft support tools; such as TCAS, ADS-B derivatives, and intelligent

traffic management tools. The need for significant developments in the verification of such systems is cited in [109, 147, 178, 212].

Emerging unmanned aerial systems today are facing the strict requirements of being able to interlace with existing air-traffic infrastructure on a macro scale to increase awareness and the safety of nearby aircraft. On a micro scale, systems must be able to assure an equivalent level of safety in negotiating dynamic time-varying environments, obstacles or communication loss [123, 225]. The importance of such systems to be able to operate in real-time, with low-latency, is clear when the implications of collision pose a risk to people, other vehicles in operation, or the environment.

5.5 A Review of the State of the Art

With the breath of use cases for MAVs and UAS in the civilian sector, there has been substantial development in the autonomy of aerial systems. As a result of this, interest in more sophisticated systems for handling fundamental interactions such as collision avoidance is increasing. MAVs are well suited for applications in close, cluttered environments involving many agents. Collaborative tasks such as construction, package delivery and object manipulation make collision avoidance essential in the development of the technology.

In the literature surrounding multi-agent and collaborative robotic systems there have been numerous methods proposed for handling collision avoidance. However dependence on, or inference of, a low latency omnipresent network is often cited. In this section, several of the key approaches to MAV collision avoidance are presented along with relevant concepts from the autonomous air-traffic domain. An emphasis is also made towards techniques for handling more stochastic environments in which such networks cannot be guaranteed.

5.5.1 Potential Field Approaches

Stigmergic or "potential" field based coordination techniques are one of the more widely used approaches to inherit collision avoidance within the MAS community. Typically, a game field is assigned wherein all agents, obstacles and goals are given a charge or cost metric [103, 131]. Goal locations are assigned a metric, force or potential to differentiate them from other agents in order to exemplify attractive behaviour. Assigning identical charges to obstacles and agents are also shown to shown to create repulsive behaviour in [41, 54]. The resultant force acting on the agent is then used to infer it's optimal

trajectory at the next time step or over a defined horizon. This allows trajectories to be evaluated quickly using conventional gradient descent methods. The approach is well summarised in [5, 21, 91, 103], outlining some of the issues regarding local minima inherit in these methods.

An example of the application of stigmergic fields applied in the air-traffic domain can be seen in the works of Tomlin et al [136]. Trajectories are calculated as a result of the current global obstacle configuration. The *navigation function model* initially proposed by Rimon and Koditschek [203], demonstrates how geometric functions can be used to manipulate field lines to represent complex obstacle structures. The representation of the obstacles does not consider the possibility of (zero velocity) local minima occurring; a feature common to many potential field methods.

A more recent example of complex obstacle avoidance applied in the context of MAVs can be found in [172]. A quadcopter MAV is shown to acquire obstacle data locally, with a mounted LIDAR, ultrasonic rangefinder and stereoscopic camera to navigate a cluttered environment [212]. The cumulative force acting on the agent is the weighted sum of the goal influence and obstacle influences as they are observed within a bounded 3D horizon. A motion model is presented to predict the cost of a set proposed trajectories based on the terminal cost at the horizon [54, 172]. The work conducted in [41] proposes the introduction of a gyroscopic force, inspired from the navigation function model study [203]. The perspective of the problem is also moved to the local domain by representing agent knowledge of obstacles through a detection shell. Obstacle forces and manoeuvre trajectories are then calculated directly from sensed data, presenting an alternative to the methods highlighted in [112, 138]. The gyroscopic force is appended to the agent force summation in order to calculate the control inputs from the total influence [172]. These forces correspond to the goal orientated potential force, a damping force and the gyroscopic forces respectively. The derivation of their force expression is described in more detail in [41] alongside the novel concept of "braking" force - a virtual force used as a mechanism for regulating the distance between the agent and the nearest obstacle.

Paul et al. introduces a similar method of 3D coordination and collision avoidance applied to *unmanned aerial vehicles* (UAVs) in [188]. Although the work was limited to simulation, the author describes a technique of using 3D potential fields to orientate a virtual UAV relative to a leader UAV using time variant potential fields. The author's formulation of the resultant force can be seen as the sum of the influences on the virtual leader, the inter-vehicle forces, collision and obstacle avoidance forces. Using this technique, successful separation is shown to be maintained in their presented example

assuming that communication is present. The effects of communication delay, or limited data availability are not however discussed.

The works of Suzuki and Uchiyama in 2009, demonstrate an application of a decentralised structure in the proposal of *bifurcating potential fields* [240]. The study outlines the coordination and intrinsic collision avoidance of a team of helicopters in formation flight using transitioning local potential fields similar to that seen in [188]. The author proposes two virtual forces; repulsive and *steering* based influences. Their proposed steering potential function, together with an exponential repulsive force, function to ensure safe separation during a formation flight. Using this approach the author successfully simulated both linear and orbital manoeuvres by manipulating the local potential field [103, 240].

The advantage of many of these presented methods is their comparative simplicity and limited requirement for obstacle knowledge. As a result, use of potential fields is well documented in the multi-agent community. The principle disadvantage of this approach is the appearance of local minima due to the inability to describe the complete array of available escape trajectories. In such cases, a supplementary approach may be necessary to move the system out of deadlock (a static freeze) and modify the associated fields.

5.5.2 Protocol & Rule-Based Approaches

Another more established mechanism for collision avoidance within aerial systems, air-traffic literature and MASs is the concept of protocol-based conflict resolution. This category of collision avoidance involves the construction of predefined responses to behaviour of a second approaching obstacle. Typically the established protocols, or rules, are unilateral across all active agents. In the context of air-traffic management, many localised negotiation techniques have been proposed under the free-flight concept (introduced in Section 5.1).

In [77], the *extended flight rules* (EFR) are proposed as an iteration on the conventional *visual flight rules* (VFR) for the purpose of autonomous system integration. The premise of the EFR is that the agent's responsive behaviour is characterised by the obstacle's approach vector, position and priority in accordance a series of predefined rules. Conventionally, protocol-based algorithms assume unilateral adoption of the avoidance protocol to prevent contradictory behaviour. While simple to implement, in the event the obstacle's behaviour cannot be correctly compartmentalised into a known regime, then the behaviour of the agent becomes ambiguous and unsafe. This is especially relevant when there is uncertainty in the obstacle's trajectory, or if the agent is exposed to unforeseen events or behaviours.

In the works of Tomlin and Hwang [111, 112], they define the concepts of an *exact conflict* and *inexact conflict* as a means of protocol design. These scenarios characterise collision events where all agents are mutually convergent, or only a subset of involved agents converge respectively. The prospect of an exact conflict occurring for agent numbers greater than three is known to be unrealistic but aids in generalising an approach that considers it's possibility. The airspace is partitioned into segments for which the worst-case conflict is computed in addition to the necessary safe heading change which results in the theoretical minimum deviation from the desired trajectories. Rules for determining the minimum heading change for each agent are presented for either case, with the author demonstrating both robustness to trajectory uncertainty and tolerance for asynchronous agent participation in the macroscopic manoeuvre. Despite the described method being effective at ensuring safe separation at the time of collision, the resulting trajectories are sub-optimal with respect to the resultant course deviation and heading change. The author uses this algorithm to propose a new "augmented ADS-B" protocol; appending the conflict resolution centre, position and velocity and heading of each aircraft involved. The limitation of the algorithm proposed by Tomlin and Hwang is the issue of communication between cooperative and non-cooperative groups. While a protocol that propagates the required conflict data to all receiving aircraft would be valid, the presence of an apparent non-cooperative aircraft would create an unsafe operating environment.

Many of these concerns are discussed in [227]; in which a hybrid system is proposed to handle both cooperative and non-cooperative elements in a multi-agent context-through the use of a rule-based mechanism. Conventionally, protocol-based algorithms assume unilateral adoption of the avoidance protocol to prevent contradictory behaviour. The mechanism is proposed as an initial layer of a *negotiation protocol*. This infers a layer of established procedures be enacted within all transponding agents, while the non-cooperative agents attempt to infer the optimal negotiated path by drawing on a hybrid, game theory based element. Although the proposed system is shown to support a diversity of *subscribing* and *non-subscribing* aircraft, the resulting protocol is shown to be difficult to compute in real time, with heavy dependence on network communication and available hardware.

A probability-based alert protocol system is also proposed in [137] for the characterisation of collision scenarios under the free-flight principle. The mechanism involves the determination of appropriate avoidance manoeuvre via a layered alert system as the probability of collision exceeds threshold values. The author considers numerous sources of uncertainty in the trajectory measurement of the agent and the obstacle. A probability

for the obstacle entering the agent's protected zone is then assigned, and is reevaluated through an iterative Monte-Carlo process. A conflict is confirmed when probability thresholds are violated in both the lateral and longitudinal dimensions and used to initiate an established response manoeuvre. The presented method is shown to mitigate conflicts between two agents within a 95% confidence interval, however is only shown in the context of a two agent collision. Look-up tables of known flight data are also used with linear interpolation to determine the contours of probability, suggesting such systems may be susceptible to novel obstacle behaviour. The proposed manoeuvres are also cited not to consider properties such as increased fuel burn and course deviation.

5.5.3 Game Theoretical Approaches

Game theory is a concept that was originally created in the field of mathematical economics in the early 1930*s* to describe strategies in cooperative, non-cooperative and adversarial economics. More recently game theory has found applications in other fields, including computer science and robotics. Furthermore it has been developed along side *optimal control* theory in search for more sophisticated control methodologies for the aerospace industry. In the 1950*s*, game theory and game-based decision making was applied in the analysis of dynamical systems represented by differential equations, this approach was later given the more general name *dynamic games* [22].

The literature highlights three distinct game theoretic approaches used in collision avoidance namely i) *cooperative*, ii) *non-cooperative* and iii) *adversarial* games. Under the cooperative regime, conflict scenarios are typically analysed in order to encourage mutual benefit; non-collision. It is often assumed that information describing the intended trajectory of the second party is communicated, either directly by the agent or through some central air-traffic controller. Coalition(or independence) values are typically assigned to a set of defined actions so that their associated utility reflects both its impact on the agent and the cooperator.

In non-cooperative regimes, the cost of a specific actions is evaluated based on a one-sided analysis of the conflict scenario. Here, communication with a central controller, or other agents, is assumed denied. The utility instead reflects the likely actions taken by the second party based on a representative model and information provided by local sensing mechanisms. As a result the non-cooperative approach emphasises the benefit to the agent, rather than the obstacle, in the calculation of the escape trajectory [22,44,114].

The third case, referred to as adversarial collision avoidance, focuses on cases where the obstacle is an aggressor that aims to induce a collision with the agent. The agent,

seeking to avoid collision, typically adopts a *minimax* strategy minimising the worst-case utility of the set of actions available to both the agent and aggressor. While not the emphasis of this thesis, adversarial avoidance is a concept well established in the aerospace industry.

From the types of dynamic games, dynamic game theory (the expression of dynamics through differentials) lends itself most to the problem of collision avoidance. This is because, for vehicles moving in three dimensional space there exists a continuum of possible actions that cannot be resolved in discrete or Boolean form. Such games can also be described in the discrete or continuous time domain, corresponding to a continuum of game cycle iterations $(u_t, u_1, u_2..u_\infty)$, referred to as *levels of play*. Whether the games are discrete or continuous then determines whether the decisions are formulated from a dynamical difference or a differential equation respectively, which describes the evolution of the scenario [22, 44, 105, 168, 246]. In [114, 140] the concept of a differential game is presented in the context of a *pursuit-evader* collision avoidance scenario. The proposed strategy is based on a Taylor series approximation of the value function gradient over a discrete grid in the agent's kinematic state-space. By calculating numerous optimal trajectories through the state-spaces, a switching logic is used to move between the optimal path conditions in order to refine the avoidance of an incoming vehicle. Similarly in [23], the concept of dynamic games is applied in the context of an alerting mechanism in a two aircraft conflict scenario. Using a primitive model of the agent's kinematics, a conflict region is defined indicating a likelihood of collision with the conflicting aircraft as the computational solution to their proposed value function. The concept of infinite differential games is explored in the context of UAV coordination and collision avoidance in [248]. The author constructs a formation control mechanism through the analogy of a pursuit-evader game [105, 164, 252]. The controller then acts to maintain the evader separation. The principle of the approach aims to minimise the final system control vector whilst maximising the final vector of desired points. The author determines the Retrograde Path Equations in order to define the trajectories necessary to meet the optimal solution. The resultant strategy takes the form of a value function that penalises deviation from a desired position, as a function of the author's linear drag assumption and the time step. Ten agents are used in the evaluation of the proposed strategies in [248], where some degree of instability can be seen in both providing collision avoidance and coordination. The author comments on this; citing the simple *Bang Bang* control regime used to enact the trajectories. A more sophisticated control mechanism would likely improve performance and limit the observed overshoot. A noticeable limitation of the method

proposed is the potential non-existence of a solution to the retrograde path equations. In such cases approximations or reductive assumptions may be necessary to modify the agent's dynamics to ensure a valid solution exists.

A more recent branch of dynamic game theory, termed "Hybrid Game Theory" (HGT), facilitates the modelling and coordination of continuous dynamic systems with discrete logical elements [246]. This is done through the presentation of a "Hybrid Time Trajectory"; defined as a continuous sequence of trajectory intervals where discrete actions take place instantaneously. The author utilises the hybrid control structure to ingrain a series of trajectory acceptance conditions, designed to reflect safety constraints on the system's states through defining safe and unsafe state subsets [164]. The authors demonstrate the application of HGT using an avoidance scenario containing two aircraft with dual *cruise* and *avoid* modes. The application of the HGT demonstrates how a logical transition between modes can be integrated into a continuous game strategy in order to preserve aircraft separation [246].

An advantage of the game-theoretic principle is that it allows the inherent representation of many agent and vehicle constraints. The cost representation often allows the consideration of aspects such as, actuator dynamics, asymmetrical performance, or fuel consumption. Effective, scalable conflict representations is often cited as a drawback of this method as it's application on-board small-scale decentralised systems quickly becomes infeasible.

5.5.4 Multi-Agent Approaches

Within the multi-agent literature and computer science, there exists several methods of collision avoidance based on multi-agent communication and negotiations. Some of these approaches take inspiration from the works of Reynolds in [199] in the proposition of the "boids" model; mimicking the social behaviour of birds. This approach proposes the use of three principles of *separation*, *alignment* and *cohesion*. The sophisticated emergent behaviour in swarms resulting from these simple principles is though to have contributed to many of the collision avoidance methods relating to large multi-agent systems.

Multi-agent methods are otherwise well documented within the context of the air traffic domain in [226, 227, 254, 255, 260], but only more recently the reflexive sense and avoid problem. Typically, agents enter a negotiation by presenting proposed solutions to the collision scenario based on their optimal path. A global utility function is applied to the proposals to assess the most viable solution. This can be seen in distributed agent-to-agent negotiation mechanism proposed in [279]. Here, the author devises a *monotonic*

concession protocol as a mechanism for selecting a pair of avoidance trajectories from a collective set assembled by the agent set. The author is able to define a horizon within which the negotiations take place, designated as twenty minutes, to emulate the effects of reduced range on the cooperative system.

Another approach that takes inspiration for the game theory mechanisms is the principle of "satisfication" applied in [14, 104, 124, 224]. This approach encourages a more social approach to traditional game theory in which the agents optimise their trajectory based on it's own objective parameters. Instead, priorities are allocated to the agent set and assessed in accordance to their "selectability" and "rejectability". These measures provide a means of assessing a proposed solution in terms of the mutual intentions of the agents, their priorities and the costs incurred. These solutions are evaluated centrally and subgroups are devised that favour solutions with highest reject-ability or select-ability based on their possible trajectory changes. While this concept is known in the air-traffic domain, some progress has been made towards the free-flight application (synonymous to the SDA problem) in [104, 204]. Due to the assumption of fixed flight lanes in these scenarios, it is widely assumed that avoidance is occurring at a fixed altitude (within a 2D plane) within a system of homogeneously communicating and cooperating aircraft.

The concept of a MAS is presented in the form of a aircraft/airspace system by Wangermann and Stengel in [255], similar to that of [104] except under the domain of a traditional ATM system. This system is collectively referred to as *multi-aircraft agent system* [204, 226] which expresses the airspace as several agent types; aircraft, operators and traffic management units. In this system, avoidance negotiations are peer-to-peer; negotiated directly between two agents. Despite the assumption of comparatively simple aircraft dynamics, the demonstrated structure is shown to be applicable to localised conflict resolution. In [125] the multi-aircraft system concept is furthered through a virtual currency/token system that facilitates transactions between avoidance agents in a selfish manner. Although Jonker et al. presents the concept through a tactical airport planning system, the economics/trust based swarm negotiation presents an alternative approach to assessing proposed trajectory reassignments in the SDA problem.

Several of the presented multi-agent methods are demonstrated to be promising approaches to conflict resolution. The assumption of a universal communication method is often made to allow the brokering of tokens or consensus broadcasts. While this is often the case in high-altitude TCAS operations, it is not valid when considering mixed airspaces where communication cannot be assumed and obstacles may be inert.

5.5.5 Geometric Approaches

Geometric approaches to SDA avoidance are centred around the use of geometric obstacle and conflict descriptions to generate an appropriate avoidance trajectory. One concept based on the principle of the *collision cone* (CC) is the *velocity obstacle* (VO), which has seen increasing attention in the multi-agent and intelligent robotics community for it's efficiency and intuitive nature. Originally proposed in [82], constraints on the agent's velocity are formulated using known obstacle geometries as they move through the agent's field of view. From this constraint set, a region containing all collision trajectories is constructed from the obstacle's characteristic radius and translated into the velocity space. The resulting region constrains the agent's choice of velocities at the next time step [65, 219,259]. Similarly, a multi-obstacle scenario can be represented through the union of their VO. The compound region then demonstrates where a selected velocity would induce a collision with either obstacle [219, 259].

In [261] a similar VO method is used to define velocity constraints whose union over a time define the permissible velocities for the agent using an iterative planner. The current method is defined using the Dublin's car model for a car-like robot, whose dynamics are used in the formulation of these sets. The complex description of the boundary regions make the approach largely unfeasible in small systems despite it's reduced dependence on accurate sensor data.

While the VO is shown to be valid for static obstacles and obstacles moving with constant velocity, the *reciprocal velocity obstacle* (RVO) was later proposed to negate oscillatory behaviour between intelligent agents [215]. This is achieved by mediating the VO constraint in accordance to the approach trajectory of the second agent by averaging their velocities. While resulting in smoother trajectories, the direction of pass is not explicitly agreed- giving way to a phenomena known as *reciprocal dance*. In such cases, agents fail to resolve headings that reduce the chance of collision at the next time step. As the first agent determines that passing the second agent on the right would be most optimal, the opposing option (to pass on the left) becomes the most feasible for the second agent. This combination only acts to perpetuate the collision encounter to the next time step. Although the RVO method is shown to improve the generation of smooth avoidance trajectories, it cannot guarantee it theoretically [27].

The VO principle was later expanded upon in the works by Berg et al. [25, 27]; by furthering the concept with the consideration of obstacle accelerations in the *acceleration-velocity obstacle* (AVO). The kinematic velocity obstacles (KVO) presented in [258] demon-

strate how additional agent kinematic assumptions and constraints can be incorporated into the approximation of the escape velocity search volume. This is achieved using a simple kinematic representation of the system to augment the shape of the velocity obstacle. The study of reciprocal collision avoidance also brings to light the reciprocal nature of decision-making within a homogeneous multi-agent system [87, 219]. This phenomenon is considered by parameterising agent responsibility for the avoidance manoeuvre in order to redefine the AVO set as the "Reciprocal" Velocity Obstacle (RVO). Car-like agents are used in this study however; for which the method and it's scalability is assessed. An advancement on the RVO problem has been proposed to negate the causes of reciprocal dance by augmenting the VO and RVO regions to define the *hybrid-reciprocal velocity obstacle* (HRVO). The HRVO alters the apex of VO regions in order to exemplify different behaviours depending on the relative motion of the obstacle. This is shown to effectively establish direction of pass in [27, 65], and greater resilience to obstacle trajectory uncertainty. This method is also shown to demonstrate a greater ability to resolve escape trajectories, however the non-linearity of the constraint set is known to result in abrupt and inefficient course corrections.

One of the most recent derivatives of the VO concept, termed *optimal reciprocal collision avoidance* (ORCA), bases it's constraint formulation on linear obstacle definitions. The result is a distinctly linear and smoother response to obstacle motion when compared to previous VO derivatives in [65,259]. The agent's minimum trajectory correction is computed from the obstacle geometry as the change necessary to move the agent onto a plane (line) of feasibility. Resolving feasible trajectories from geometric constraints often results in multiple proposed solutions in the literature and is subject to strategy [82]. The agent's velocity is typically selected from a region of feasibility to determine the agent's trajectory over the proceeding time step. The ORCA algorithm is shown to have a distinct advantage over other geometric algorithms in [65, 259] due to it's linear constraint formulation. Although these approaches have been widely used in multi-agent systems such as pedestrian modelling and small robotic systems, they face challenges in symmetric scenarios where a phenomenon known as *deadlock* can occur. Otherwise, numerous search methods are known to have been proposed for determining optimal trajectories from geometric constraint sets; global and heuristic methods are cited in [82], region reductive methods in [25], avoidance parameter optimisation [229, 259], genetic algorithms [156, 234] and dimensional decomposition in [38].

A similar collision cone based cost function optimisation method is proposed in the works of Daniels et al. [54]. The authors present a hybrid concept of a geometric

obstacle description with a simulated annealing optimisation procedure to determine an appropriate aircraft trajectory. The cost function penalises the UAV separations and adjusts the agent headings for both cohesive motion and ensure the trajectory does not violate the conical constraints. The results of this study show potential despite only being proposed as a method of reducing near-misses. However globalised approaches are shown to struggle with larger agent populations.

5.5.6 Optimisation & Hybrid Approaches

With respect to the collision avoidance problem, optimisation is also cited as a technique for ensuring aircraft safe separation. Generally speaking, these approaches attempt to express the collision avoidance scenario in terms of observed obstacle trajectory parameters, an agent kinematic model and a series of constraints (objectives) to maintain. A current example of this procedure is TCAS, which preforms an optimisation across potential climb/descent flight paths to determine the option with the least control effort to achieve the safety constraint. Although the TCAS recommendation is far from optimal in that no consideration is made for the objectives of each agent, safety of both aircraft can be assured using basic assumptions and limited situational knowledge [5, 76].

Fox et al. in [86], introduces the concept of a "dynamic window" to search for viable avoidance velocities. The dynamic window cost function relates the agent's trajectory cost in terms of it's heading, obstacle separation, and speed subject to scalar weightings. Their approach aims to reduce the computational cost of the optimisation method by reducing the search to the vehicle's linear and angular velocity. The author also defines a region of admissible velocities assembled from a) the trajectories that do not intersect with the obstacle and b) the available breaking velocities to avoid interception. This is achieved by introducing a parameterised expression representing the robot's braking acceleration. Finally, the dynamic window is introduced to unify the robot's acceleration limits with the feasible region. The resulting dynamic window is then discretised and searched by maximising their objective function. The resulting velocities are then implemented for the next time step [82, 86, 229].

Another more recent concept relevant to UAV collision avoidance, is outlined in the works of Sislak et al. in [226, 229]. This paper explores the application of trajectory optimisation from the perspective of a centralised and decentralised airspace optimisation problem. The author defines a common objective function describing the global utility of the aircraft configuration. In this approach, the local cost functions are derived to reward compliance towards this global utility, seen by the summation of each agent's cost,

providing a cost of the global configuration [84,144,229]. The costs of loss of separation are defined as violations made on the separation requirements across the prediction horizon and summed for all agents. The deviation costs are attributed to the sum of the Euclidean separations between the optimal position (waypoint) and the position at the end of each agent's prediction horizon. Control actions are selected based on the cost optimisation through a simulated annealing process.

Dynamic programming, or the dynamic decision problem, is introduced in the context of collision avoidance in [107] as an extensions of Bellman's principle of optimality [231]. Here, the author presents an algorithm for planning an obstacle avoidance manoeuvre through a discrete decision space that converges on a target location. The process of dynamic programming is inherently efficient due its recursive nature and the propagation of optimal sub-trajectories (sub-structures) through *memoization* by retaining costs of prior sub-trajectories.

The authors present a cost function that considers a set of possible actions for each agent as a result of the current system state to determine the current reward value. The potential trajectories, are optimised based on their implied pay-offs over a defined horizon as a result of proposed actions. A "discount" factor is used to reduce the cost of the actions towards the horizon where the trajectory is less certain. A series of transition models are then used to evaluate the trajectory costs over the horizon citing the *Bellman equation* [231]. This allows a trajectory to be optimised based on the current state evaluation, in addition to the performance over the proceeding horizon via known transition models. An application of this method to the traffic avoidance are outlined in [29, 107, 119, 120, 134]. The formulation of the value functions and transmission models are multi-dimensional and can be expanded to infer the evolution of other performance parameters also. While the global avoidance scenario is considered in [29, 134], the assumption that each aircraft has unilateral knowledge is not representative of real world scenarios.

5.5.7 Biologically Inspired Approaches

Biologically inspired algorithms and systems have been a long-standing interest to the robotics community, with numerous techniques being adopted in a wide range of fields [233]. In the context of coordination of unmanned systems, there are several cited works that draw upon biological analogies for the purpose of guidance and obstacle avoidance; see [88, 133, 195, 196, 233]. A good example of this is the conception of a visual field based model, termed the "angular velocity detector unit", that emulates the in-flight behaviour of a

honeybee [52]. This neuron-based model infers the relative motion of neighbouring surfaces in the visual field by observing the apparent change in spatial frequency of surface features. The difference between left/right stimuli and their proposed controller is shown to be sufficient to both imitate electro-physiological data of a bee, and facilitate collision avoidance in their corridor example. The algorithm uses limited environmental data to avoid collision, however further development is needed for more sophisticated obstacle topologies.

Direct applications of *neural networks* (NN) to collision avoidance in autonomous systems can be found in [97, 171, 235, 238]. Durand et al. in [171] demonstrates how the approach bearing and distance data is used to define an appropriate heading change using a single layer NN. The concept is demonstrated in a typical two agent aerial conflict. The size of the neural network is however related to the number of obstacles and resultant increase in complexity in the presence of more obstacles demonstrates it's feasibility is limited to a single obstacle. Delahaye et al. in [60] highlights the importance of the ability to gain altitude as an option for aerial vehicles. It stands to reason that a parallel NN based approach could be proposed to assess trajectory changes in the vertical dimension also, in order to portray the full landscape of trajectories available to modern UAV systems.

Several articles examine the possibility of hybrid systems involving NNs with independent *genetic algorithms* (GAs) as a means of assessing the fitness of a population of candidate trajectories [171,244]. GAs themselves have seen a rise in interest more recently for the purpose of UAV trajectory management. The principle is outlined well in [60,171], involving a population of candidate solutions that are assessed in accordance to a defined fitness function. A sample of the fittest solutions is taken forward and mutated via a set of classical operations to form the basis of the next population. This process is applied in the generation of trajectories for autonomous vehicles in [200] and collision free Cartesian path generation for manipulators in [2, 20]. The concept of "tournaments" being held between populations using a hybrid simulated annealing approach as a method of assessing fitness can be found in [1, 97, 226, 235, 263].

Delahaye et al. in [60], presents genetic algorithms in both the context of air-traffic collision prevention, but also for air-space partition design. This premise is formed from their proposed GA algorithm which proposes a lateral "sharing" process in addition to inter-generational tournament (via simulated annealing) to aid in the convergence and avoiding local minima. The proposed method assesses the proposed trajectory against an ideal trajectory subject to linearised separation constraints. The chromosome solutions are then assessed based on the implied terminal distance from the flight path, increase

in flight time and the summed control effort. While this method is able to represent the scenario through an appropriate model and constraint set, convergence on a solution is inherently slow due to the approach to the selection process. It is worth highlighting that the dynamics of an air traffic avoidance scenario are considerably slower than the sense and avoid problem.

5.6 Summary

It is clear from challenges posed by applications for coordinated unmanned systems that a hybrid systems composed of cooperative and non-cooperative methods of collision avoidance are highly desirable. A reliable non-cooperative approach is shown to be a fundamental component in establishing the autonomy necessary for independence in unknown environments where reliable communication cannot be guaranteed. Recent developments in sensing, tracking and object classification have made the possibility of decentralised and non-cooperative approaches more tangible for small-scale UAS.

In Table 5.2, a high level summary of the methods presented in Section 5.5 is provided. Here, each method is compared qualitatively and quantitatively with respect to several key parameters that define the nature of the avoidance problem addressed. While it is clear that several promising algorithms supporting an advancement towards dependable SDA exist in the literature, their tolerance to uncertainty in dynamic environments is below that necessary for establishing effective level of autonomy [107, 142].

One of the most promising areas of research identified in this review utilises a purely geometric description of the conflict (see Table 5.2). This minimalist approach has seen increasing interest recently in the form of the proposed HRVO and ORCA methods; due to their scalability and effectiveness in cluttered environments. In associated works [65,67], it is shown how their scalability quickly deteriorates in communication denied environments where the agents must instead rely on more realistic trajectory measurements. There is therefore a need to develop scalable obstacle avoidance approaches for coordinated systems that are both able to handle highly dynamic environments, but also tolerate the measurement uncertainty brought about the sourcing of trajectory data from local sensor arrays.

Table 5.2: *A comparison of the contributions from the literature most relevant to the objectives of these works. Here, the key parameters indicating the source of obstacle trajectory data and nature of the avoidance problem are given. A qualitative assessment of each method's scalability, a consideration for trajectory uncertainty is also presented [5, 95, 138].*

Approach Name	**Vehicle Model**	**Dimensions**	**Obstacle Number**	**Obstacle Data Source**	**Scalability**	**Uncertainty Consideration**
Protocol-based						
Finite Information Horizon [112]	Point	2	Open	Local	High	No
Separation Assurance baseline [130]	Point	2	Open	Local(ADS-B)	High	No
Conflict resolution protocol selection [30]	Dynamic	3	Pairwise	Global	Medium	No
Localised conflict probability [183]	Point	3	Pairwise	Global	Medium	Yes
Geometry-based						
Collision cone clustering [54]	Point	2	Open	Global	High	No
Prediction-free reachable sets [261]	Point	2	Open	Local	Medium	No
Generalised Velocity Obstacle [26, 259]	Point, Simple Car	2	Open	Global	High	No
Hybrid-Reciprocal Velocity Obstacle [27]	Differential Drive	2	Open	Global	Medium	No
Optimal Reciprocal Avoidance [148, 251]	Point	2	Open	Global	High	No
Non-linear Velocity Obstacle [219]	Point	2	Open	Global	Medium	No
3D velocity obstacles [38, 48]	Point	3	Pairwise	Global	Low	No

Approach Name	Vehicle Model	Dimensions	Obstacle Number	Obstacle Data Source	Scalability	Uncertainty Consideration
Potential Field-based						
COSMOS artificial potential field [131]	Dynamic	3	Open	Global	High	No
Predictive Potential Field [172]	Dynamic	2	Open	Global	High	No
3D Bifurcating Fields [240]	Dynamic	3	Open	Global	High	No
Game Theory-based						
Hybrid optimal control/game theory [246]	Point, Dynamic	2	Pairwise	Sensed	Low	No
Differential game trajectory control [248]	Point	3	Open	Global	Medium	No
Reachable set continuous dynamic games [164]	Dynamic	3	Pairwise	Global	Low	Yes
Probabilistic pursuit-evasion game [252]	Dynamic	2	Open	Sensed	Medium	No
Optimisation-based						
Dynamic velocity window [86]	Simple Car, Dynamic	2	Open	Sensed	Medium	No
Dynamic programming [107]	Dynamic	3	Open	Local(ADS-B)	Medium	No
Probability collective optimisation [189, 229]	Point	2	Open	Global	Low	No
Biologically inspired						
ATC through Genetic Algorithms [60]	Dynamic	2	Pairwise	Global	Low	No
NN Occupancy Grid [97]	Dynamic	2	Open	Sensed	Low	Yes
NN learned by supporting GA [171, 244]	Point	2	Pairwise	Local	Low	No

Chapter 6

Geometric Obstacle Avoidance

The notion of *sense detect and avoidance* (SDA) is well established within the literature. One of the most promising, and intuitive, branches of SDA techniques are those based entirely on deductions from the obstacle's geometry and pose. In associated works [65–67], several of the most popular methods of geometric avoidance are presented in the form of a technical review. Some of the challenges these algorithms face are used to provide context for the method *interval avoidance* (IA), presented initially in [63]. In this section, each of these geometric algorithms are presented, in addition to the assumptions and findings of these associated works.

6.1 Problem Description

Consider the scenario where two *micro aerial vehicles* (MAVs), or agents referred to as i and j, are moving at constant altitude defined by a plane XY. The velocities of the two MAVs are denoted as $\boldsymbol{v}_i \in \mathbb{R}^{2\times 1}$ and $\boldsymbol{v}_j \in \mathbb{R}^{2\times 1}$ respectively and are assumed to be moving in accordance to some global objective. The agents are defined to have a characteristic radius r_i and r_j which is assumed to contain the physical extents of i and j respectively (see Figures 6.1(a) and 6.1(b)). From the perspective of i, j is an obstacle to be avoided. If both agent trajectories are maintained, a collision may occur at a certain time in the future.

6.1.1 Sensor Model

It is assumed that an obstacle's position and width can be measured by the MAV's on-board camera and range-finder [172]. The pixel location of the obstacle provides measurements of the relative heading angle $\tilde{\psi}_j = \mathcal{N}(\psi_j, \sigma_c^2) \in [-\pi, \pi]$ and angular width $\tilde{\alpha}_j = \mathcal{N}(\alpha_j, \sigma_c^2) \in [-\pi, \pi]$. The range-finder is assumed able make instantaneous prox-

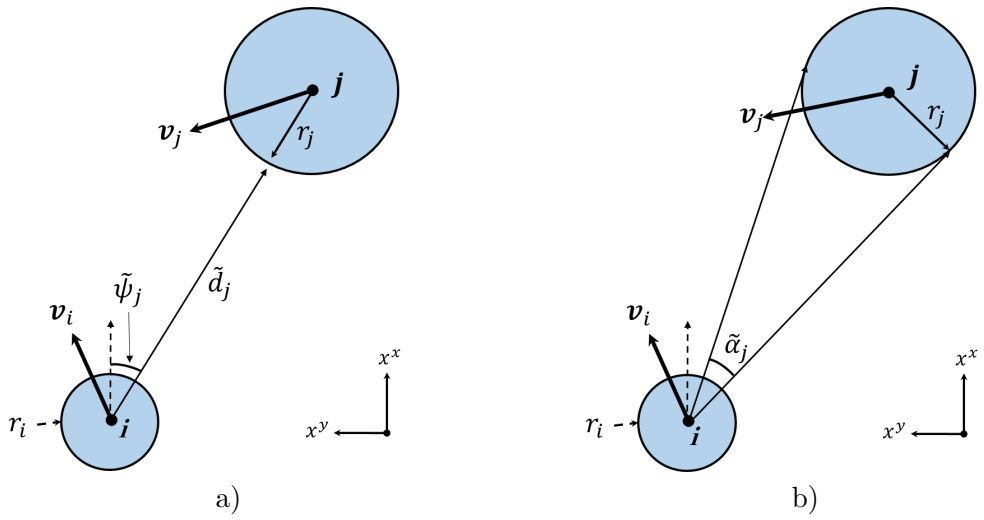


Figure 6.1: *a) Agent i observing the trajectory of agent j and it's planar position and velocity $\boldsymbol{p}_j$ and $\boldsymbol{v}_j$ respectively. b) Definition of the angular width measurement α_j characterising the obstacle's radius r_j.*

imity measurements $\tilde{d}_j = \mathcal{N}(d_j, \sigma_r^2) \in [0, d_{max}]$. Each sensor is known to have it's own uncertainties σ_c and σ_r that corrupts the measurements and is assumed to be Gaussian with a zero mean. More information about sensor emulation in OpenMAS can be found in Section 2.1.3 and in associated works [66].

The sensor measurement transformation is defined in Equation (6.1) in order to express $\boldsymbol{p}_j$ in the coordinate axes of i relative to the body axes origin O_b. The obstacle's equivalent Cartesian position $\boldsymbol{p}_{k,j}$ at the time step t_k, given the sensor's relative position $\boldsymbol{p}_{i,sensor}$ and orientation $\mathbf{R}_{i,sensor}$, can then be written:

$$\boldsymbol{p}_{k,j} = \begin{bmatrix} p^x_{k,j} \\ p^y_{k,j} \end{bmatrix} = \mathbf{R}^T_{i,sensor} \cdot \left(\begin{bmatrix} \cos(\tilde{\psi}_{k,j}) & -\sin(\tilde{\psi}_{k,j}) \\ \sin(\tilde{\psi}_{k,j}) & \cos(\tilde{\psi}_{k,j}) \end{bmatrix} \cdot \tilde{d}_{k,j} \right) - \boldsymbol{p}_{i,sensor} \tag{6.1}$$

The obstacle's Cartesian velocity is then calculated from the obstacle's change in position $\boldsymbol{v}_{k,j} = \frac{1}{\Delta t}(\boldsymbol{p}_{k,j} - \boldsymbol{p}_{k-1,j})$ where $\Delta t = t_k - t_{k-1}$. The measurement of obstacle j's angular width α_j at t_k can similarly be used to determine the obstacle's representative radius r_j. This is calculated from the geometric parameters shown in Figure 6.1(b) and the relationship:

$$\begin{aligned} \sin\left(\frac{1}{2}\tilde{\alpha}_{k,j}\right) &= \frac{r_{k,j}}{\tilde{d}_{k,j} + r_{k,j}} \\ \tilde{d}_{k,j} \cdot \sin\left(\frac{1}{2}\tilde{\alpha}_{k,j}\right) &= \left(1 - \sin\left(\frac{1}{2}\tilde{\alpha}_{k,j}\right)\right) \cdot r_{k,j} \\ r_{k,j} &= \frac{sin\left(\frac{1}{2}\tilde{\alpha}_{k,j}\right)}{1 - \sin\left(\frac{1}{2}\tilde{\alpha}_{k,j}\right)} \cdot \tilde{d}_{k,j} \end{aligned} \tag{6.2}$$

From Equations (6.1) and (6.2) the current knowledge of i regarding j is represented by it's position, velocity and radius; $\boldsymbol{p}_{k,j}$, $\boldsymbol{v}_{k,j}$ and $r_{k,j}$ respectively in it's local coordinate axes. Agents are assumed able to observe all obstacles within a defined radius d_{nei} and retain memory of $\boldsymbol{p}_{k-1,j}$.

6.1.2 Neighbourhood Consideration

In the multi-agent literature the concept of a neighbourhood is often employed to limit the visibility of the population. The function of a neighbourhood is shown to be critical in limiting the number of constraints, and therefore complexity, of the problem in [250,259]. High numbers of constraints are typical in a cluttered environment and as a result, the solution space may become saturated leading to deadlock events [67].

In some cases it may be useful to draw distinction between the sensoral horizon d_{max} and a local neighbourhood d_{nei}. This may simply be to create dependent behaviours for surveillance or monitoring but also for collision avoidance when d_{nei} is reached. In these works it is assumed that each agent begins collecting data when the object enters d_{max}, but will not instigate an avoidance response until it enters a local neighbourhood d_{nei} [66,67]. However, it is often assumed that $d_{nei} = d_{max}$ in the SDA literature, as by definition a response is to be generated immediately in such encounters.

6.1.3 Flight Paths and Objectives

It is assumed that all agents are operating towards some common objective in a communal airspace. In these works, flight paths are defined as a sequence of waypoints $W_i \in [W_1, W_2, ..., W_n]$ between which collisions may occur [278]. OpenMAS innately assigns each waypoint a 3D global state $\boldsymbol{\chi}_{i,\text{wp}}$ with a designated a priority indicating the order in which they must be achieved [66]. This is intended to emulate a set of *non-directional beacons* (NDB) positioned within the airspace similar to conventional air-traffic scenarios. For agents operating within a defined plane, the planar projection of W_i is used to define $\boldsymbol{p}_{i,\text{wp}}$ in the axes of agent i:

$$\boldsymbol{p}_{i,\text{wp}}^{xy} = \boldsymbol{p}_{i,\text{wp}} - \frac{\boldsymbol{p}_{i,\text{wp}} \cdot \boldsymbol{n}^{xy}}{||\boldsymbol{n}^{xy}||^2} \boldsymbol{n}^{xy} \tag{6.3}$$

Here $\boldsymbol{n}^{xy}$ denotes the planar normal aligned with the agent's z axis. Once the waypoint condition is satisfied (see Section 2.1.2), the target waypoint W_i is immediately reallocated as the agent moves into the next segment of the objective. The preferred velocity of agent

i, in terms of it's nominal speed v_{pref}, is then determined by Equation 6.4.

$$\boldsymbol{v}_{i,\text{pref}} = \frac{\boldsymbol{p}_{i,\text{wp}} - \boldsymbol{p}_i}{||\boldsymbol{p}_{i,\text{wp}} - \boldsymbol{p}_i||} \cdot v_{\text{pref}} \tag{6.4}$$

For the purposes of evaluation, it is important to ensure that a collision will exist without intervention. In such cases, the flight path of agent i is designed to induce collision by specific placement of it's waypoints.In accordance with the SDA concept, the global position of the target waypoint is presented locally to the agent by transforming into it's body axes via Equation (6.1). Here $\mathbf{R}_i$ defines the rotation matrix of agent i. At all times the position of agent i's waypoint $\boldsymbol{p}_{i,\text{wp}}$ is assumed observable to agent i from it's current position $\boldsymbol{p}_i$ [278]. The agent's preferred velocity of $\boldsymbol{v}_{i,\text{pref}}$ is then defined in Equation (6.4); as a vector directed toward the target waypoint position $\boldsymbol{p}_{i,\text{wp}}$.

6.2 Implementation of the State of the Art

With an agent able to track the motion of obstacles through their local visual field, each agent is tasked with computing viable trajectories to avoid collision. In the literature, there are numerous techniques for defining viable trajectories based on constraints formed geometrically. One of the most popular examples of this is the *velocity obstacle* which is discussed alongside several similar techniques in [65, 67] as part of a comprehensive technical survey of the state of the art.

6.2.1 The Velocity Obstacle

The *velocity obstacle* (VO) concept is based on the geometric assembly of the *collision cone* (CC); initially presented in [82]. Obstacles are observed in the agent's local horizontal plane (XY) with their planar cross-section centred at $\boldsymbol{p}_j$ as seen in Figure 6.2. Here, the collision cone for obstacle j is defined as CC_j from the geometric properties of the obstacle's relative position $\boldsymbol{p}_j$, configuration radius r_c and velocity $\boldsymbol{v}_j$ [65, 67].

Velocities that will bring about a collision with obstacle j are then represented in the velocity space by translating CC_j by $\boldsymbol{v}_j$ via the Minkowski sum: $\text{VO}_j = \text{CC}_j \oplus \boldsymbol{v}_j$. In the consideration of multiple obstacles, the union of multiple $\text{VO}_{1:n}$ is taken. Agent velocities are therefore considered valid if $\boldsymbol{v}_{k+1,i} \notin \text{VO}_k = \cup_{j=1}^{n} \text{VO}_{k,j}$ [82]. Velocities satisfying this constraint describe a collision free trajectory for agent i in the presence of obstacles $\text{VO}_{j=1:n}$ for time t_k.

In practice, oscillatory trajectories are often observed in instances where two agents

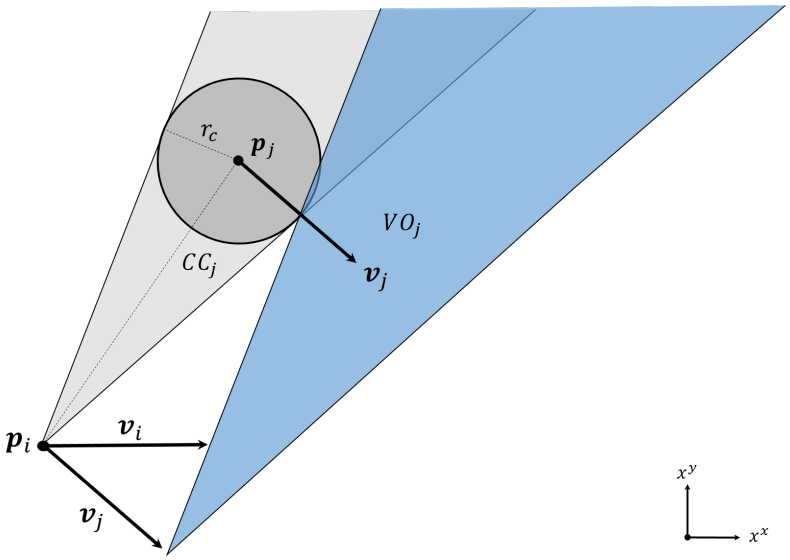


Figure 6.2: *The VO_j (blue) from the initial CC_j (grey) corresponding to obstacle j. Here the VO_j is defined in the configuration space of i, in terms of the relative position $\boldsymbol{p}_j$ of obstacle j, configuration radius $r_c = r_i + r_j$ and velocity $\boldsymbol{v}_j$.*

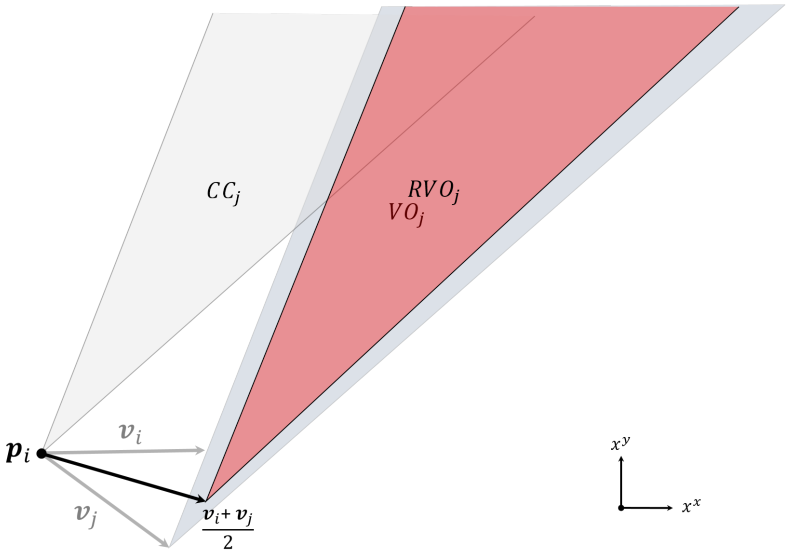


Figure 6.3: *The construction of RVO_j (red) from its apex positioned at the average of $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$. Its relation to the original collision cone CC_j and VO_j is also expressed geometrically.*

attempt to resolve a conflict with one another using the VO method. This often propagates until the point of collision occurs; as the two agents repeatedly resolve velocities $\boldsymbol{v}_{i,k+1}$ that imply a new conflict at t_{k+1}. Obstacles that are static, or moving with constant velocity can otherwise be handled using the VO approach [26].

6.2.2 The Reciprocal Velocity Obstacle

An iteration of the conventional VO method known as the *reciprocal velocity obstacle* (RVO) attempts to consider the reciprocal motion of the second decision making agent j in order to produce smoother avoidance trajectories [26]. The agent generates a VO with an apex augmented by the average of the two object velocities $\boldsymbol{v}_{k+1,i} \notin \mathrm{CC}_j \oplus (\boldsymbol{v}_{k,i} + \boldsymbol{v}_{k,j})/2$. This can be seen in the placement of the RVO apex in Figure 6.3. This concept effectively allows the agent to mediate it's correction trajectory $\boldsymbol{v}_{k+1,i}$ in accordance with $\boldsymbol{v}_j$. At time t_k, the RVO represents the region of velocities that are the average of both the velocity of agent i and the velocity of obstacle j.

The RVO is shown to eliminate the VO oscillation mentioned in Section 6.2.1, and the resultant resolution trajectories are seen to be smoother [26]. While this is the case, agent i and obstacle j do not explicitly agree on which sides they will approach each other. This can lead to scenarios where agents will mirror the trajectories of their respective obstacles in an attempt to avoid them. The oscillations induced by this behaviour, distinct from those of the VO, are often referred to as a *reciprocal dance* [65, 67].

6.2.3 The Hybrid Reciprocal Velocity Obstacle

An advancement on the VO problem has been proposed to negate the causes of reciprocal dance by augmenting the VO and RVO regions. The *hybrid-reciprocal velocity obstacle* (HVRO), shown in Figure 6.4, alters the apex of the HRVO in order to create differential behaviour depending on the relative motion of the obstacle $\boldsymbol{v}_j$.

The centre-line of VO_j and RVO_j are co-linear in nature, therefore if the obstacle is moving right, the agent should resolve a trajectory $\boldsymbol{v}_{i,k+1}$ to pass the obstacle on the left and vice-versa. Failure to do so brings about the phenomena of the reciprocal dance. Although the method is shown to improve the generation of smooth avoidance trajectories, it cannot guarantee it theoretically [27]. In the example given in Figure 6.4, directional bias is established by adjusting the apex of the HRVO_j to be the intersection of the leading edge of RVO_j the trailing edge of VO_j (i.e. $\mathrm{HRVO}_j = \mathrm{CC}_j \oplus \boldsymbol{v}_{\mathrm{HRVO}}$. The resulting constraint set imposed upon agent i at time t_k is then written $\boldsymbol{v}_{i,k+1} \notin \mathrm{HRVO}_k = \cup_{j=1}^{n} \mathrm{HRVO}_{i,k}$ [27].

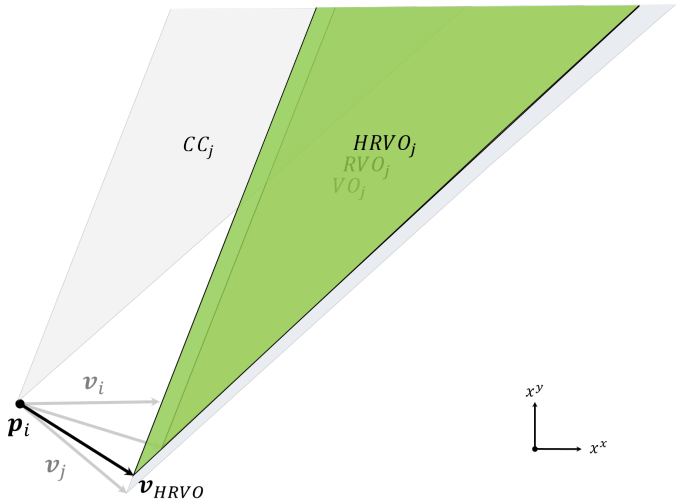


Figure 6.4: *The definition of the HVRO (green) as the extension of the VO, RVO and CC (grey) concept. The geometry of the HRVO is defined by relative position and velocity of obstacle j. The HRVO's apex position $\boldsymbol{v}_{HRVO}$ is defined in accordance to the direction of $\boldsymbol{v}_i$ and encodes an appropriate direction of pass for obstacle j using from components of the VO an RVO.*

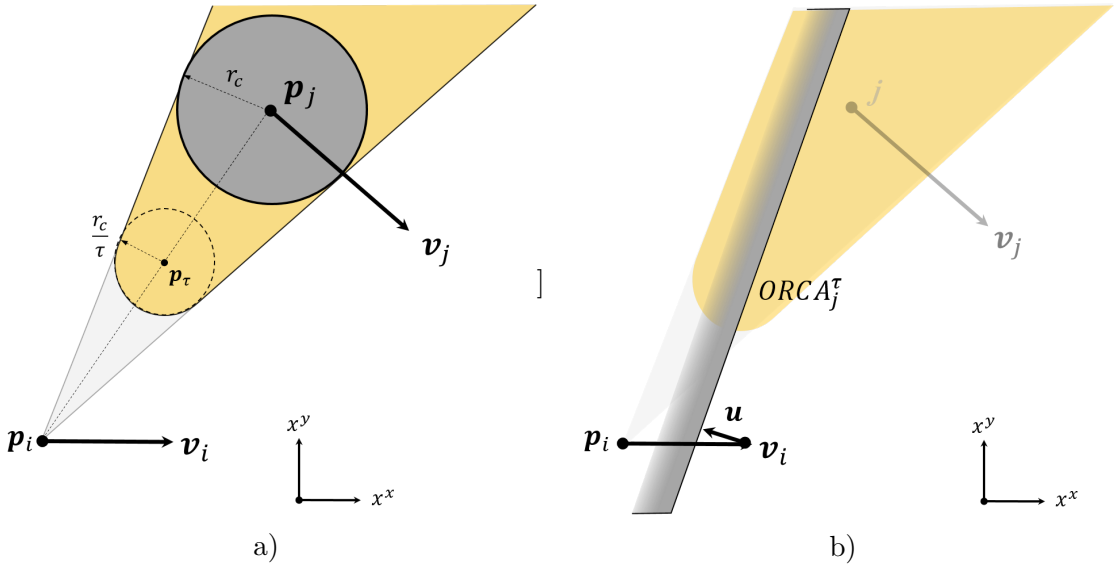


Figure 6.5: *a) The geometric description of the truncated VO of obstacle j (shown in yellow) defined by the truncation parameter τ, relative position ($\boldsymbol{p}_j$) and configuration radius $r_c = r_i + r_j$. b) The assembled ORCA line $ORCA_j^\tau$ (grey half-plane) and associated velocity correction $\boldsymbol{u}$ as a result of obstacle j are shown.*

Typically the RVO and HRVO are only necessary in the computation of inter-agent avoidance trajectories. The global VO set for agent i can instead be written as the union of the reciprocal variants (RVO or HRVO) for surrounding agents A_j and the VO for obstacles O_j: $\boldsymbol{v}_{i,k} \notin \text{HRVO}_k = \bigcup_{A_j=1}^{n} \text{HRVO}_{A_j} \cup \bigcup_{O_j=1}^{n} \text{VO}_{O_j}$ [65, 67].

6.2.4 Optimal Reciprocal Collision Avoidance

A more recent technique that employs the concepts of the RVO technique is referred to as *optimal reciprocal collision avoidance* (ORCA). The ORCA approach is described well in [250], demonstrating how the ORCA velocity obstacle is formulated for a given reciprocally collision avoiding agent pair i and j. The resultant trajectory is not only smooth but, for small time steps, can be seen as continuous in the velocity space. The truncation parameter τ represents the time window for which a collision free trajectory should be guaranteed, i.e the agent can move at it's new velocity for τ seconds.

If it is assumed that $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ are those that will bring about a collision in the future, then $\boldsymbol{u}$ is the vector to the point closest to the boundary of VO_j: $\boldsymbol{u} = (\arg\min_{\boldsymbol{v} \in \delta \text{VO}^\tau} ||\boldsymbol{v} - (\boldsymbol{v}_i - \boldsymbol{v}_j)||) - (\boldsymbol{v}_i - \boldsymbol{v}_j)$ (see Figure 6.5(b)). Here $||\boldsymbol{v}||$ denotes the Euclidean norm of $\boldsymbol{v}$. Using the "outward" facing normal $\boldsymbol{n}$ of the boundary at the point $(\boldsymbol{v}_i - \boldsymbol{v}_j) + \boldsymbol{u}$ and the assumption that the responsibility that the avoidance is shared equally, the formulation for the ORCA_j constraint can be written as $\text{ORCA}_k^\tau = \boldsymbol{v} | \boldsymbol{v} - (\boldsymbol{v}_i + \frac{1}{2}\boldsymbol{u}).\boldsymbol{n} \geq 0$. The geometric representation of $\boldsymbol{v}$ is given in Figure 6.5(b) and Figure 6.5(a). Here it is represented as a "half-plane" with normal $\boldsymbol{n}$, with the initial point at $\boldsymbol{p} = \boldsymbol{v}_i + \frac{1}{2}\boldsymbol{u}$ [250].

The ORCA lines themselves allow the scenario to be described using only linear constraints. In addition, representation of the RVO as half-planes allows for simplification of the constraint set by eliminating those already covered by other ORCA lines, whilst guaranteeing continuously smooth agent trajectories [65, 67].

6.2.5 Trajectory Selection

With the constraint sets defined in Sections (6.2.1-6.2.4), an applicable avoidance velocity must be determined to be implemented at t_{k+1}. The definition of optimality is known to be subject to strategy in the literature [82] and is typically associated with "control effort". This selection criteria is represented, considering the minimum deviation from a desired trajectory $\boldsymbol{v}_{i,\text{pref}}$; subject to the union of the VO_k set. In such cases the optimal velocity can then be expressed as $\boldsymbol{v}_i^* = \arg\min_{\boldsymbol{v} \notin VO}(||\boldsymbol{v} - \boldsymbol{v}_{i,\text{pref}}||)$. In the proceeding Sections the *clear path* method is used to extract candidate escape trajectories from each constraint set. [67, 102, 250].

6.3 Agent Dynamics & Control

The SDA problem is initially posed such that motion of the agents(MAVs) $A_{i=1:n}$ is restricted to a common plane. In previous sections it is demonstrated how viable avoidance trajectories are attained, given the set of obstacles $O_{j=1:n}$ observed at time t_k. How the commanded velocity $\boldsymbol{v}_{i,\text{pref}}$ is typically achieved is subject to the dynamics of the individual agent. The representation of object dynamics and their integration with OpenMAS are discussed in more detail in Section 2.2.3 and in accompanying works [66].

6.3.1 Euler Heading

It is assumed that the agent is enacting control inputs in response to observations made locally. As result of a new commanded velocity $\boldsymbol{v}_{i,\text{pref}}$, agent i computes its required forward speed $v_i = ||\boldsymbol{v}_i||$ and angular rate $\omega_{i,k} = \frac{1}{\Delta t}(\psi_{i,pref} - \psi_{i,k})$. The equivalent Euler heading $\psi_{i,pref}$ is defined by vector decomposition of $\boldsymbol{v}_{i,\text{pref}}$ in Equation (6.5 and 6.6):

$$\mu = det(\hat{\boldsymbol{n}}^x, \boldsymbol{v}_{i,\text{pref}}) \tag{6.5}$$

$$\psi_{i,pref} = \frac{\mu}{||\mu||} cos^{-1}\left(\frac{\hat{\boldsymbol{n}}^x \cdot \boldsymbol{v}_{i,\text{pref}}}{||\hat{\boldsymbol{n}}^x||}\right) \tag{6.6}$$

Here $\hat{\boldsymbol{n}}^x$ defines a unit vector along the local x-axis of agent i. The parameter μ is the determinant of a square matrix composed of vectors $\hat{\boldsymbol{n}}^x$ and $\boldsymbol{v}_{i,\text{pref}}$.

6.3.2 Single Integrator Systems

Within the collision avoidance and multi-agent systems literature, the use of *single integrator* models to approximate agent behaviour is widely cited [67, 276–278]. A system is said to be a single integrator when $\boldsymbol{u}_{k,i} = \dot{\boldsymbol{x}}_{k,i}$. In this chapter, the single integrator is derived from the planar pose agent i at t_k where $\boldsymbol{x}_{k,i} = [x_{k,i}, y_{k,i}, \psi_{k,i}]^T$ and $\boldsymbol{u}_{k,i} = [\dot{x}_i, \dot{y}_i, \omega_i]^T$. The discrete representation of a system under this assumption then takes the form $\boldsymbol{x}_{k+1,i} = \boldsymbol{x}_{k,i} + \Delta t \cdot \boldsymbol{u}_{k,i}$.

In practice v_i and ω_i are bounded between some defined maximum in order to represent actuator constraints (see Table 6.1). Characterising agent behaviour in this way allows for the agent's true dynamics to abstracted to descriptor function $\dot{\boldsymbol{x}}_{k,i} = d(\boldsymbol{x}_{k,i}, \boldsymbol{u}_{k,i}, \boldsymbol{w}_{k,i})$ as seen in Section 2.2.3. An high-level overview of the computation loop of agent i can be seen shown in Algorithm 1. In each of the studies presented in Section 6.4, a selected geometric avoidance method is used to compute the VO constraint set.

Algorithm 1: A high-level overview of several of the popular geometric methods for collision avoidance. The agent generates a constraint set corresponding to the to each obstacle $\mathbf{O}_{j=1:n}$ using the methods introduced. Once resolved, the feasible velocity is passed to agent controller to determine the agent's state update $\boldsymbol{x}_{k+1,i}$.

Data: $\boldsymbol{x}_{k,i}$,$\mathbf{W}_j$,$\mathbf{O}_j$
Result: $\dot{\boldsymbol{x}}_{k,i}$,$\boldsymbol{v}_{\text{pref}}$
`// Read the agent's local state.`
$\boldsymbol{p}_i, \boldsymbol{v}_i, r_i$ = GetAgentState()
`// Measure the new obstacle states.`
$\mathbf{W}_j$,$\mathbf{O}_j$ = ReadSensorBuffer()
`// Calculate waypoint(desired) trajectory.`
$\boldsymbol{v}_{\text{pref}}$ = GetWaypointPath(W_i)
for *j=1:length($\boldsymbol{O}_j$)* **do**
  `// Get the obstacle states.`
  $\boldsymbol{p}_j, \boldsymbol{v}_j, r_j$ = GetObstacleTrajectory($\mathbf{O}_j$)
  `// Compute constraint representation`
  VO_j = ComputeVOConstraint($\boldsymbol{p}_i, \boldsymbol{v}_i, r_i, \boldsymbol{p}_j, \boldsymbol{v}_j, r_j$)
  `// Store constraint region`
end
`// Parse the avoidance trajectory`
$\boldsymbol{v}_{\text{pref}}$ = ComputeSelectionStrategy($\boldsymbol{v}_{\text{pref}}$,$\text{VO}_j$)
`// Compute the inputs`
$\omega_{i,k}$, $v_{i,k}$ = CalculateControlInputs($\boldsymbol{v}_{\text{pref}}$)
`// Pass the inputs to the controller`
$\dot{\boldsymbol{x}}_{k,i}$ = AgentController($\boldsymbol{x}_{k,i}$,$\omega_{i,k}$, $v_{i,k}$)

Parameter	Value	Unit's
Maximum speed (v_{max})	4	$m \cdot s^{-1}$
Preferred speed (v_{pref})	2	$m \cdot s^{-1}$
Maximum yaw rate ($\dot{\psi}_{max}$)	0.25	$rad \cdot s^{-1}$
Agent critical radius (r_i)	0.5	m
Neighbour horizon (d_{nei})	15	m
Camera standard deviation (σ_α)	5.208×10^{-5}	rad
Range-finder standard deviation (σ_r)	0.5	m
Airspeed standard deviation (σ_s)	0.5	$m \cdot s^{-1}$
Position standard deviation (σ_p)	0.5	m
Agent orbital radius	10	m
waypoint orbital radius	20	m
Cycles	1000	-
Sampling rate (Δt)	0.25	s
Waypoint & collision tolerance (ι)	1×10^{-3}	m

Table 6.1: *The assumptions and agent parameters used in the following example scenarios, including the sensor uncertainties used in the representative sensing condition.*

6.4 Performance Evaluation

In this section the presented methods are compared in a series of conflicts representing real-world scenarios. The agent population is initialised with the parameters defined in Table 6.1, representing a concentric multi-agent collision. Concentric scenarios are well established within the collision avoidance literature as they represent the worst case scenario for a multi-agent conflict [27, 63, 67, 94, 111, 112, 117, 250]. This is due to the fact that in this condition all agents are initialised with a zero miss-distance that ensures a collision will occur without intervention. By distributing the agent set radially about the collision centre, a scalable mechanism for evaluating the effectiveness of the presented algorithms is established.

6.4.1 Experimental Conditions

Agents are designated a target waypoint at the antipodal position of a concentric circle with a radius of $20m$. The agents are tasked with crossing the circle to reach their waypoint positions $\boldsymbol{p}_{i,\text{wp}}$ whilst ensuring their separation does not violate the collision condition. In addition to this, measurements made by each agent at time t_k are are assumed to be corrupted by the noise parameters given in Table 6.1. These are applied in order to better represent sensor-derived measurement uncertainty of both the agent and the obstacle's trajectory. In Figure 6.6(a) the agents are initialised at their origins (circles) and move through the collision centre to reach their respective waypoints (triangles). Events such as collisions or waypoint incidence are said to occur in accordance to the conditions specified

in Section 2.1.2. The agent and scenario parameters used in the following examples are otherwise explicitly stated in Table 6.1.

6.4.2 Performance Evaluation

The selected algorithms presented in Section 6.2 are validated over several scenarios with increasing agent populations. Of these scenarios the ten agent scenario is presented and discussed to highlight and contrast the performance of each algorithm. Figure 6.6(a) demonstrates the trajectories generated by the VO algorithm. When compared to the RVO in Figure 6.6(b) the trajectory adjustments can be seen to be abrupt, with greater oscillation throughout, until all conflicts are resolved.

The compensation for obstacle movement is clearly seen in Figure 6.6(b) under the RVO method as the trajectories are shown more gradual with fewer instances of harsh correction. Oscillation in the form of reciprocal dance can still be observed however as the direction of pass is resolved. In comparing the RVO trajectories to those of the HRVO in Figure 6.6(c), there is a clear reduction in the oscillation as the agents initially determine their direction of pass. The HRVO directional bias can also be observed from the agent trajectories, indicated by the emergent spiral behaviour around the conflict centre.

The representation of the VO as ORCA constraints is shown to produce trajectories similar to that of the HRVO in Figure 6.6(c). The linearity of the constraints however is shown to create smooth trajectories throughout the conflict scenario, resulting in smaller overall course deviations. The selected algorithms were demonstrated in scenarios with two, five, ten and twenty agents and their performance measured over one thousand Monte Carlo independent iterations. In addition to this, two sensor conditions were observed; A) Ideal Sensing: the agents are given perfect knowledge of the surrounding obstacles B) Representative Sensing: the agents adopt more realistic sensor properties, which are defined in Table 6.1.

The mean behaviour of the presented approaches are shown in Table 6.2, where a clear difference can be seen between the *ideal* and *representative* sensing conditions during the ten agent example scenario. Under the assumptions of ideal obstacle telemetry, the compensative nature of the RVO is shown to reduce the mean number of collisions to 3.140. This is a significant reduction from the mean of 9.203 collisions in the same scenario using the original VO method. The encoding of directional information in the formation of the HRVO is shown to result in a greater ability to maintain safe separation when compared to the VO and RVO methods in Table 6.2. This is reflected in the number of collisions being reduced further to an average of 0.053 collisions. The lowest mean collisions however, was

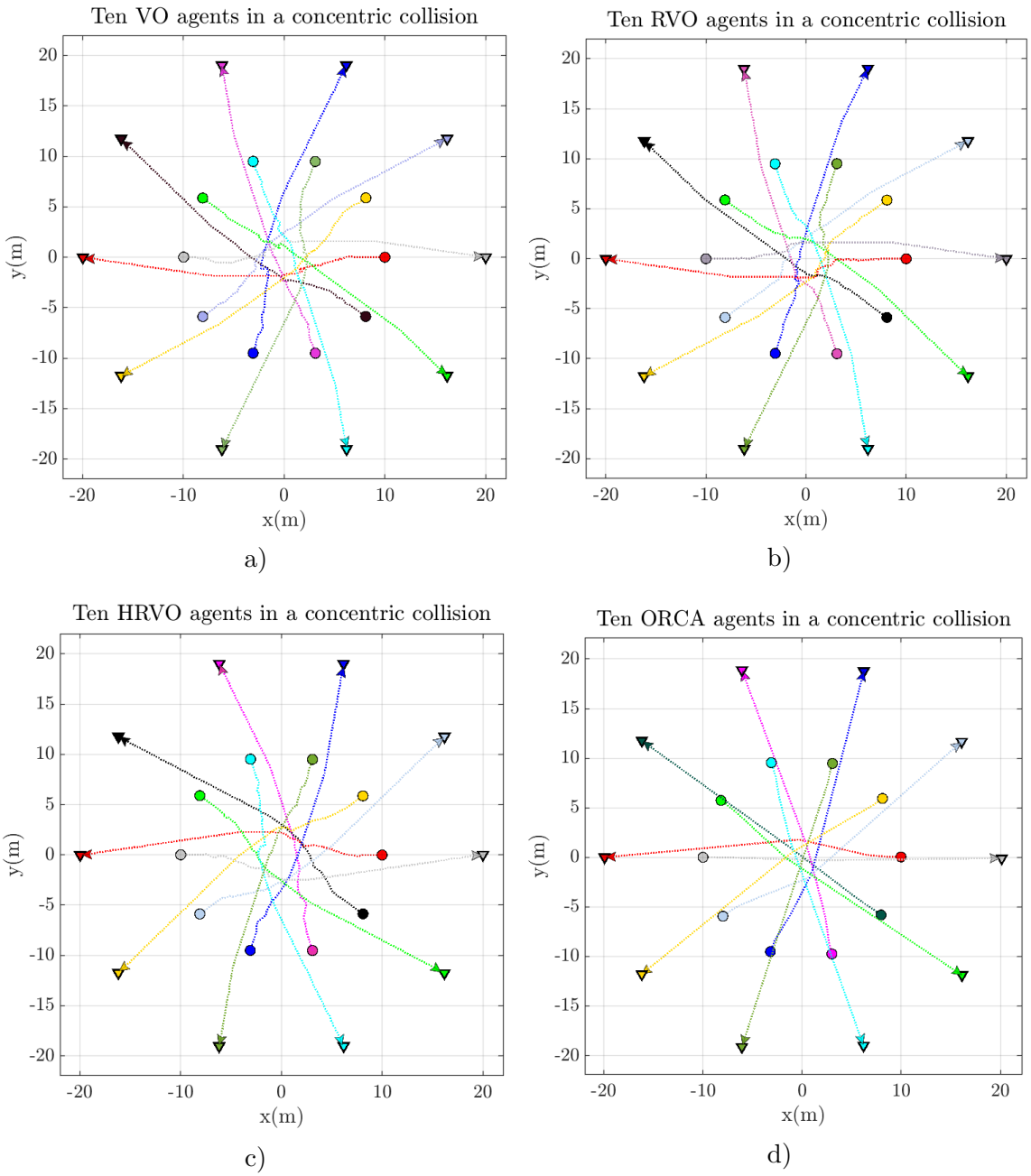


Figure 6.6: *a) The resulting trajectories of ten agents using VO-based reactive avoidance in a concentric collision scenario. The oscillations due to obstacle compensative motion can be clearly observed as the agents progress towards the collision centre. b) The ten agent concentric scenario applying the RVO-based avoidance method. Abrupt trajectory changes can be observed, with distinct oscillations as novel agent j enter the visual horizon of agent i. c) The ten agent concentric scenario repeated with the HRVO obstacle generation method applied. Trajectory oscillations can be observed initially, however as the direction of pass is resolved linearity can be observed. d) The same ten agent scenario repeated under the ORCA obstacle generation method. The resultant trajectories appear as smoother, more gradual adjustments than the previous methods.*

Algorithm Condition	Mean Collisions	Mean Minimum Separation (m)	Mean Computation Time (ms)
Condition A			
VO	9.203	0.581	2.000
RVO	3.140	0.831	2.100
HRVO	0.053	0.996	2.400
ORCA	0.038	1.000	0.460
Condition B			
VO	7.749	0.624	2.000
RVO	9.380	0.577	2.100
HRVO	2.878	0.836	2.600
ORCA	6.881	0.757	0.463

Table 6.2: *The performance of the presented algorithms in the same ten agent scenario. In condition A, the benchmark performance of the VO derivatives can be seen when sensing capabilities are assumed ideal. In condition B, where realistic sensor conditions are assumed, the performance of all methods with exception of the VO method is shown to reduce. The most resilient methods are shown to be the HRVO and ORCA algorithms with the HRVO being the most effective at minimising collision in the presented scenarios. Each value represents the mean across one thousand independent Monte Carlo iterations.*

found using the ORCA method; averaging 0.038 collisions over the one thousand cycle analysis.

Observing the behaviour of the algorithms in the presence of sensor uncertainty demonstrated a mean increase in computation time of $0.051ms$. This can be seen more clearly in Figure 6.7. A disadvantage of the RVO method is shown here by a factor of three increase in the mean number of collisions across one thousand iterations. This may be due the aggravation of the reciprocal corrections (reciprocal dance) by the uncertainty in obstacle trajectory. Similar behaviour can also be observed for the ORCA algorithm, as the sensor uncertainty is shown to significantly reduce it's effectiveness under this regime. The mean minimum separation achieved by the ORCA approach was shown to be the closest to the $1m$ boundary condition. This suggests a clear benefit of the ORCA method - it's consistency in achieving safe separation in ideal conditions. Although, considering uncertainty resulted in a mean increase of 4.003 collisions over the HRVO approach that demonstrated similar performance in condition A. Studying Figure 6.7, a square relationship can be observed between the agent population and the mean algorithm computation time for the VO, RVO and HRVO methods. The ORCA approach however, with it's linear representation of the constraint set, is shown to yield computation times that scale linearly with increasing agent numbers. The relationship between the performance reduction rate $r_{ORCA} = 3.4 \times 10^{-5} s/n$ is shown to be distinctly lower than the other presented approaches. The ORCA algorithm therefore has a clear advantage when con-

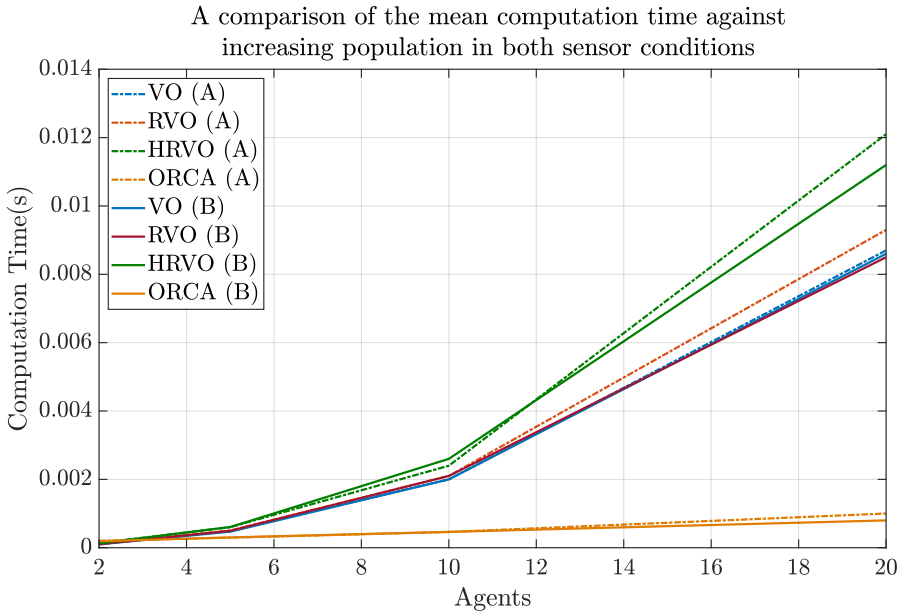


Figure 6.7: *A comparison of the mean computation times in sensing conditions A and B. A) with ideal obstacle knowledge is assumed, each method is shown scale more effectively with increasing number. B) When the obstacle data is subject to interference, the computation time of each method is shown to increase with respect to condition A, but also in the presence of an increasing agent population.*

sidering scalability for larger multi-agent systems, albeit more susceptible to uncertainty than the HRVO. All analyses were completed using an Intel Core i7-6600HQ quad-core (@2.8GHz) CPU. Code for the presented algorithms and scenarios are available on Github [65]. The relation between the agent density and the number of collisions is shown in Figure 6.8. As expected, the addition of obstacle uncertainty is shown to generally induce a higher rate of collision across the presented methods. This is with the exception of the original VO method; where the method is shown to be more effective with uncertainty. Methods considering both the velocity of i and j in the design of their constraints are shown to be more adversely affected by sensor noise. The HRVO and ORCA methods are shown to be the most effective methods of avoiding collision despite the ORCA method demonstrating higher sensitivity to sensor uncertainty than the HRVO approach.

6.4.3 A Problem of Symmetry

In collision scenarios involving more than two agents, there exists a problem of symmetry. While unlikely to occur in real systems, the situation may occur where an agent is presented with a constraint set that is symmetric about the forward direction $\boldsymbol{v}_i$, as seen in Figure 6.9. The agent will naturally choose a velocity minimising the separation with the waypoint $\mathbf{W}_i$ under the strategy imposed in Section 6.2.5. Any velocity that acts

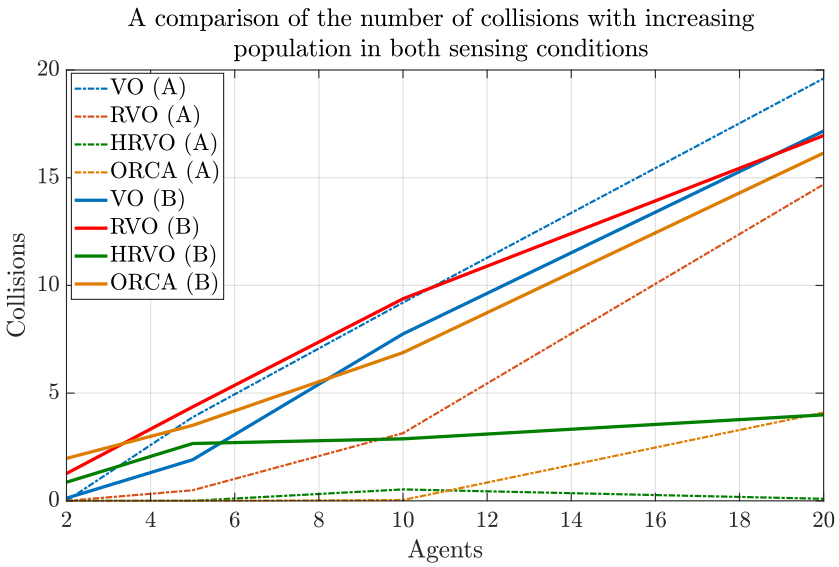


Figure 6.8: *A comparison of the mean rate of collision with respect to increasing agent number for VO, RVO, HRVO and ORCA geometric methods. In condition A, ideal obstacle knowledge is assumed. In condition B Obstacle and agent trajectory data is subject to measurement corruption. The effectiveness of all methods can be seen reduced in condition B, with the HRVO and ORCA methods shown to be the most resilient with respect to increasing agent number.*

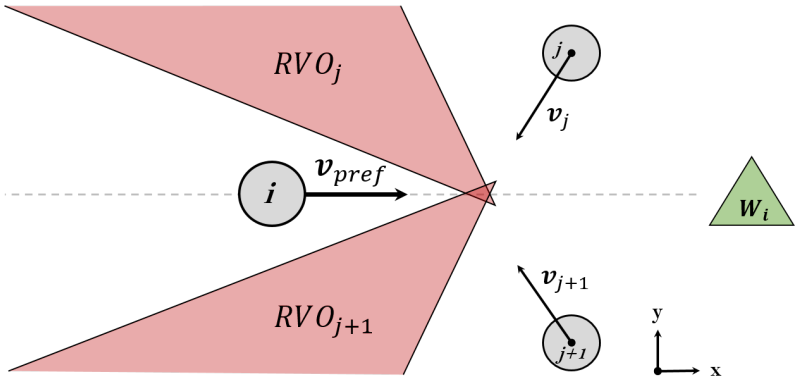


Figure 6.9: *A depiction of the scenario where the symmetry of the constraint set will induce a* dead/live-lock *scenario. The agents preferred velocity $\boldsymbol{v}_{pref}$ dictates the optimal trajectory toward the goal location W_i. Due to the symmetry of the constraint set RVO_j and RVO_{j+1} (shown in red), agent i will naturally resolve a trajectory that will reduce its velocity as it enters further into the deadlock.*

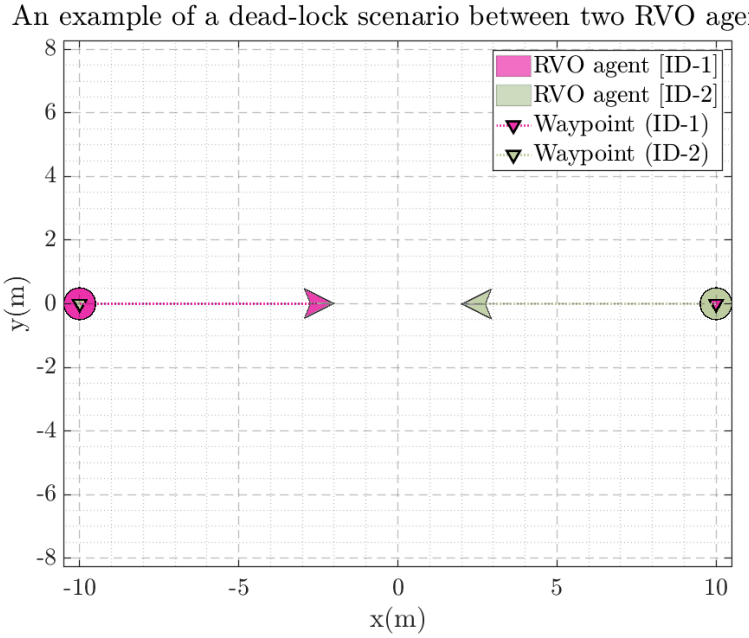


Figure 6.10: *A depiction of a* deadlock *scenario occurring in a symmetrical avoidance scenario between agents* 1 *and* 2*. Neither agent is able to resolve a direction of pass without the candidate velocity being less optimal than the current trajectory locking the agents in 3D space. A secondary protocol is necessary to override the trajectory and temporarily select a sub-optimal solution to instigate a direction of pass.*

to alleviate the situation is considered less optimal than the current preferred velocity. Unless a provision is made to allow the agent to violate a constraint momentarily, such as in [27, 250], the agent's behaviour will remain fixed. In the literature, such a condition is termed *deadlock* or *livelock* based on whether the agents are terminally static or mobile as seen in Figures 6.10 and 6.11.

Such situations may also occur where density of the constraint sets mutually prevents any agent from progressing to their target positions. In such scenarios a higher level strategy must be applied to intelligently preserve a collision-free trajectory by manipulating the constraint set or designing a new desired velocity $\boldsymbol{v}_{i,\,\mathrm{pref}}$. As part of the Monte Carlo analysis, the initial positions of the agents are perturbed by a noise signal $\boldsymbol{\chi}_{i,0}^{xy} = \mathcal{N}(\boldsymbol{\chi}_i^{xy}, \mathbf{I}_{2\times 2} \cdot \sigma_p^2)$ where σ_p is defined in Table 6.1. This process also aids in the prevention of the phenomena described in Section 6.4.3 by ensuring that the scenario is asymmetrical.

6.5 Interval Avoidance

Representing obstacles using deterministic geometric expressions naturally introduces problems when data describing the trajectory of the obstacle, or the agent, becomes un-

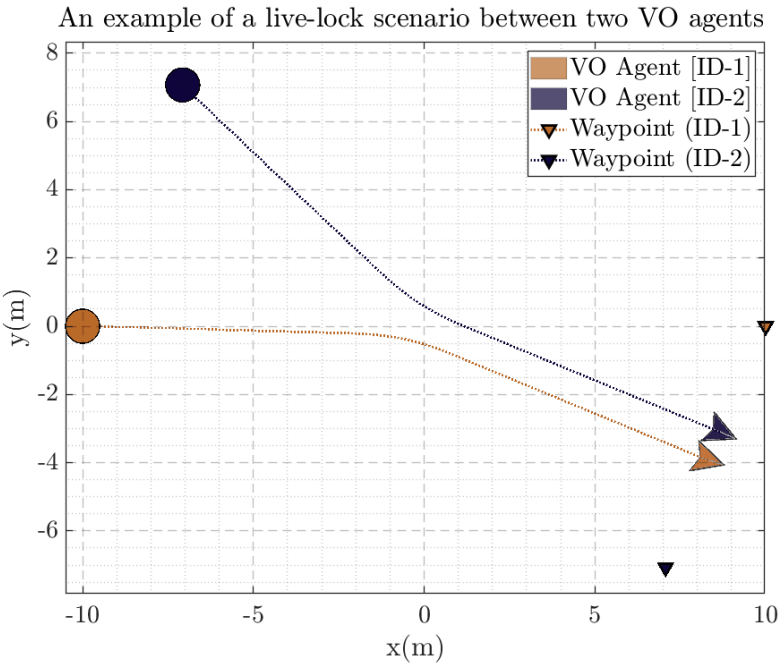


Figure 6.11: *A* livelock *scenario occurring between two agents, holding the agents in a fixed configuration until a secondary process breaks the symmetrical behaviour. Unlike the deadlock scenario, the agents main in a fixed configuration despite them both continuing to move through the environment.*

reliable. This is typical of scenarios where SDA algorithms are implemented on on-board physical systems relying on additional filters and conditioning to achieve smooth obstacle signals. Utilising some of the aspects of interval analysis it is possible to propagate all sources of measurement uncertainty, numerical errors and represent them within an interval to allow further optimisation or reduction. In this section the *interval avoidance* (IA) approach is introduced. It is able to handle parameter uncertainty without linearisation or approximation in order to best estimate a region of optimal obstacle avoidance [63].

6.5.1 Interval Analysis

The concept of *interval analysis* as a tool for describing uncertainty was initially introduced to estimate computer rounding errors, and later extended to guarantee state estimation [7, 115, 207]. The premise of intervals dictates that a state cannot be observed directly: instead belonging to an interval $x \in [x]$ where $[x] = [\underline{x}, \overline{x}]$ [166,207]. It is assumed in these works that the *unmanned aerial vehicle* (UAV) has the capability of measuring an obstacle's state x using an on-board array of sensors. An interval may then be defined that encapsulates the measurement $\tilde{x}$ in addition to the uncertainty derived from the reliability of the sensor which is assumed normally distributed with standard deviation σ_x. In these works a $3\sigma_x$ interval centred about the measurement $[x] = [\tilde{x} - 3\sigma_x, \tilde{x} + 3\sigma_x]$ is used

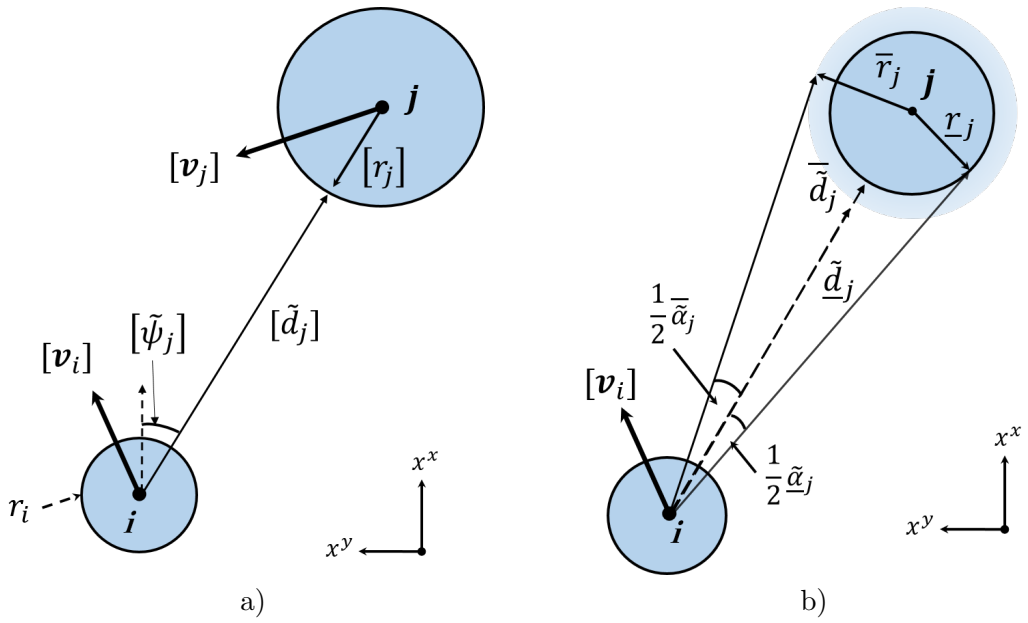


Figure 6.12: *a) Agent i observing obstacle j with uncertainty. The obstacles proximity, heading and radius are observed to be within uncertainty intervals $[\tilde{d}_j]$, $[\tilde{\psi}_j]$ and $[r_j]$ respectively. Agent radius r_i is assumed known. b) The obstacle's radius interval $[r_j]$ may be calculated from the angular width and proximity intervals $[\tilde{\alpha}_j]$ and $[\tilde{d}_j]$ respectively. The agent velocity interval $[\boldsymbol{v}_j]$ is then calculated by repeated sampling the angular position of j.*

in order obtain a ninety-seven percentile description of the possible values of x [63, 115].

6.5.2 Interval Sensing

In Section 6.1.1 the model adopted for the observation of obstacle j by agent i is presented given the various errors associated with the sensing system. The true measurements of the proximity of j, relative heading and angular width are assumed to belong to the defined intervals $[\tilde{d}_j] = [\tilde{d}_j - 3\sigma_r, \tilde{d}_j + 3\sigma_r]$, $[\tilde{\psi}_j] = [\tilde{\psi}_j - 3\sigma_c, \tilde{\psi}_j + 3\sigma_c]$ respectively as Figures 6.12(a) and 6.12(b) describe.

The relationship between the Cartesian position of j and its spherical measurements $\tilde{d}_j, \tilde{\psi}_j$ and $\tilde{\alpha}_j$ is initially defined in Equation (6.1). From this it can be deduced that the corresponding region bounding all possible Cartesian obstacle positions $[\boldsymbol{p}_j] = [p_j^x, p_j^y, p_j^z]^T$ in the axes of i may be expressed as Equation (6.7.

$$[\boldsymbol{p}_{k,j}] = \begin{bmatrix} [p_{k,j}^x] \\ [p_{k,j}^y] \end{bmatrix} = \begin{bmatrix} \cos([\tilde{\psi}_{k,j}]) & -\sin([\tilde{\psi}_{k,j}]) \\ \sin([\tilde{\psi}_{k,j}]) & \cos([\tilde{\psi}_{k,j}]) \end{bmatrix} \cdot [\tilde{d}_{k,j}] - \boldsymbol{p}_{i,sensor} \tag{6.7}$$

Using the relationship defined in Equation (6.2) the interval describing the minimal and maximal radii of j is propagated. This relationship incorporates the uncertainty the

angular extents of j as $[\alpha_j] = [\tilde{\alpha}_j - 3\sigma_c, \tilde{\alpha}_j + 3\sigma_c]$ and proximity $[\tilde{d}_j]$. Calculation of the radii interval can then be written as Equation (6.8).

$$[r_j] = [\underline{r}_j, \overline{r}_j] = \frac{sin\left(\frac{1}{2}[\alpha_j]\right)}{1 - \sin\left(\frac{1}{2}[\tilde{\alpha}_j]\right)} \cdot [\tilde{d}_j] \tag{6.8}$$

As with the deterministic sensor model shown in Section 6.1.1, it is otherwise assumed that agent i has no other prior knowledge of obstacle j. The velocity interval $[\boldsymbol{v}_{k,j}]$ for obstacle j is calculated via the discrete differencing of the known position intervals $[\boldsymbol{p}_{k,j}]$ and $[\boldsymbol{p}_{k-1,j}]$ over the time samples Δt. It follows that the components of the interval vectors $[\boldsymbol{p}_{k,j}]$, $[\boldsymbol{v}_{k,j}]$ and $[r_{k,j}]$ represent the Gaussian uncertainty in the geometric parameters of j. Equally, it is assumed that agent i measures it's own true position and velocity using the same assumptions in order to propagate the uncertainty intervals $[\boldsymbol{p}_{k,i}] = [\boldsymbol{p}_{k,i} - 3\sigma_p, \boldsymbol{p}_{k,i} + 3\sigma_p]$ and $[\boldsymbol{v}_i] = [\boldsymbol{v}_{k,i} - 3\sigma_v, \boldsymbol{v}_{k,i} + 3\sigma_v]$ respectively. This is under the premise that these measurements are obtained locally using on-board sensors and with their own sources of measurement uncertainty parameterised by their standard deviations σ_p and σ_v respectively.

6.5.3 Discerning the Likelihood of Collision

It has been demonstrated how the intervals in obstacle j's relative position $[\boldsymbol{p}_j]$, relative velocity $[\boldsymbol{v}_j]$ and defining radius $[r_j]$ are defined from geometric obstacle deductions. Based on j's trajectory intervals, an estimate of the current *miss interval* $[\boldsymbol{r}_m]$ can be formulated containing all projected geometric separations at the time of closest approach $[\tau]$ (see Figure 6.13). Similarly, a region enveloping all the projected points of minimal separation can be defined and is referred to as the *closest approach* interval $[r_{ca}]$.

It can then be said that given $[\boldsymbol{v}_j]$, the true time to collision between i and j belongs to the interval $\tau \in [\tau]$ where the obstacle will pass through the region defined by $[r_{ca}]$. From Figure 6.13, it can also be said that if $\boldsymbol{v}_j \in [\boldsymbol{v}_j]$ and $\boldsymbol{r}_m \in [\boldsymbol{r}_m]$ then a perpendicular miss distance $\boldsymbol{r}_m$ exists for all possible values of $\boldsymbol{v}_j$. This may be expressed in terms of the inner product of the two interval vectors as $[\boldsymbol{v}_j] \cdot [\boldsymbol{r}_m] = 0$. Using the geometric obstacle configuration the planar miss interval $[\boldsymbol{r}_m]$ is then defined from the relationship between $[\boldsymbol{v}_j]$ and the position interval $[\boldsymbol{p}_j]$ in Expressions (6.9) and (6.10).

$$[\hat{\boldsymbol{r}}_m] = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} [\hat{\boldsymbol{v}}_j] \tag{6.9}$$

$$[\boldsymbol{r}_m] = det([\boldsymbol{p}_j], [\hat{\boldsymbol{v}}_j]) \cdot [\hat{\boldsymbol{r}}_m] \tag{6.10}$$

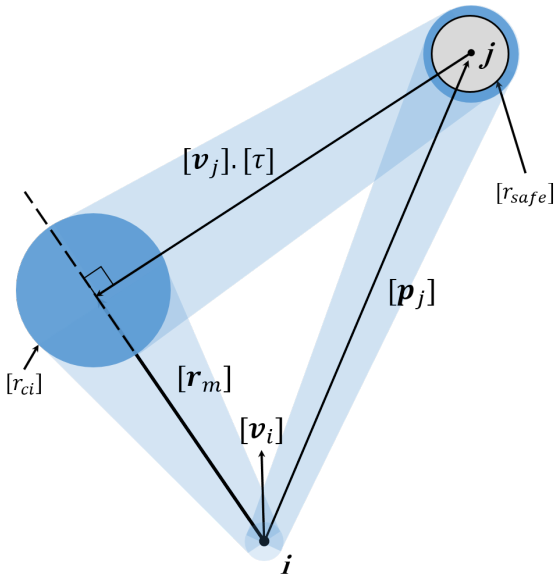


Figure 6.13: *The miss interval* $[\boldsymbol{r}_m]$ *defined geometrically, in the configuration space of* i*. The shaded regions define the set of vectors belonging to each interval. Here* $[r_{safe}]$ *defines the uncertainty in the required obstacle separation. The definition of the time closest approach interval* $[\tau]$*, and point of closest approach* $[r_{ca}]$*, may be seen geometrically constructed from the obstacle's relative position* $[\boldsymbol{p}_j]$*, velocity* $[\boldsymbol{v}_j]$ *and* $[\boldsymbol{r}_m]$*.*

Taking further advantage of the orthogonality of the member vectors of $[\boldsymbol{r}_m]$ in Equation (6.10), the range of prospective times until collision naturally takes the form of the interval $[\tau]$. The physical meaning of this property is described geometrically in Figure (6.13), is defined in Equation (6.11) and assumes $\boldsymbol{v}_j \in [\boldsymbol{v}_j]$ remains constant for Δt:

$$\begin{aligned}
[\boldsymbol{r}_m] &= [\boldsymbol{p}_j] + [\boldsymbol{v}_j] \cdot [\tau] \\
[\boldsymbol{v}_j] \cdot [\boldsymbol{r}_m] &= [\boldsymbol{p}_j] \cdot [\boldsymbol{v}_j] + ([\boldsymbol{v}_j] \cdot [\boldsymbol{v}_j])[\tau] \\
([\boldsymbol{v}_j] \cdot [\boldsymbol{v}_j])[\tau] &= [\boldsymbol{v}_j] \cdot [\boldsymbol{r}_m] - [\boldsymbol{p}_j] \cdot [\boldsymbol{v}_j] \\
[\tau] &= -\frac{[\boldsymbol{p}_j] \cdot [\boldsymbol{v}_j]}{[\boldsymbol{v}_j] \cdot [\boldsymbol{v}_j]}
\end{aligned} \tag{6.11}$$

From Equation (6.11) it can be inferred that if the bounds of $[\tau]$ are both positive (i.e $\overline{\tau} > 0, \underline{\tau}$) then a collision is likely to occur and the avoidance routine is necessary. Ambiguity does however occur when $0 \in [\tau]$ as this implies the uncertainty in the obstacle trajectory means there exists both a possibility of collision and that no collision may occur. To assure the safety of the vehicle, it is clear that if a possibility of collision exists then the UAV should act to avoid the threat. This condition is represented simply in interval terms as sign of the supremum $\overline{\tau}$. The avoidance routine should therefore be executed when $\overline{\tau} \geq 0$ is met.

6.5.4 Safe Separation

The notion of collision events, object classifications and representations in OpenMAS are initially presented in Section 2.1.2 and associated works [66]. A collision event is said to occur between two agents when i and j violate the condition $(r_i + r_j) \geq ||\boldsymbol{p}_j||$. Here r_i and r_j represent the radii characterising the physical geometry of i and j respectively. r_i is assumed to be a constant known to agent j, while $r_j \in [r_j]$ is subject to measurement uncertainty as seen in Equation (6.8). By inclusion of an obstacle safety factor s_f, the minimum safe separation $[r_{safe}]$ is defined by Equation (6.12).

$$[r_{safe}] = r_i + s_f \cdot [r_j] \tag{6.12}$$

$$[r_{res}] = [r_{safe}] - ||[\boldsymbol{r}_m]|| > 0 \tag{6.13}$$

Relating Equation (6.12) to the miss interval $[\boldsymbol{r}_m]$, the term $[r_{res}]$ is defined in Equation (6.13) [187] to describe a *resolution interval*. A conflict between i and j then exists when $\underline{r}_{res} \leq 0$ by indicating that separation is insufficient at $t = t_k + \tau$.

6.5.5 Optimal Resolution Intervals

In the event that $\overline{\tau} > 0$, an manoeuvre must be designed to avert collision with the obstacle. In these works, an avoidance trajectory is said to be optimal when its selection represents a min-max relationship of the control effort (deviation from current course) and the rate of increasing separation from the obstacle. This can be seen shown in Figure 6.13, by the velocity that maximises the miss interval $[\boldsymbol{r}_m]$ at the time of closest approach τ to minimise the risk of collision (see Equation (6.15)). The optimal direction of avoidance can be represented as the minimal solution to the following Hamiltonian (6.14).

$$H_a = -\boldsymbol{r}_m \cdot \boldsymbol{v}_j - (||\boldsymbol{a}||\cdot\tau) \cdot \boldsymbol{r}_m) \cdot \hat{\boldsymbol{a}}_i \tag{6.14}$$

$$\min_a J = -\frac{1}{2}||\boldsymbol{r}_m||^2 \tag{6.15}$$

Here the relationship between $\boldsymbol{r}_m$ and the correction acceleration $\boldsymbol{a}$ is represented as a cost function. Examining Equation (6.14), it can be deduced that H is minimal when $\boldsymbol{a} \cdot \boldsymbol{r}_m = 0 \therefore \boldsymbol{a} \parallel \boldsymbol{r}_m$. It can therefore be said that acceleration $\boldsymbol{a}$ is optimal when $\boldsymbol{a}_i \parallel \boldsymbol{r}_m$ and the manoeuvre optimally increases the separation between i and j [163,187]. We can therefore define the set of optimal accelerations geometrically to be such that $[\hat{\boldsymbol{a}}] \equiv [\hat{\boldsymbol{r}}_m]$.

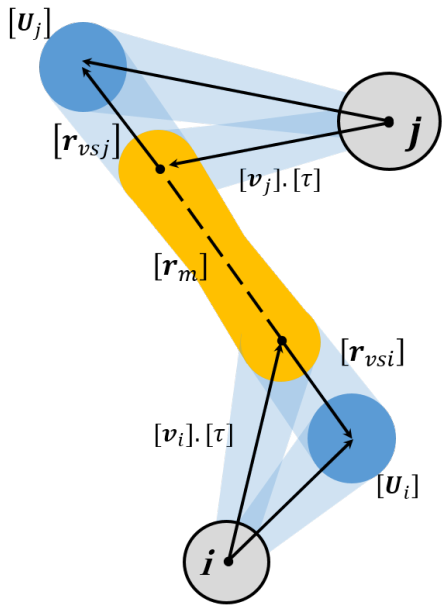


Figure 6.14: *The definition of the shared resolution intervals* $[\boldsymbol{r}_{vsi,vsj}]$ *(blue) as a result of the uncertainty in the miss interval* $[\boldsymbol{r}_m]$ *(yellow). The optimal region* $[\boldsymbol{U}_{i,j}]$ *(dark blue) then describes the set of candidate velocities which consider the obstacle trajectory uncertainty. Selection of a velocity from* $[\boldsymbol{U}_{i,j}]$ *is then subject to strategy.*

6.5.6 Vector Sharing

The magnitude of the interval correction vector required to avoid the obstacle is determined through the process of *vector sharing* (VS) [187]. We assume that if j is able, it would act to prevent collision with i by enlarging $[\boldsymbol{r}_m]$. Based on the uncertainty in both agent's trajectories it is possible to define an expression for the *shared separation interval* (6.16).

$$[\boldsymbol{r}_{vsi}] = \frac{|[\boldsymbol{v}_j]|}{|[\boldsymbol{v}_i]|+|[\boldsymbol{v}_j]|} \cdot [r_{res}] \cdot [\hat{\boldsymbol{r}}_m] \tag{6.16}$$

Equation (6.16) describes the distribution of i's correction velocity $[\boldsymbol{v}_{vsi}]$ proportional to $[\boldsymbol{v}_i]$ and $[\boldsymbol{v}_j]$, scaled by resolution magnitude $[r_{res}]$. The physical meaning of $[\boldsymbol{v}_{vsi}]$ is shown in Figure 6.14, where the agent with the larger velocity takes responsibility for a larger correction interval vector $[\boldsymbol{r}_{vsi}]$ in the optimal direction $\hat{\boldsymbol{r}}_m$. We aim to define the interval containing the optimal heading vector $\boldsymbol{U}_i^* \in [\boldsymbol{U}_i]$ given the obstacle's trajectory uncertainty. The interval itself can be assembled geometrically by extrapolating the agent's current velocity to the time of closest approach τ (see Figure 6.14). The resulting avoidance heading interval $[\boldsymbol{U}_i]$ for the agent can then be seen expressed in

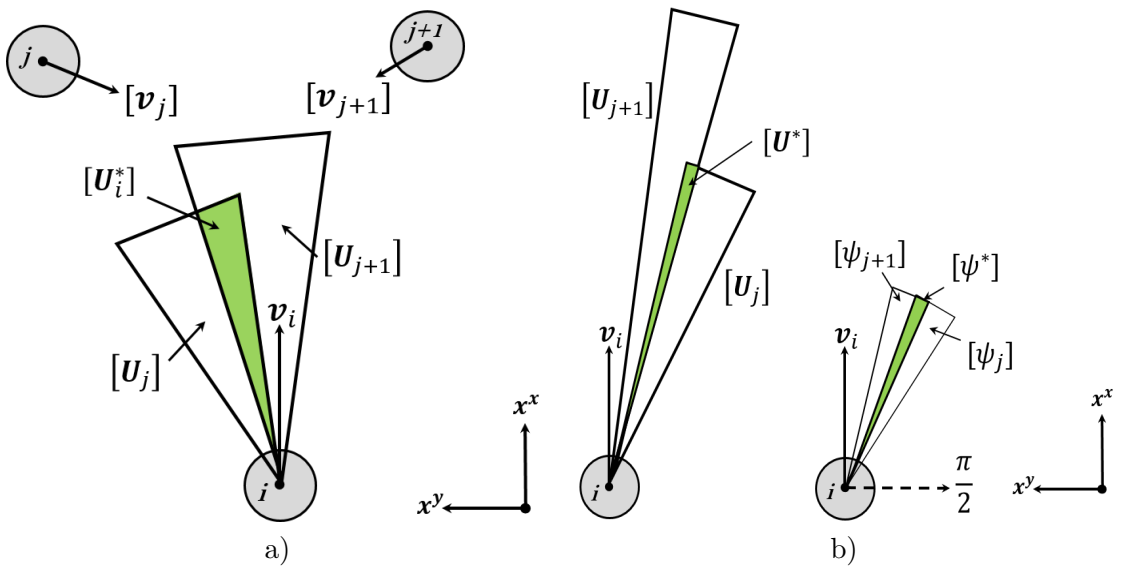


Figure 6.15: *a) A multi-agent collision scenario from the perspective of i, observing obstacle's j and $j+1$ moving with velocities $[\boldsymbol{v}_j]$ and $[\boldsymbol{v}_{j+1}]$ respectively. The mutual avoidance region $[\boldsymbol{U}_i^*]$ can be seen defined as the intersection of the optimal correction intervals $[\boldsymbol{U}_j^*]$ and $[\boldsymbol{U}_{j+1}^*]$ (shaded green). b) Extraction of the mutual avoidance heading interval $[\psi^*]$ from the planar projection of the interval $[\boldsymbol{U}^*]$. This shown to be equivalent to the intersection of the heading intervals $[\psi_j]$ and $[\psi_{j+1}]$ (shaded green).*

Equation (6.17).

$$[\boldsymbol{U}_i] = [\boldsymbol{v}_i] \cdot [\tau] + [\boldsymbol{r}_{vsi}] \tag{6.17}$$

The avoidance interval $[\boldsymbol{U}_i]$ defines a region enveloping the resolution vector necessary to optimally avoid obstacle j. The relative heading angle intervals $[\psi_i]$ can now be assembled geometrically from the interval $[\boldsymbol{U}_i]$.

6.5.7 Multiple Obstacle Consideration

In the event that there is a collision likelihood for multiple obstacles ($\tau_i > 0$), an interval must be defined containing the globally valid avoidance trajectories $[\boldsymbol{U}^*]$. Under the principle of interval analysis we are able to consider multiple obstacles by defining their intersection as seen Figures 6.15(a) and 6.15(b).

The global optimal region $[\boldsymbol{U}_i^*]$ is defined such that $[\boldsymbol{U}_i^*] = [\boldsymbol{U}_i^*] \cap [\boldsymbol{U}_j]_{j=1:n}$ where n is the number of obstacles. Scenarios may exist where there can be no intersection between avoidance headings (i.e $[\boldsymbol{U}_1] \notin [\boldsymbol{U}_2]$). In such cases a strategy must be applied in order ensure a valid trajectory interval is always available. An immediate solution is to prioritise the avoidance trajectory set based on obstacle proximity $P_{i=1:n} = \frac{1}{||\underline{\boldsymbol{p}}_{i=1:n}||}$. Here $P_{i=1:n}$ is calculated from the lower bound of the interval containing the possible proximity of the

obstacle. This ensures that in the worst case, optimal avoidance of the closest obstacle is to be prioritised [115]. The complete algorithm is summarised in Algorithm (2)

Algorithm 2: Calculation of the global optimal resolution region $[\boldsymbol{U}^*]$ from the observed obstacle set $\mathbf{O}_{j=1:n}$. The optimal heading and speed intervals $[\psi]$ and $[v_i]$ respectively are then passed to the low-level controller.

Data: $\mathbf{W}_j$,$\mathbf{O}_j$, n
Result: $[v_i]$,$[\psi^*]$
`// Read the agent's local state.`
$[\boldsymbol{p}_i], [\boldsymbol{v}_i], r_i$ = GetAgentState()
`// Measure the new obstacle states.`
$[\mathbf{W}_j$,$\mathbf{O}_j$,n] = ReadSensorBuffer()
`// Calculate waypoint(desired) trajectory.`
$[\boldsymbol{v}_{\text{pref}}]$ = GetWaypointPath(W_i)
`// Calculate priorities`
$\boldsymbol{P}_j$ = GetProximities($\mathbf{O}_j$)
`// Prioritise the obstacles.`
$\mathbf{O}_j$ = sort($\mathbf{O}_j$,$\boldsymbol{P}_j$, descending)
for *j=1:n* **do**
  `// Get the obstacle states.`
  $[\boldsymbol{p}_j], [\boldsymbol{v}_j], [r_j]$ = GetObstacleTrajectory($\mathbf{O}_j$)
  `// Compute optimal avoidance trajectory.`
  $[\boldsymbol{U}_i]_j$ = ComputeOptimalRegion($[\boldsymbol{p}_i], [\boldsymbol{v}_i], r_i, [\boldsymbol{p}_j], [\boldsymbol{v}_j], [r_j]$)
  `// Store candidate region`
end
j = 1
while $j \leq n$ **do**
  `// Recursively compute the intersection.`
  $[\boldsymbol{U}^*]$ = intersect($[\boldsymbol{U}^*]$,$[\boldsymbol{U}_i]_j$)
  j++
end
`// Actuation limit's intersection.`
$[\boldsymbol{U}^*]$ = ApplyAcuatorConstraints($[\boldsymbol{U}^*]$)
`// Evaluate the control intervals.`
$[v_i^*], [\psi_i^*]$ = ComputeControlInputs($[\boldsymbol{v}_{pref}]$,$[\boldsymbol{U}^*]$)

The agent first makes a measurement of it's own state intervals $[\boldsymbol{p}_i]$ and $[\boldsymbol{v}_i]$ in accordance to Algorithm 2. The agent samples it's immediate ($||d_j|| < d_{max}$) environment using it's on-board sensors and attains measurements of each obstacle's position $[\boldsymbol{p}_{j,1:n}]$, velocity $[\boldsymbol{v}_{j,1:n}]$ and characteristic radius $[r_{j,1:n}]$. Based on the obstacle's priority (i.e. proximity) the optimal avoidance region is stored in a vector of candidate velocity intervals. The mutual avoidance region $[\boldsymbol{U}_i^*]$ is then found through successive intersections of $[\boldsymbol{U}_1, \boldsymbol{U}_2, ..., \boldsymbol{U}_n]$. The resulting interval represents the velocity region that will act to maximise the separation with the obstacle set and decrease collision likelihood.

Initial Condition	Position (m)	Velocity (ms^{-1})	Heading (rad)
MAV 1 (alpha001)	[-10.00,0.00]	[2.00,0.00]	0.00
MAV 2 (beta001)	[10.00,0.00]	[-20.0,0.00]	1.57

Table 6.3: *The initialisation conditions of the two MAVs in example one representing an direct collision scenario.*

6.5.8 Trajectory Selection

Once the target velocity interval $[\boldsymbol{U}_i^*]$ is defined it must be evaluated against the velocity constraints of agent i. The attainable velocities of i are a result of the accelerations that can be generated by the system at time t_k. These are intrinsic to the dynamical behaviour of agent i, represented by the function $[\boldsymbol{a}_{i,max}] = d(\boldsymbol{x}_{k,i}, [\boldsymbol{u}_{i,max}])$.

The inclusion of acceleration constraints can be represented under the notion of intervals by assigning the maximal accelerations to an interval $[\boldsymbol{a}_{i,max}] \in \mathbb{R}^{2\times 1}$. The viable velocity region can then be found by intersecting the target velocity region with the velocities that can be actuated by the system $[\boldsymbol{v}_{i,\text{pref}}] = [\boldsymbol{U}_i^*] \cap ([\boldsymbol{v}_i] + \Delta t \cdot [\boldsymbol{a}_{i,max}])$. At t_{k+1}, agent i must enact a single control input $\boldsymbol{v}_{i,\text{pref}} \in [\boldsymbol{v}_{i,\text{pref}}]$ to mitigate the collision at $t = t_k + \tau$. In associated works [63], the vector central to $\boldsymbol{v}_{i,\text{pref}} = mid([\boldsymbol{v}_{i,\text{pref}}])$ is taken as the "best" estimate of the true avoidance trajectory. The corresponding control inputs $\phi_{k,i}$ and $v_{k,i}$ can then be resolved using the methods described in Section 6.3.1.

6.6 Performance Evaluation

In this section, the proposed IA algorithm is demonstrated in parallel to the state of the art geometric approaches in a series of representative scenarios. The evaluation parameters and sensing conditions presented in Section 6.4.1 are applied again in the following scenarios.

6.6.1 Direct Collision

The most challenging scenario for SDA algorithms classically is a direct collision [63, 204]. Here, the geometric angle of approach is such that the initial miss distance is minimal and the resolution interval is maximal $[r_{res}] = r_i + [r_j]$. In addition, due to the fact that the trajectories of the two agents are parallel initially (i.e. $\boldsymbol{v}_i \parallel \boldsymbol{v}_j$) the direction of pass becomes ambiguous (see Section 6.2). The initial conditions for the direct collision scenario are given in Table 6.3.

The trajectories generated by the IA method in the direct collision scenario are seen in Figure 6.16. Here, the agents are operating under the imperfect sensing conditions

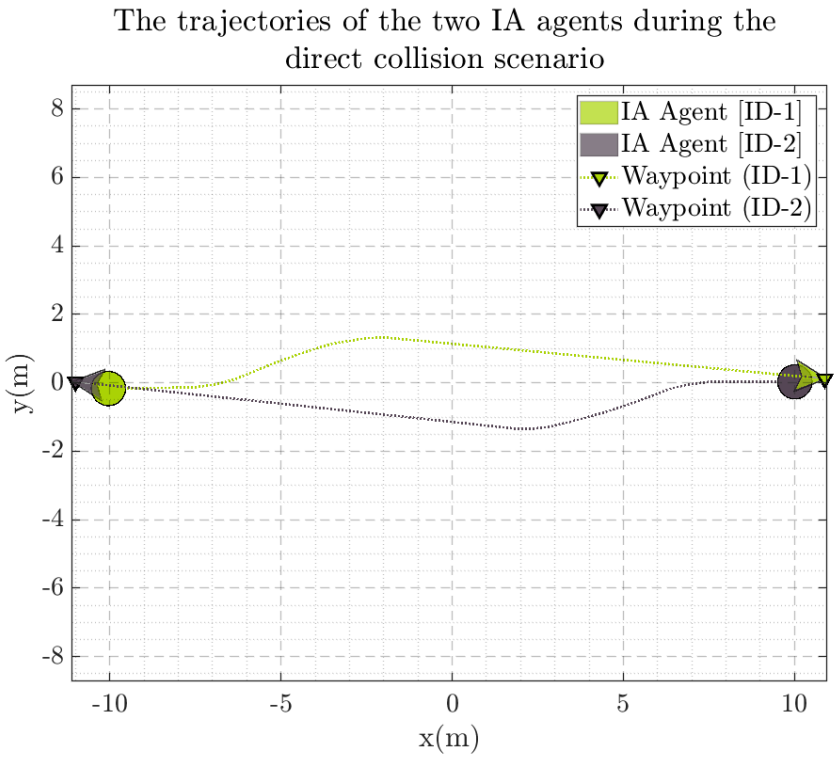


Figure 6.16: *A planar view of the direct conflict scenario whereby the IA method is shown to negotiate imminent collision between agents* 1 *and* 2. *Here each agent is tasked with proceeding from their initial position (circles) in order to each their associated waypoint (triangles). Both agents can be seen to successfully enlarge their separation as they move towards each other in the presence of the presented sensing conditions.*

presented in Table 6.1. The agents can be seen to negotiate the direct conflict by actively extending the miss interval $[\boldsymbol{r}_m]$ before the point of closest approach $\boldsymbol{r}_{min}$ is reached. This can be seen in Figure 6.18; by the adjusted heading ψ and forward speed v^x as soon as the neighbourhood condition of $15m$ is satisfied. The algorithm is shown to extend the miss interval until no opportunity for collision exists. This is clearly indicated by the momentary increase in acceleration at $t = 3.25s$ as the trajectory toward the agent's allocated waypoint becomes available. As a result, a conservative $2.2m$ minimum separation seen at the time of closest approach in Figure 6.17. The performance of several geometric approaches introduced in Chapter 5 are presented here for comparison in this conflict scenario. Using the assumptions and parameters presented in Table 6.1, a one thousand cycle Monte-Carlo analysis was used to statistically evaluate the mean performance of each method under the presented conditions.

The performance of the algorithms with respect to several key parameters are presented in Table 6.4. Here, it is clear from the mean number of collisions that the IA algorithm is unique in it's ability to generate safe separation under the presented sensing conditions. This may be due to the interval representation of the trajectory of j

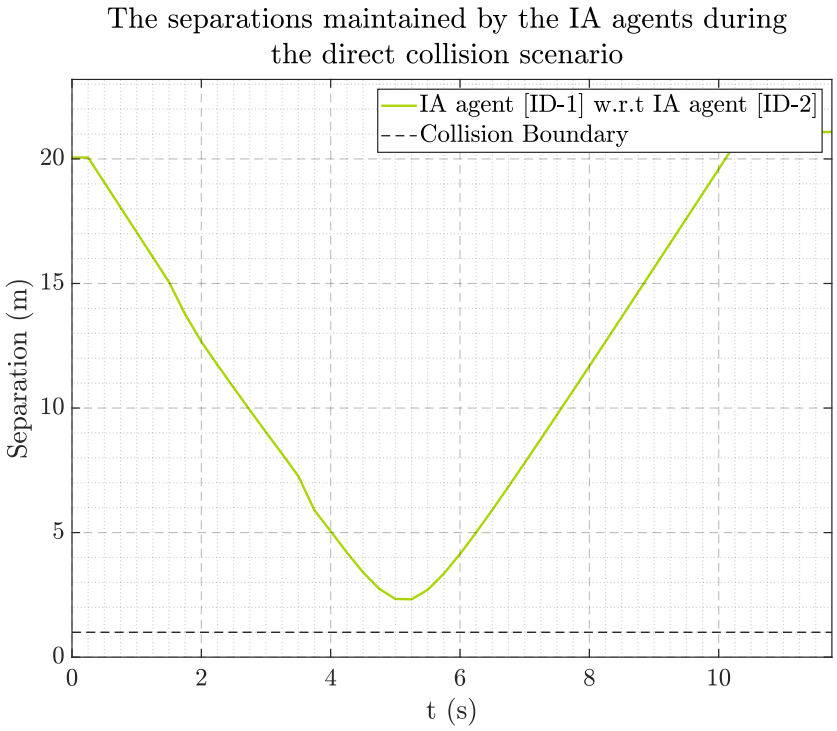


Figure 6.17: *The separation maintained by the IA algorithm as agents* 1 *and* 2 *through the direct collision scenario. The agents are shown to generate a minimum separation of* $2.5m$ *at the point of closest approach* ($t = 5.25s$) *under the presented sensing conditions.*

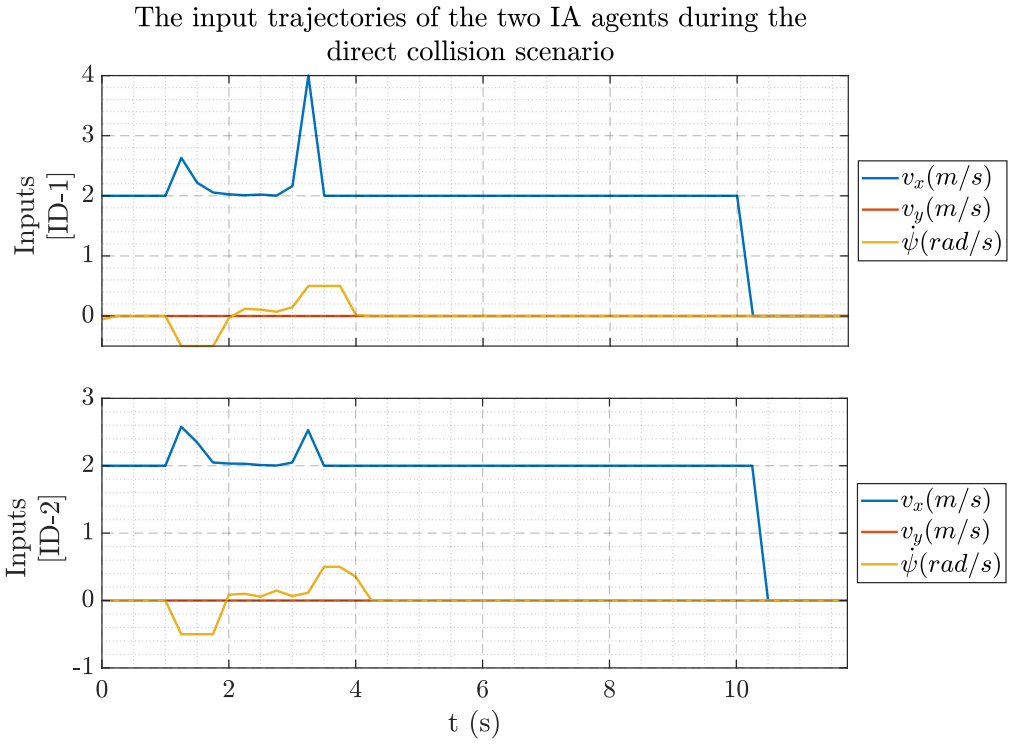


Figure 6.18: *The input trajectories of agents* i *and* j *throughout the direct collision planar encounter. The IA algorithm is shown to generate a right-hand direction pass indicated by the heading changes* $\dot{\psi}$*. both agents are shown to accelerate in order to achieve the optimal avoidance trajectories and once after waypoint trajectory becomes unimpeded.*

Approach	Mean Collisions	Mean waypoints	Minimum Separation (m)	Computation Time (ms)
VO	0.8560	2.0000	1.0095	0.1200
RVO	0.9000	2.0000	0.9658	0.1300
HRVO	0.6460	2.0000	1.0214	0.1300
ORCA	0.4840	2.0000	1.0711	0.1900
Vector Sharing	0.7460	2.0000	1.0240	0.2500
Interval Avoidance (IA)	0.0000	1.9940	2.6305	1.0300

Table 6.4: *The statistical results of one thousand cycle Monte-Carlo analysis on the direct collision scenario under representative sensing conditions. A comparison of the common VO derivatives, the classical Vector Sharing method and the proposed IA method is given.*

when it is parallel with i's (i.e. $\boldsymbol{v}_i \parallel \boldsymbol{v}_j$). As the obstacle approaches the condition, all trajectories around and including this condition are expressed simultaneously due to the associated sensor uncertainty. In the unlikely event of perfect symmetry, a trajectory may instead be selected from the extents of the optimal region $\boldsymbol{v}_{i,k+1} = [\underline{\boldsymbol{U}}]_j \vee [\overline{\boldsymbol{U}}]_j$, assuring inherent "pass on the right" or "left" behaviour respectively.

The mean minimum separation suggests that the approach is inherently more "conservative" in the allowable miss distance. This behaviour is defined by the collision likelihood interval $[\tau]$, which includes the trajectory uncertainty of both the obstacle and agent. As a result i will continue to extend the miss interval until $[\tau] < 0$ is satisfied. It is also clear from Table 6.4 that the mean computation time for the IA algorithm is considerably longer than other associated methods in the same conditions. This suggests that while able to generate safe separation effectively, the mean time to compute the avoidance trajectories may result in the algorithm being less scalable in more densely cluttered environments.

6.6.2 Four-Agent Intersection

In the next scenario, a four way flight path intersection is presented as a re-framing of the direct collision scenario as a multi-agent conflict. Here each agent is tasked with negotiating a flight path conflict with three other agents on route to their assigned waypoint. The initialisation conditions for the four agents are given in Table 6.5. This scenario is analogous to concentric collision scenario with low agent population, where initial conditions of the agents are such that a collision will occur unless an alternative trajectory is generated (see Figure 6.19) [27, 65, 251].

As described in Table 6.6, the IA method is distinct in it's ability to avoid collisions by maintaining safe separation throughout each independent Monte-Carlo run. An example of the separations maintained through the point of closest approach is also

Initial Condition	Position (m)	Velocity (ms^{-1})	Heading (rad)
MAV 1 (alpha001)	[-10.00,0.00]	[2.00, 0.00]	0.00
MAV 2 (beta001)	[0.00,10.00]	[0.00,-2.00]	0.79
MAV 3 (gamma001)	[10.00,0.00]	[-2.00,0.00]	1.57
MAV 4 (delta001)	[0.00,-10.00]	[0.00, 2.00]	2.36

Table 6.5: *The initialisation conditions of the four MAVs in example two representing a four agent intersection scenario.*

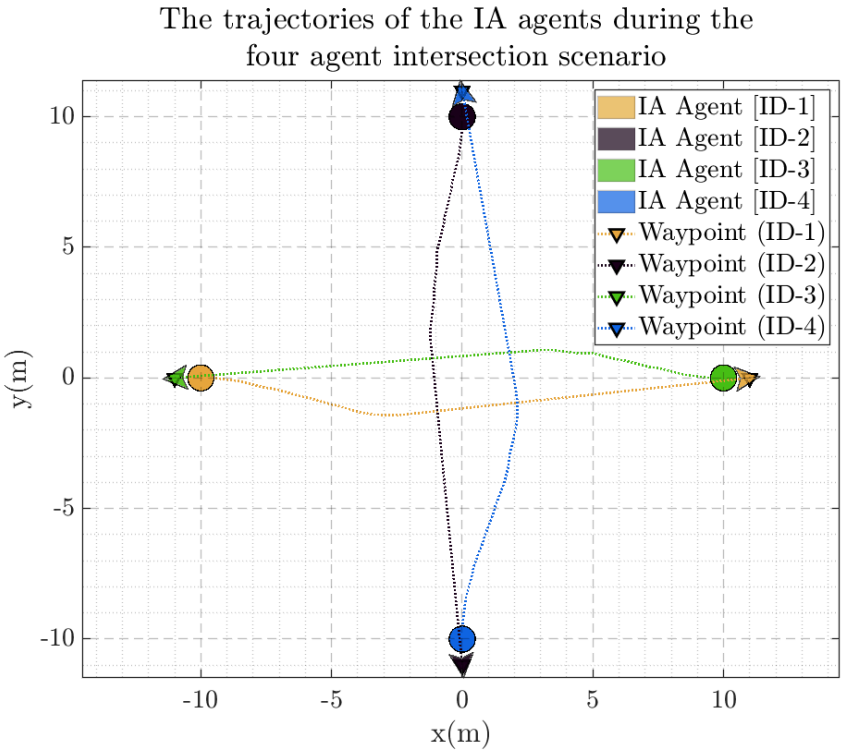


Figure 6.19: *A depiction of the IA algorithm resolving a four-agent planar intersection example. Each agent is assigned an antipodal waypoint that requires the agent to move through the conflict centre. Each agent is shown to compute a trajectory that successfully mitigates collision with each of the other agents.*

Approach	Mean Collisions	Mean waypoints	Minimum Separation (m)	Computation Time (ms)
VO	2.2040	4.0000	0.8710	0.3400
RVO	1.7730	4.0000	0.8810	0.3300
HRVO	1.5500	4.0000	0.8606	0.3000
ORCA	1.0880	4.0000	0.9868	0.2800
Vector Sharing	3.9300	4.0000	0.3543	0.1400
Interval Avoidance (IA)	0.0000	3.9514	3.6708	2.7300

Table 6.6: *The statistical results of a one thousand cycle Monte-Carlo analysis of the four agent intersection scenario under representative sensing conditions. A comparison of the common VO derivatives, the classical Vector Sharing method and the proposed IA method is given.*

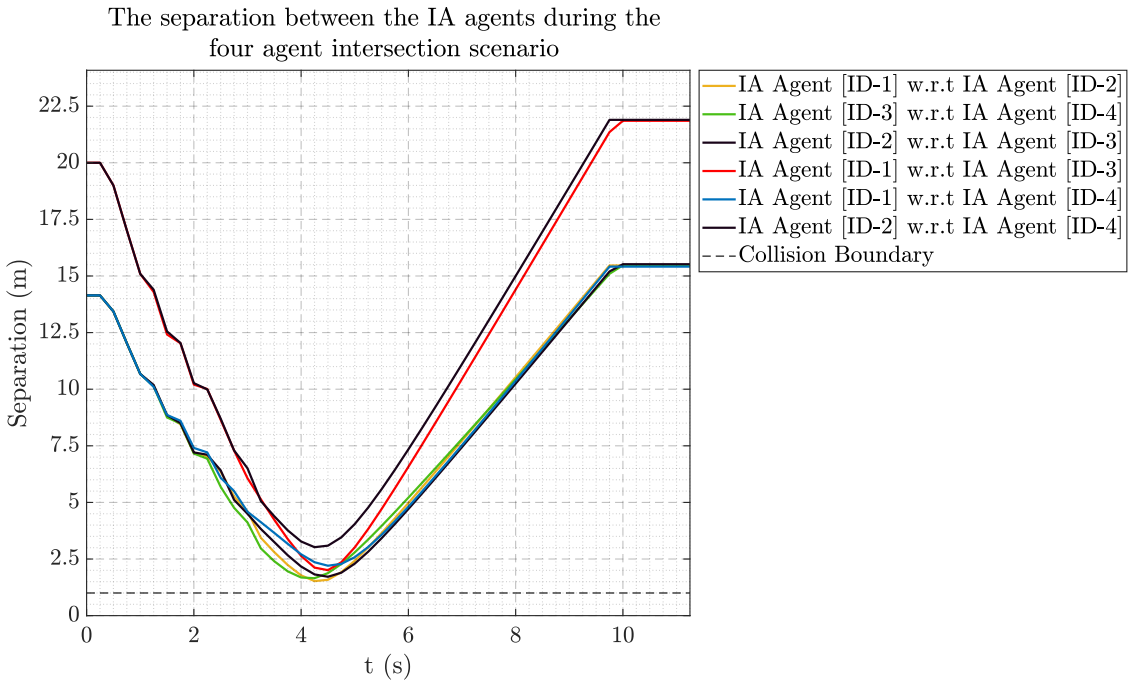


Figure 6.20: *A comparison of the inter-agent separations during the four-agent intersection scenario. Here, each relationship represents the separation of agent j (shown in brackets) with respect to agent i. Throughout the collision the IA algorithm is able to maintain a minimum separation of* $1.5m$

given in Figure 6.20. The performance of each of the VO derivatives is shown to vary as expected; with the original VO methods mean collision rate being higher than that of more recent VO methods. The most competitive method is the ORCA method, where the measurement uncertainty induced 1.0880 collisions on average in the intersection scenario. The vector sharing approach, while only presented as a single agent avoidance method in [187], is shown here to be least effective in tolerating both measurement uncertainty and simultaneous avoidance of multiple obstacles.

6.6.3 Concentric Collision

In [94, 106, 251] the authors consider a scenario where many agents are set to collide at the same point in 2D space. Here we consider ten agents approaching the collision concentrically, with collision ensured by the antipodal positioning of their waypoint (seen in Figure 6.21). Similar to Section 6.4.2, this scenario is presented to demonstrate the performance of the IA algorithm with respect to several state of the art geometric methods for collision avoidance. The problem posed by this example examines the proposed method's ability to handle complex, cluttered environments where the collision is implied. The initialisation conditions for the ten agents are presented in Table 6.7.

A statistical comparison of performance of each avoidance approach is shown in

Initial Condition	Position (m)	Velocity (ms^{-1})	Heading (rad)
MAV 1 (alpha001)	[-10.00,0.00]	[2.00, 0.00]	0.00
MAV 2 (beta001)	[-8.09, 5.88]	[1.61,-1.18]	0.63
MAV 3 (gamma001)	[-3.09, 9.51]	[0.62,-1.90]	1.26
MAV 4 (delta001)	[3.09, 9.51]	[-0.62,-1.90]	1.89
MAV 5 (epsilon001)	[8.09, 5.88]	[-1.62,-1.18]	2.51
MAV 6 (zeta001)	[10.00, 0.00]	[-2.00, 0.00]	3.14
MAV 7 (eta001)	[8.09,-5.88]	[-1.62, 1.18]	-2.51
MAV 8 (theta001)	[3.09,-9.51]	[-0.62, 1.18]	-1.89
MAV 9 (iota001)	[-3.09,-9.51]	[0.62, 1.90]	-1.26
MAV 10 (kappa001)	[-8.09,-5.88]	[1.62, 1.18]	-0.63

Table 6.7: *The initialisation conditions of the third example scenario involving ten MAVs in a concentric collision.*

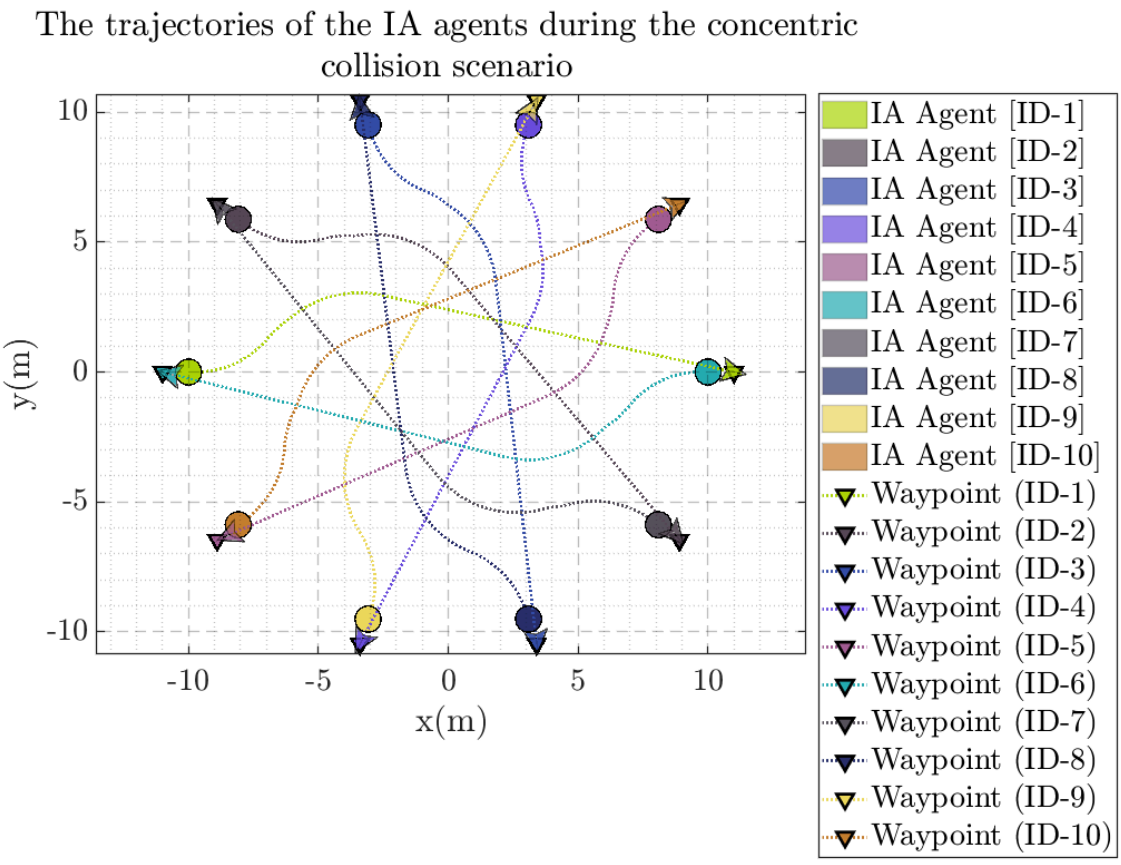


Figure 6.21: *A plan view of the planar concentric collision scenario involving ten interval avoidance (IA) agents. Here the agents are tasked with computing appropriate escape trajectories as a result of obstacle configuration within it's local neighbourhood d_{nei}. It is assumed that each agent is unable to communicate and that all measurements are subject to distortion.*

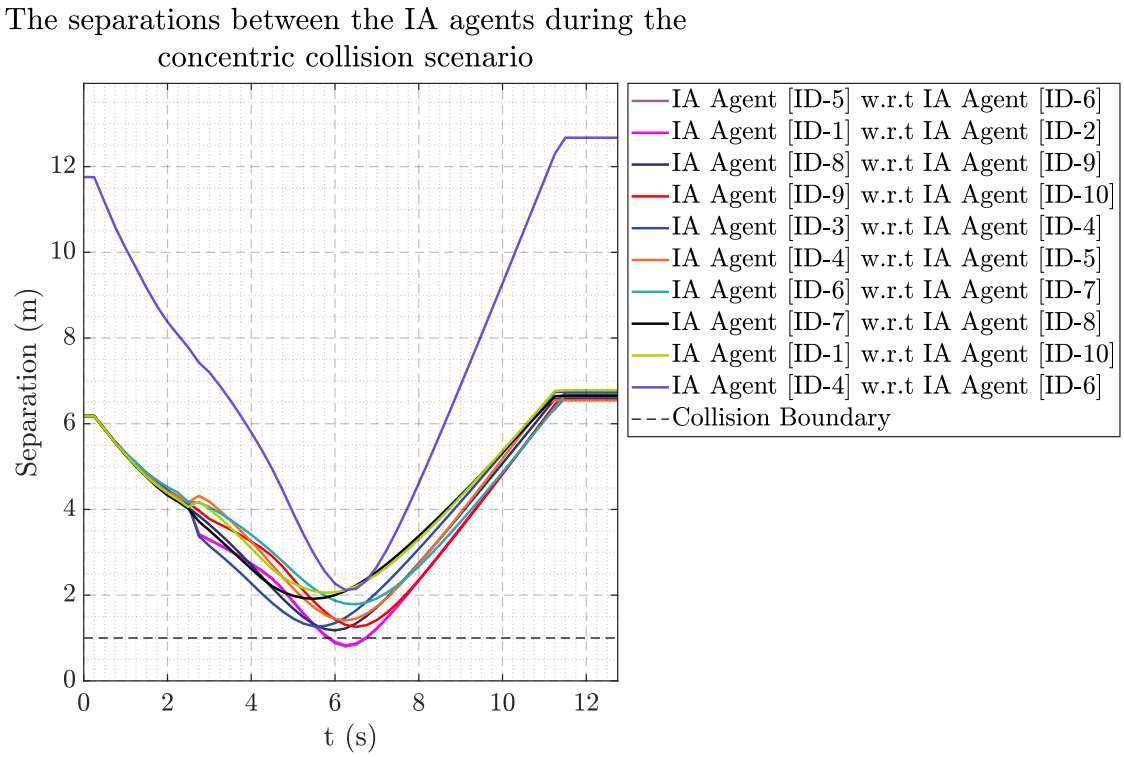


Figure 6.22: *A depiction of the ten inter-agent separations that were closest to collision through the ten agent concentric collision scenario. Here, each relationship represents the separation of agent j (shown in brackets) with respect to agent i. At $t = 5.75s$ agents 1 and 2 are shown to collide as the method is unable to determine a region of mutual avoidance that satisfies every member of the conflict.*

Algorithm	Mean Collisions	Mean waypoints	Mean Minimum Separation (m)	Mean Computation Time (ms)
VO	3.4090	10.0000	0.3487	0.7900
RVO	1.9960	10.0000	0.4710	1.0400
HRVO	2.0000	10.0000	0.3515	0.8700
ORCA	1.9800	10.0000	0.7983	0.5800
Interval Avoidance (IA)	3.9400	10.0000	0.5683	6.2600

Table 6.8: *The statistical results of a one thousand cycle Monte-Carlo analysis of a ten agent concentric collision scenario are presented. Under presented sensing conditions, the ORCA method is shown to yield the lowest mean collisions as indicated by the largest mean minimum separation. The IA algorithm is shown to be competitive in the minimum separation, but is unable to preserve the collision boundary in the presence of this of nine obstacles.*

Table 6.8. Here, the IA approach is competitive in it's ability to negate collision when compared to other methods for multiple obstacle avoidance. That said, it is clear that the performance of the IA algorithm is decreased in the presence of highly cluttered environments; indicated by an increased mean collision rate of 39.4% across the one thousand Monte-Carlo cycles. An example of this can be seen in Figure 6.22, where the collision boundary is violated at $t = 5.75s$ as the algorithm fails to main separation between agents 1 and 2.

Under the convention of the presented VO derivatives, the trajectory selection is made from the velocity constraints representing the obstacle trajectories at time t_k. The formulation of $\boldsymbol{U}_i^*$ under the IA approach however, does not explicitly state that the trajectories selected from the optimal region will increase separation with *all* obstacles; only a prioritised subset. It is therefore possible for agents to select a trajectory from $\boldsymbol{U}_i^*$ that may act to avoid obstacles in their nearest proximity, but increase the likelihood of collision with obstacles where no intersection is found. While this is shown to be effective in earlier examples, the formulation of the optimal region is shown here to be unable to guarantee that the $\boldsymbol{v}_{i,k+1}$ will optimally increase separation with large obstacle sets. This is also supported by the lower mean minimum separation distance of $0.5683m$.

Under the presented sensor assumptions, it is demonstrated how without additional filtration steps, none of the approaches are able to guarantee collision free trajectories in the presented conflict scenario. More recent VO derivatives however; such as the HRVO and ORCA methods, are shown to be more effective in maintaining separation than the earlier VO and RVO techniques. Examining the computational times given in Table 6.8, we can see that the VO derivatives scale more effectively with increasing obstacle numbers than the proposed IA method. The most computationally efficient method being the ORCA algorithm, where the mean minimum separation was also closest to the collision condition of $1m$ despite the sensor distortion.

6.6.4 Scalability Analysis

In the context of multi-agent collision avoidance, the scalability of routines for low-level operations such as collision avoidance is paramount. In support of data presented in Table 6.8 a scalability analysis of the IA method was conducted to examine the algorithm's performance through increasing agent populations.

As initially speculated in Section 6.6.3, the performance of the IA method is shown to deteriorate in collision scenarios involving more than six obstacles. This is indicated by the increasing mean number of collisions beyond $n = 6$ in Figure 6.23. Observing

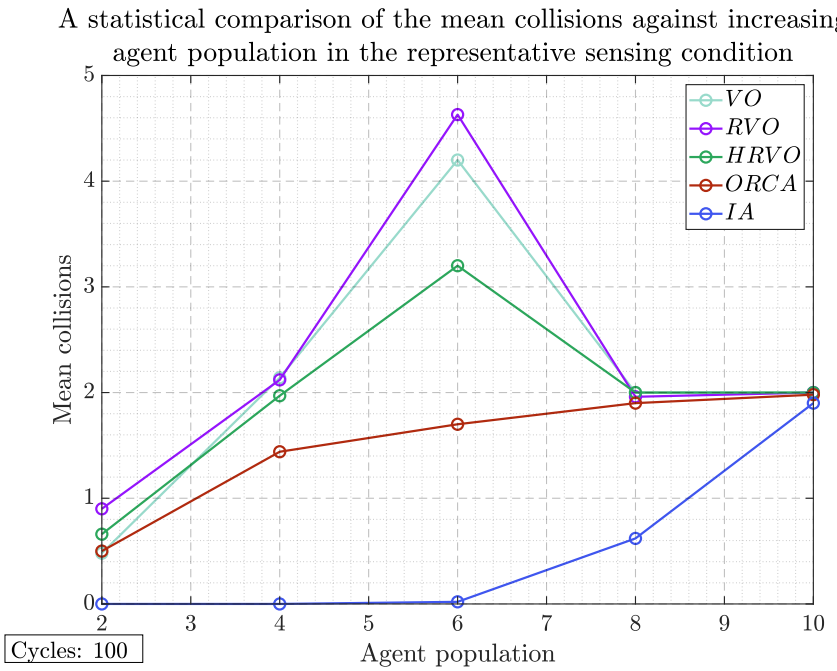


Figure 6.23: *The effect on increasing agent number on the IA algorithm's ability to maintain safe separation in the presented concentric collision scenario. The IA method is shown to be effective in maintaining separation for small (up to six agent) conflicts, however beyond this the performance begins deteriorate.*

the temporal analysis seen in Figures 6.24 and 6.25, it can be seen that the number of obstacles is strongly correlated with an increase in computation time. A comparison of the mean computation times to the preliminary analysis of the VO derivatives demonstrates that the IA algorithm scales similarly to the VO, RVO and HRVO methods. This said, the higher computational load of the IA algorithm indicates it is not as scalable for more cluttered environments despite it's better performance in maintaining safe separation for lower obstacle numbers.

6.7 Conclusions

In this chapter several established geometric approaches to collision avoidance are presented in the form of a technical review. The methodology of each approach is presented and discussed. A quantitative analysis is also presented, examining the effects of increasing both agent number and sensor uncertainty on the ability for each method to safely avoid collision.

In the preliminary analysis of the collision cone derivatives, uncertainty in obstacle trajectory is shown to increase the mean computation time of all the proposed approaches without compensative measures. Of the VO methods, the HRVO and ORCA approaches are shown to be more effective in negotiating obstacle cluttered environments whilst en-

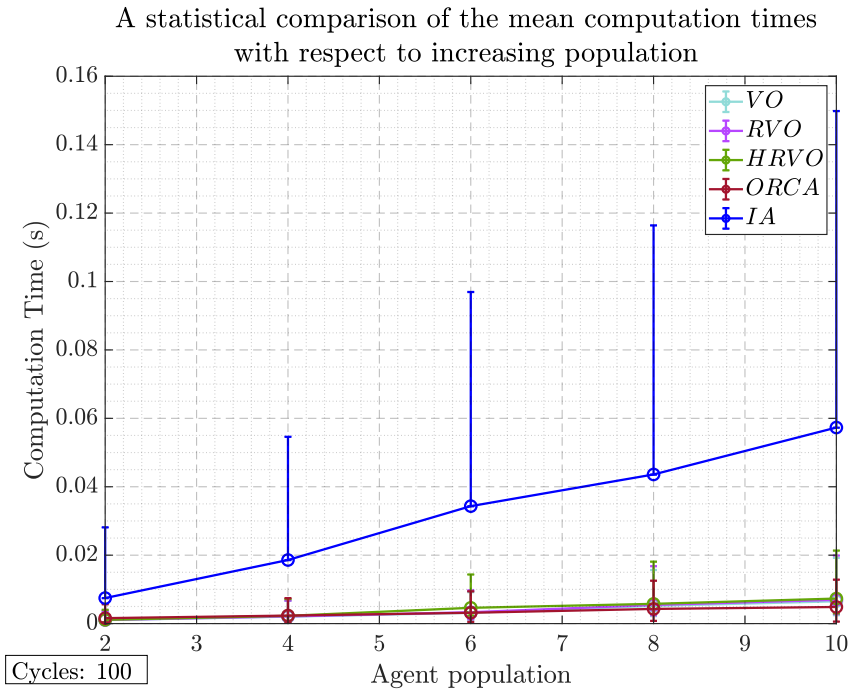


Figure 6.24: *A comparison between the mean computation times of the interval avoidance method with the increase of agent population. The errors bars demonstrate the variation between mean-minimum and mean-maximum computation time of each method. Heres, the interval method is shown to scale less effectively when compared to conventional VO derivatives. The ORCA algorithm is shown to be the most efficient with respect to increasing obstacle number.*

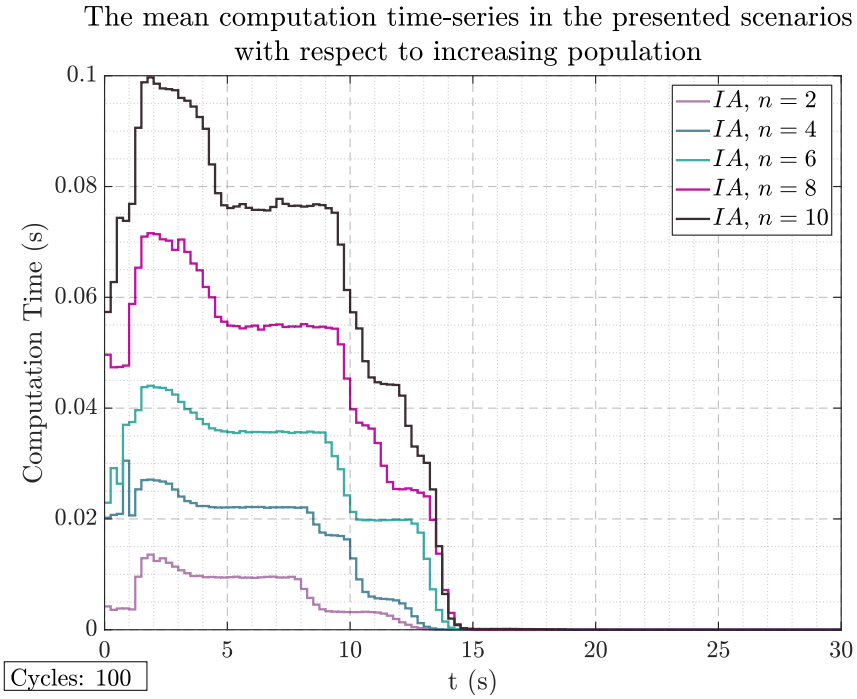


Figure 6.25: *A depiction of the effect of increasing agent population in the presented concentric scenario on the mean computational time-series. Here the change in mean response time is seen to peak as the agents resolve their initial heading direction and plateau as the agents move through the conflict centre.*

during uncertainty in obstacle trajectory. The ORCA method is also shown to generate both smoother resolution trajectories than the other presented methods despite low tolerance to obstacle uncertainty. The HRVO is shown to be statistically competitive with the ORCA in likelihood of collision, with higher tolerance to obstacle uncertainty. The benefit of the ORCA approach can clearly be seen in it's scalability; yielding computation times distinctly lower than the other methods.

The presented analysis of the state of the art is used to provide context for the proposed IA method. It is shown how corrupted trajectory measurements taken from local sensors may be used to propagate a region describing the set of trajectories needed to optimally avoid an obstacle. It is also shown how by considering the intersection of multiple optimal regions, avoidance of multiple objects can be achieved simultaneously. The proposed method is presented in a series of representative conflict scenarios where it is evaluated with respect to several key performance parameters. The IA algorithm is shown to be effective in avoiding collision in the presence of corrupt sensor measurements and multiple obstacles. This is evidenced by a one thousand cycle Monte-Carlo analysis of two presented scenarios whereby no collisions occurred.

A similar analysis of several popular geometric approaches demonstrated the IA method has a distinct advantage in lower obstacle numbers where trajectory uncertainty is present. Simulation of the IA method in scenarios with higher obstacle numbers demonstrated a reduction in performance. This is shown be a result of the strategy applied to the trajectory selection from optimal region and shall be the focus of future work in multiple obstacle avoidance. A statistical comparison of the methods demonstrated that the IA method, while more effective in less cluttered scenarios, is less scalable than the VO problem formulation. This is later supported by a scalability analysis of the algorithm which examines it's performance over increasing agent populations.

The representation of obstacle trajectory uncertainty intervals presents some interesting opportunities in dynamic, non-cooperative and uncertain conflict scenarios. It is clear that further investigation into the design of optimal regions as a result of higher obstacle numbers is needed. In these works, inter-agent avoidance is emphasised as the main cause of conflict in scenarios analogous to swarm operations and conventional air-traffic control. To be applicable to MAVs operating at low altitude, effective representation of more complex obstacle structures will also be the subject of future work. Finally, the IA method is initially presented under the assumption that the vehicles are restricted to a defined plane. The question of how the approach may be abstracted to facilitate avoidance in 3D space can then be seen as the subject of Chapter 7.

Chapter 7

Interval Avoidance in Three Dimensions

Up until now the motion of the *micro aerial vehicles* (MAVs) has been confined to a common plane under the assumption that avoidance is occurring at a set altitude. In this chapter we extend the *Interval Avoidance* (IA) method introduced in Chapter 6 to facilitate avoidance in *three dimensions* (3D). An adapted sensor model is presented alongside our algorithm to identify the optimal avoidance region from the measurement uncertainty. The performance of the algorithm is then examined in several proposed scenarios representative of typical MAV operations.

7.1 Sensor Model

Consider the scenario where agents i and j, are moving through three dimensional (3D) Cartesian space with global velocities $\boldsymbol{v}_i \in \mathbb{R}^{3\times 1}$ and $\boldsymbol{v}_j \in \mathbb{R}^{3\times 1}$ and positions $\boldsymbol{p}_i \in \mathbb{R}^{3\times 1}$ and $\boldsymbol{p}_j \in \mathbb{R}^{3\times 1}$ respectively. Similar to the assumptions given in Section 6.1.1, we assume the physical geometry of either agent can be contained within a spherical volume defined by representative radii r_i and r_j respectively (see Figure 7.1).

The sensor model facilitating the collection of obstacle trajectory data is initially introduced in Section 6.1.1 and is later adapted for the propagation of uncertainty intervals in Section 6.5.2. To represent obstacles in 3D space, the assumption of a camera and range-finder sensor is extended to include a relative elevation measurement $\tilde{\theta}_j = \mathcal{N}(\theta_j, \sigma_c^2) \in [-\pi, \pi]$ at time t_k. This is assumed possible using the same on-board camera and range-finder [63, 172] utilising the vertical pixel position.

Reintroducing the sensor confidence assumptions from Section 6.5.2, we present the

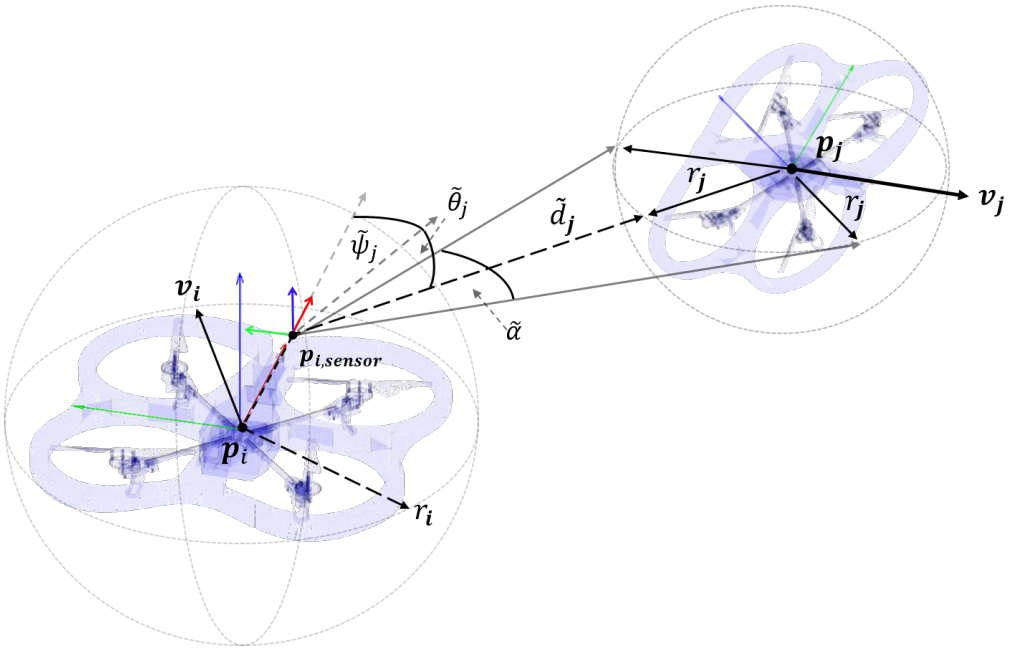


Figure 7.1: *The position $\boldsymbol{p}_{i,sensor}$ of the sensor model in the frame of i used in the detection of the relative heading, elevation, proximity and angular extents of j as $\tilde{\psi}_j$, $\tilde{\theta}_j$, $\tilde{d}_j$ and $\tilde{\alpha}_j$ respectively. The spherical volumes constraining the physical geometry of i and j can be seen defined by r_i and r_j. The obstacle's relative Cartesian position and velocity, $\boldsymbol{p}_j$ and $\boldsymbol{v}_j$ respectively, are calculated from its spherical position at time t_k.*

measurements as bounded uncertainty intervals in relative elevation $[\tilde{\theta}_j] = [\tilde{\theta}_j - 3\sigma_c, \tilde{\theta}_j + 3\sigma_c]$, heading $[\tilde{\psi}_j] = [\tilde{\psi}_j - 3\sigma_c, \tilde{\psi}_j + 3\sigma_r]$, range $[\tilde{d}_j] = [\tilde{d}_j - 3\sigma_r, \tilde{d}_j + 3\sigma_r]$ and angular extents $[\tilde{\alpha}_j] = [\tilde{\alpha}_j - 3\sigma_c, \tilde{\alpha}_j + 3\sigma_c]$. The obstacle's equivalent Cartesian position $\boldsymbol{p}_{k,j}$ at time step t_k, given the sensor's relative position $\boldsymbol{p}_{i,sensor}$ and orientation $\mathbf{R}_{i,sensor}$ in the body frame, is then given in Equation (7.1):

$$[\boldsymbol{p}_{k,j}] = \begin{bmatrix} [p^x_{k,j}] \\ [p^y_{k,j}] \\ [p^z_{k,j}] \end{bmatrix} = \mathbf{R}^T_{i,sensor} \left(\begin{bmatrix} \cos([\tilde{\psi}_{k,j}]) \cdot \cos([\tilde{\theta}_{k,j}]) \\ \sin([\tilde{\psi}_{k,j}]) \cdot \cos([\tilde{\theta}_{k,j}]) \\ \sin([\tilde{\theta}_{k,j}]) \end{bmatrix} \cdot [\tilde{d}_{k,j}] \right) - \boldsymbol{p}_{i,sensor} \quad (7.1)$$

In practice we assign the sensors relative pose to be aligned with the body axes of i for simplicity and so $\mathbf{R}_{i,sensor} = \mathbf{I} \in \mathbb{R}^{3\times3}$ and $\boldsymbol{p}_{i,sensor} = [0,0,0]^T$. The obstacle's Cartesian velocity is again inferred by the discrete differencing of the obstacle's position $\boldsymbol{v}_{k,j} = \frac{1}{\Delta t}(\boldsymbol{p}_{k,j} - \boldsymbol{p}_{k-1,j})$ where $\Delta t = t_k - t_{k-1}$. The angular width α_j at t_k can similarly be used to propagate an interval describing the obstacle's radius $r_{k,j} \in [r_{k,j}] = [\underline{r}_{k,j}, \overline{r}_{k,j}]$ using Equation (6.8), calculated from the geometric parameters shown in Figure 7.1.

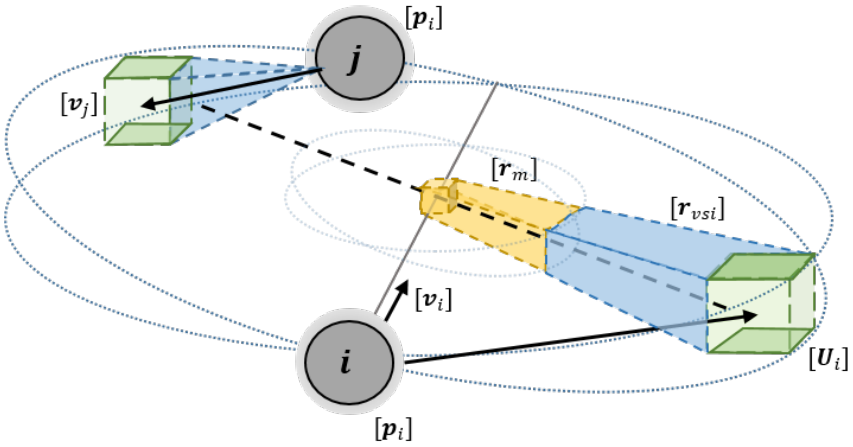


Figure 7.2: *A depiction of a 3D collision scenario between agents i and j. The interval avoidance problem is framed across the conflict planes described by $[\boldsymbol{n}^c]$. The optimal region $[\boldsymbol{U}_i]$ for agent i is shown propagated from a region containing the set of possible miss vectors $[\boldsymbol{r}_m]$ and uncertainty in the required correction vectors $[\boldsymbol{r}_{vsi}]$.*

7.2 Definition of an Arbitrary Conflict Plane

Collectively, the i's current knowledge of obstacle j can be expressed as intervals encapsulating it's true position, velocity and radius; $[\boldsymbol{p}_{k,j}]$, $[\boldsymbol{v}_{k,j}]$ and $[r_{k,j}]$ respectively at time t_k [63]. In Section 6.5.3, the position and velocity of j are represented in a common plane (XY) in which an avoidance manoeuvre must be resolved. From Figure 7.2 we can see that the definition of the miss interval $[\boldsymbol{r}_m]$ can be abstracted to a set of "conflict" planes defined by the inner product of the interval vectors $[\boldsymbol{n}^c] = [\boldsymbol{p}_j] \times [\boldsymbol{v}_j]$ where $[\boldsymbol{n}^c]$ is the vector interval containing the uncertainty in the true normal vector $\boldsymbol{n}^c$.

In Section 6.5.3, the miss interval $[\boldsymbol{r}_m]$ is related perpendicularly to $[\boldsymbol{v}_j]$ such that $[\boldsymbol{v}_j] \cdot [\boldsymbol{r}_m] = 0$. The cross product of the planar normal and the velocity uncertainty, $[\boldsymbol{n}^c]$ and $[\boldsymbol{v}_j]$ respectively, can then be said to describe the 3D miss interval $[\boldsymbol{r}_m]$ uncertainty as shown in Expression (7.2).

$$[\boldsymbol{r}_m] = [\hat{\boldsymbol{v}}_j] \times ([\boldsymbol{p}_j] \times [\hat{\boldsymbol{v}}_j]) \tag{7.2}$$

Given our uncertainty in the true trajectory of j, $[\boldsymbol{r}_m]$ naturally describes a region of possible miss distances $[r_{ca}]$ in the set of planes described by $[\boldsymbol{n}^c]$ as a result of the relative velocities $[\boldsymbol{v}_j]$. A 3D representation of the interval problem and the definition of $[\boldsymbol{r}_m]$ can be seen in Figure 7.2. We know from the geometric relationship between $[\boldsymbol{r}_m]$ and $[\tau]$ (see Figure 6.13), that Equation (6.11) may again be applied to $[\boldsymbol{n}^c]$ to define our uncertainty interval in the time to closest approach $[\tau] = -([\boldsymbol{p}_j] \cdot [\boldsymbol{v}_j])/([\boldsymbol{v}_j] \cdot [\boldsymbol{v}_j])$ using our 3D description.

7.3 Constraining Planar Separation

It is known from the definition of the resolution interval $[r_{res}] \in \mathbb{R}^1$ in Section 6.5.4 that Equation (6.13) poses a constraint on the radial separation of i and j. This property is characterised by our uncertainty in the miss distance $|||[\boldsymbol{r}_m]|||$ given $[r_j]$, $[\boldsymbol{v}_j]$ and $[\boldsymbol{p}_j]$ at time t_k. We know that geometrically $\boldsymbol{p}_i$ is common to $\boldsymbol{n}^c \forall [\boldsymbol{n}^c]$ as seen in Figure 7.2. Given that the constraint is radial to $\boldsymbol{p}_i$ then all values of $|||[\boldsymbol{r}_m]|||$ must satisfy Equation (6.13) to guarantee separation across $[\boldsymbol{n}^c]$. The resulting expression for our uncertainty in the required resolution magnitude again takes the form of $[r_{res}] = [r_{safe}] - |||[\boldsymbol{r}_m]|||$ given the definition of $[\boldsymbol{r}_m]$ in Equation (7.2).

7.4 Shared Resolution Volumes

With the velocity of i and j known to belong to the defined measurement intervals $[\boldsymbol{v}_i]$ and $[\boldsymbol{v}_j]$ we aim to construct a geometric region describing the optimal trajectory of i to avoid collision with j. In these works, it is assumed that agent j is also an intelligent agent that will act to avoid i as originally stated in Section 6.5.6. It is assumed that there is no communication between i and j, and so magnitude of $\boldsymbol{a}_{k,i}$ is subject to strategy.

We know from the relationship between $[\boldsymbol{r}_m]$ and $\hat{\boldsymbol{a}}_i$ in Section 6.5.5 that the optimal direction of pass is identified by the condition $\boldsymbol{r}_m \parallel \hat{\boldsymbol{a}}_i$. By their relation to $[\boldsymbol{r}_m]$, the associated accelerations $[\boldsymbol{a}_i]$ inherently describe a manoeuvre within the conflict plane. Application of Equation (6.16) to the 3D definition of $[\hat{\boldsymbol{r}}_m]$ then allows us to construct a 3D region $[r_{vsi}]$ that is scaled in accordance to the principle of *vector sharing* (VS) [187] and our uncertainty in the resolution magnitude $[r_{res}]$ (see Figure 7.2). The resulting region $[\boldsymbol{U}_i]$ enclosing the optimal position of i at τ is then assembled geometrically as $[\boldsymbol{U}_i] = [\boldsymbol{v}_i] \cdot [\tau] + [\boldsymbol{r}_{vsi}]$ where $[\boldsymbol{r}_{vsi}]$ contains the set of necessary corrections applied by i.

It is demonstrated how multiple obstacles may be considered in planar avoidance in Chapter 6 by defining a region of avoidance that is common to obstacles $j = 1 : n$. This relationship is defined by the prioritised intersection of their optimal avoidance regions $[\boldsymbol{U}^*] = [\boldsymbol{U}_i^*] \cap [\boldsymbol{U}_j]_{j=1:n}$ as presented in Section 6.5.7. Given that $[\boldsymbol{U}^*]$ now describes a region that would optimally increases separation with subset of $O_{j=1:n}$, we then define it's centre $\boldsymbol{U}^* = mid([\boldsymbol{U}^*])$ as the best estimate of the required correction manoeuvre for actuation.

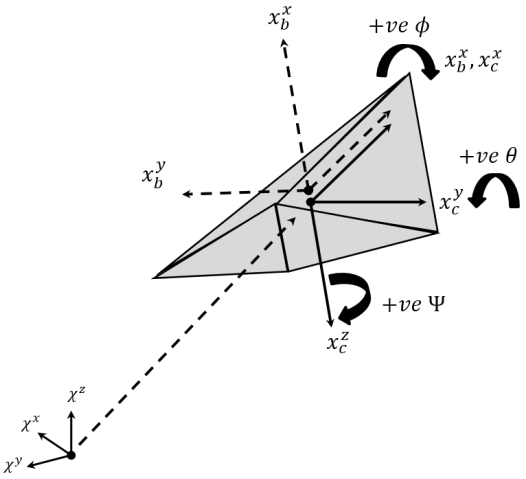


Figure 7.3: *Definition of the local sensory axes* east-north-up *(ENU) of agent i and it's corresponding control NED control axes (x_c assumed aligned with the inertial axes. Here ϕ, θ and ψ denote deflection about the local $\boldsymbol{x}_c^x$,$\boldsymbol{x}_c^y$ and $\boldsymbol{x}_c^z$ axes respectively.*

7.5 Agent Dynamics & Control

In the previous sections we demonstrate how we obtain a region of viable avoidance $[\boldsymbol{U}^*]$ trajectories given the set of obstacles $O_{j=1:n}$ observed at time t_k. Using the sensor model proposed in Section 7.1, we define the best estimate in the desired waypoint heading as $\boldsymbol{v}_{\text{pref}} = mid([\boldsymbol{v}_{i,\text{pref}}])$, from the observable waypoint set $W_i \in W_{k=1:n}$. The overview of the behaviour of agent i can also be seen in Algorithm 2.

It is shown in Figure 7.1 that these observations are made within a coordinate system defined by $\mathbf{R}_{i,sensor}$, with origin $\boldsymbol{p}_{i,sensor}$, that is assigned to the sensor. Similar assumptions may also be necessary to allow us to represent the trajectory within the axes of an on-board flight controller.

7.5.1 Axes Conventions

Depending on the control methodology applied on board of agent i, it may be necessary to express the trajectory interval $\boldsymbol{v}_{i,\text{pref}}$ in a secondary control axes. A common example of this is the standard *north-east-down* (NED) convention typically used in the aerospace sector and can be seen in our affiliated works [63, 64]. The relation between the two axes conventions can be seen in Figure 7.3 where distinction is made between the different coordinate frames for the body and control, sub-scripted by b and c respectively.

$$\mathbf{R}_{b\to c} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\frac{\pi}{2}) & -sin(\frac{\pi}{2}) \\ 0 & sin(\frac{\pi}{2}) & cos(\frac{\pi}{2}) \end{bmatrix} \tag{7.3}$$

The agent's target vector $\boldsymbol{v}_{\text{pref}}$ may then be mapped into the aircraft's dynamical axes by the constant matrix transformation $\mathbf{R}_c^b$ given in Equation (7.3). In the event that the sensor coordinate frame and control coordinate frames align then intuitively becomes $\mathbf{R}_c^b = \mathbf{I}_{3\times 3}$.

7.5.2 Euler Heading Generation

Using Equation (7.3) we are able to express the target velocity $\boldsymbol{v}_{i,\text{pref}}$ as a vector within the control axes of agent i. In this thesis, the pose of agent i is described by the Euler pose $\boldsymbol{\eta}$ at time t_k in accordance to state convention established in Chapter 3. It is therefore necessary to express $\boldsymbol{v}_{i,\text{pref}}$ in terms of a relative angular projections. These arguments can be seen calculated from $\boldsymbol{v}_{i,\text{pref}}$ in Equations (7.4)-(7.7); as projections on the body axes XY and XZ planes.

$$\boldsymbol{v}_V = \hat{\boldsymbol{v}}_{\text{pref}} - \left(\frac{\hat{\boldsymbol{v}}_{\text{pref}} \cdot \hat{\boldsymbol{n}}^{xz}}{||\hat{\boldsymbol{n}}^{xz}||^2} \right) \cdot \hat{\boldsymbol{v}}_{\text{pref}} \tag{7.4}$$

$$\boldsymbol{v}_H = \hat{\boldsymbol{v}}_{\text{pref}} - \left(\frac{\hat{\boldsymbol{v}}_{\text{pref}}] \cdot \hat{\boldsymbol{n}}^{xy}}{||\hat{\boldsymbol{n}}^{xy}||^2} \right) \cdot \hat{\boldsymbol{v}}_{\text{pref}} \tag{7.5}$$

$$\theta = \tan^{-1} \left(\frac{\boldsymbol{v}_V}{||\boldsymbol{v}_H||} \right) \tag{7.6}$$

$$\psi = sign\left(\boldsymbol{v}_{\text{pref}} \times \boldsymbol{v}_H\right) \cdot \cos^{-1} \left(\frac{\boldsymbol{v}_H \cdot \boldsymbol{v}_{\text{pref}}}{||\boldsymbol{v}_H||} \right) \tag{7.7}$$

Here we define the vectors $\hat{\boldsymbol{n}}^{xy} = [0,0,1]^T$ and $\hat{\boldsymbol{n}}^{xz} = [0,1,0]^T$ to be the normals to planes XY and XZ respectively. The direction of rotation is found by determining the *sign* of the rotation axis $\boldsymbol{v}_{\text{pref}} \times \boldsymbol{v}_H$.

7.5.3 3D Single Integrator Systems

In the literature [63, 251], a common representation of the state evolution of agent i is a single integrator. We introduce this assumption to the planar pose of agent i in Section 6.3.2. It is demonstrated how more sophisticated agent descriptions may be formed in Chapters 3 and 4. In the form of a state vector, a 6DOF single integrator system is proposed for simplicity, defined by the same state (7.8):

$$\boldsymbol{x}_{k,i} = \begin{bmatrix} \boldsymbol{p}_{k,i} \\ \boldsymbol{\eta}_{k,i} \end{bmatrix} = \begin{bmatrix} x_{k,i} & y_{k,i} & z_{k,i} & \phi_{k,i} & \theta_{k,i} & \psi_{k,i} \end{bmatrix}^T \tag{7.8}$$

To be able to represent a generic holonomic MAV moving through 3D space, i's state is said to describe its instantaneous position and Euler pose $\boldsymbol{x}_{k,i} = [\boldsymbol{p}_{k,i}, \boldsymbol{\eta}_{k,i}]^T$ at time t_k.

Parameter	Velocity Bounds	Parameter	Acceleration Bounds
$\dot{\phi}$	$\dot{\phi} \in [\dot{\phi}_{min}, \dot{\phi}_{max}]$	$\ddot{\phi}$	$\ddot{\phi} \in [\ddot{\phi}_{min}, \ddot{\phi}_{max}]$
$\dot{\theta}$	$\dot{\theta} \in [\dot{\theta}_{min}, \dot{\theta}_{max}]$	$\ddot{\theta}$	$\ddot{\theta} \in [\ddot{\theta}_{min}, \ddot{\theta}_{max}]$
$\dot{\psi}$	$\dot{\psi} \in [\dot{\psi}_{min}, \dot{\psi}_{max}]$	$\ddot{\psi}$	$\ddot{\psi} \in [\ddot{\psi}_{min}, \ddot{\psi}_{max}]$
v^x	$v^x \in [v^x_{min}, v^x_{max}]$	a^x	$a^x \in [a^x_{min}, a^x_{max}]$

Table 7.1: *The representative kinematic constraints for a six degree of freedom holonomic MAV with input actuation limits. Here the first and second order kinematic states x are bound between defined minimums and maximum absolute values, x_{min} and x_{max}) respectively, in order to represent actuator saturation.*

The discrete representation of the system can then be written as Equation (7.9) and is analogous to the one introduced in Section (6.3.2).

$$\begin{bmatrix} \boldsymbol{x}_{k+1,i} \\ \dot{\boldsymbol{x}}_{k+1,i} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{6\times6} & \Delta t \cdot \mathbf{I}_{6\times6} \\ \mathbf{0}_{6\times6} & \mathbf{I}_{6\times6} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{k,i} \\ \boldsymbol{u}_{k,i} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{6\times1} \\ \boldsymbol{w}_{k,i} \end{bmatrix} \tag{7.9}$$

Here $\boldsymbol{w}_{k,i} = \mathcal{N}(\mathbf{0}_{6\times1}, \mathbf{I}_{6\times6} \cdot \sigma_q^2)$ defines the state noise vector, where σ_q denotes the standard deviation of the state-specific noise signal.

7.5.4 Dynamical Constraints

Constraints on the physical capabilities of agent i are applied by defining the maximal allowable inputs. Under the convention of a single integrator, these changes are synonymous to the agent's state differentials [63,66]. This is done to emulate the saturation of the agent's control inputs whilst being consistent with the Euler state assumptions presented in Section 7.5.3. The representative actuator constraints on the states of agent i are therefore presented in Table 7.1. To represent a system that is kinetically holonomic in the proceeding examples, the limits are defined such that $q_{min} = -q_{max}$ where q is a kinematic state in Table 7.1.

7.6 Performance Evaluation

In [63], the 3D IA concept is applied to full scale *unmanned aerial vehicle* (UAV) operations. Within the context of this thesis, agents are assumed to be MAVs ($\leq 25kg$) operating in close proximity ($\leq 25m$) to one another moving at constant speed and *straight and level*(S&L). In this section the proposed algorithm is presented in several representative conflict scenarios.

Parameter	Value	Units
Sensor Range	15	m
Camera Standard Deviation (σ_c)	5.208×10^{-5}	rad
Range-finder standard deviation (σ_r)	0.1	m
Airspeed standard deviation (σ_v)	0.5	$m \cdot s^{-1}$
Position standard deviation (σ_p)	0.5	m
Measurement confidence	3	-
Sampling frequency	4	Hz
Neighbour horizon (d_{nei})	15	m
Maximum speed (v_{max})	4	$m \cdot s^{-1}$
Maximum turn rate ($\dot{\phi}_{max}, \dot{\theta}_{max}, \dot{\psi}_{max}$)	0.25	$rad \cdot s^{-1}$
Preferred speed (v_{pref})	2	$m \cdot s^{-1}$
Characteristic radius (r_i)	0.5	m
Event tolerance(ι)	1×10^{-3}	m

Table 7.2: *The environmental simulation parameters used in the proceeding example scenarios and performance evaluations.*

7.6.1 Experimental Conditions

The agents are assumed to have sensing capabilities sufficient to observe other agents and obstacles in the environment in accordance to the conditions presented in Table 7.2. The geometry of the MAVs are represented by a characteristic radius of $r_{i,j} = 0.5m$ with a nominal cruise speed of $2m/s \approx 4.5mph$. A simulation time step of $\Delta t = 0.2s$ is chosen as the repetition frequency of existing *Laser Obstacle Avoidance Marconi* (LOAM) sensor devices [37, 209].

The agents are initialised with no prior knowledge of their surroundings other than the location of their assigned waypoints W_i. Events such as collisions or waypoint incidence are said to occur when the following condition is violated $||\vec{p}_i - \vec{p}_{i,\text{wp}}|| < (r_i + r_{i,\text{wp}}) - \iota$, where the parameter ι is a condition tolerance that aims to eliminate ambiguity between collisions and narrow-misses caused by the nature of discrete simulation. The agent and scenario parameters are otherwise explicitly stated in Table 7.2.

7.6.2 Overlapping Flight Paths

The proposed IA algorithm is initially demonstrated in the context of a typical flight path conflict between two MAVs operating at the same altitude. The flight paths themselves are assumed to be finite segments of some unspecified global objective or coordinated task that necessitates their close proximity. A similar scenario representing a UAV conflict is presented in [63, 187]. The initial conditions for this scenario are given in Table 7.3.

Both agents are initialised within sensor range and tasked with moving from their initial positions (circles) and proceed towards their target waypoint (triangles) whilst

Initial Condition	Position (m)	Velocity (ms^{-1})	Euler Pose (rad)
MAV 1 (alpha001)	[-10.00,0.00,0.00]	[2.00, 0.00,0.00]	[0.00,0.00,0.00]
MAV 2 (beta001)	[-7.07,7.07,0.00]	[1.41,-1.41,0.00]	[0.00,0.00,0.79]

Table 7.3: *The initialisation conditions of the two MAVs in example one representing an overlapping flight path scenario.*

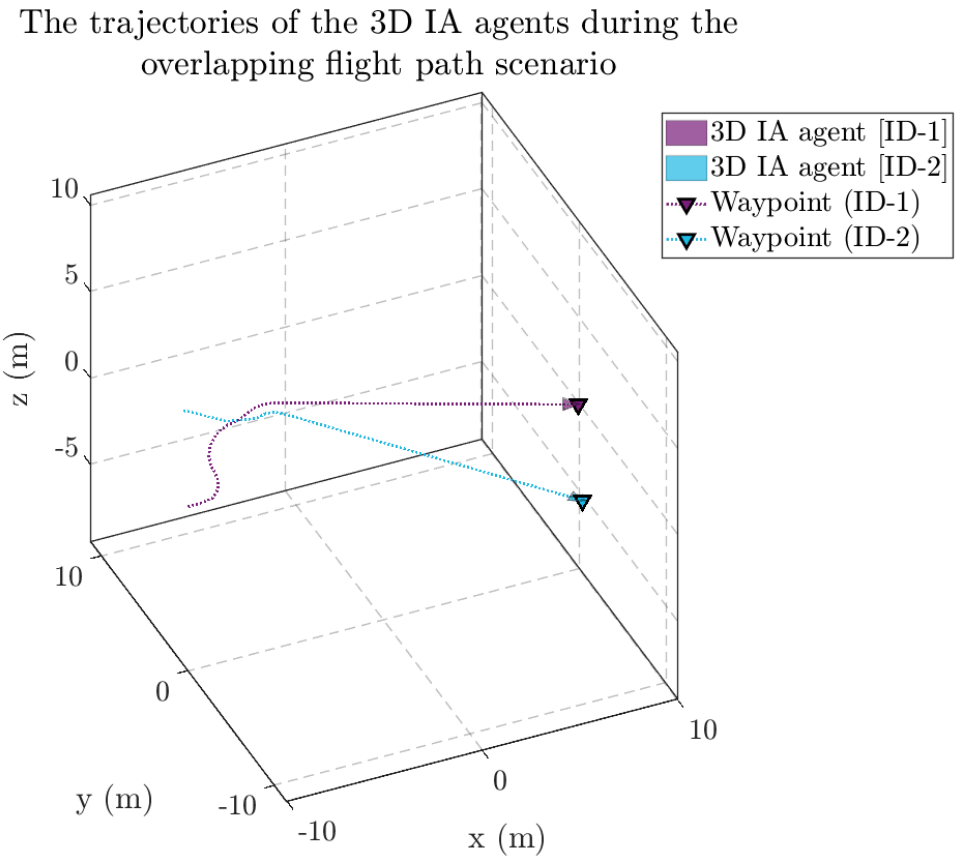


Figure 7.4: *An isometric view of the conflicting flight path example where two agents are engaged in a collision scenario defined in a common plane. Agent* 1 *can be seen to immediately climb in order to pass behind agent* 2*. Agent* 2 *initially resolves a climb trajectory also before returning to S&L in response to agent* 1*.*

avoiding collision (see Figure 7.4). The position of the waypoints W_1 and W_2, assigned to agents 1 and 2 respectively, ensure that a collision will occur unless the agent trajectories are corrected. Once agent 1 detects agent 2 it begins to evaluate the optimal avoidance region $[\boldsymbol{U}_{j,k}]$ for the current time step t_k. It is assumed the dynamics of the agent are fast acting (see Section 7.5.3) and so the trajectory changes are actuated immediately subject to the kinematic constraints introduced in Section 7.5.4.

In Figures 7.4 and 7.6 the agents are shown to resolve the conflict without collision by maintaining a minimum separation of $4.5m$ at the point of closest approach r_{ca}. The corresponding control inputs shown in Figure 7.5, indicate that the trajectory correction occurs immediately upon detection. Control oscillation can be observed to occur until

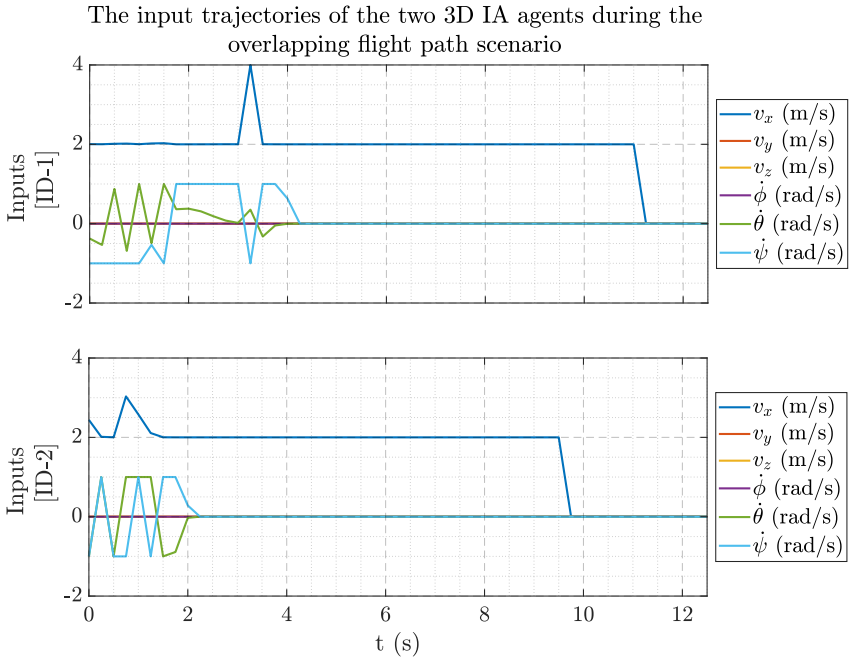


Figure 7.5: *The control inputs corresponding to agents* 1 *and* 2 *over the course of the overlapping flight path scenario. Here both agents are shown to oscillate initially before the direction of pass is resolved. The agents make smaller adjustments to their 3D headings up to the point of closest approach* r_{ca} *before briefly accelerating as the waypoint trajectory becomes available.*

the direction of pass is resolved, whereby agent 1 moves behind agent 2. This can be seen reflected in the mirrored pitch and yaw responses of the two agents for $t \leq 4.2s$. Once collision condition $\overline{\tau}_j < 0$ is satisfied, the two agents can then be seen reorienting themselves toward their designated waypoint and revert to their nominal cruise condition.

7.6.3 Direct Collision Scenario

Consider agents 1 and 2 to be on contradicting flight paths with a collision guaranteed by the placement of their assigned waypoints W_i and W_j. Such a scenario is designed to represent the "worst case" for *reflexive* collision avoidance algorithms due to the parallel nature of $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$. This is described by the initial condition $\boldsymbol{v}_i \times \boldsymbol{v}_j = 0 \therefore ||[\boldsymbol{r}_m]|| \approx 0$ [63]. This condition naturally states that the direction of pass in this scenario is ambiguous, but also requires a maximal change in trajectory $[r_{res}] = [r_{safe}]$ (see Section 6.5.6). The initial configuration for this scenario is given in Table 7.4.

Agent 1 evaluates the collision projection interval $[\tau_j]$ based on the sequential measurements of agent 2 and vice-versa. Due to the uncertainty in the measurements of agent 2, the avoidance routine is executed when any convergent behaviour $0 \geq \underline{\tau}_j$ exists between the agents. The uncertainty in the measured position of agent 2 is used in our method to

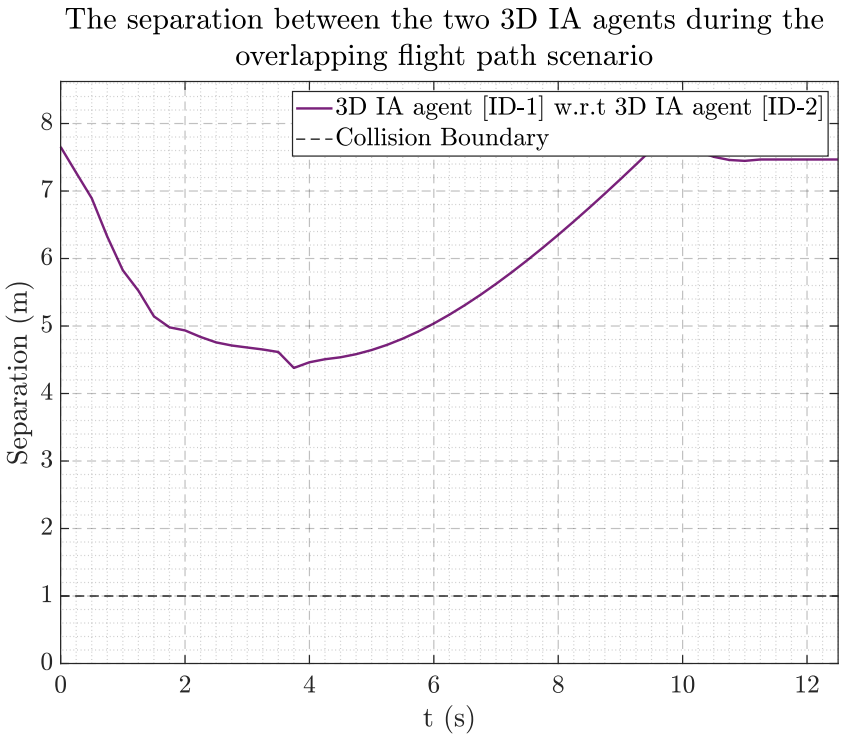


Figure 7.6: *The separations maintained between agents* 1 *and* 2 *during the overlapping flight path example scenario. The micro-adjustments of both agents as they approach* $\boldsymbol{r}_{ca}$ *can again be observed to increase as their proximity reduces. At* $t = 3.6s$ *the agents determine there is no chance of collision and immediately proceed towards their goals.*

Initial Condition	Position (m)	Velocity (ms^{-1})	Euler Pose (rad)
MAV 1 (alpha001)	[-10.00,0.00,0.00]	[2.00,0.00,0.00]	[0.00,0.00,0.00]
MAV 2 (beta001)	[10.00,0.00,0.00]	[-2.00,0.00,0.00]	[0.00,0.00,1.57]

Table 7.4: *Initial conditions of agents* 1 *and* 2 *in the second example where both agents are on parallel contradictory flight paths. Each agent is positioned with opposing headings, centred around the global origin with a radius* $10m$*, with an initial velocity of* $2m \cdot s^{-1}$.

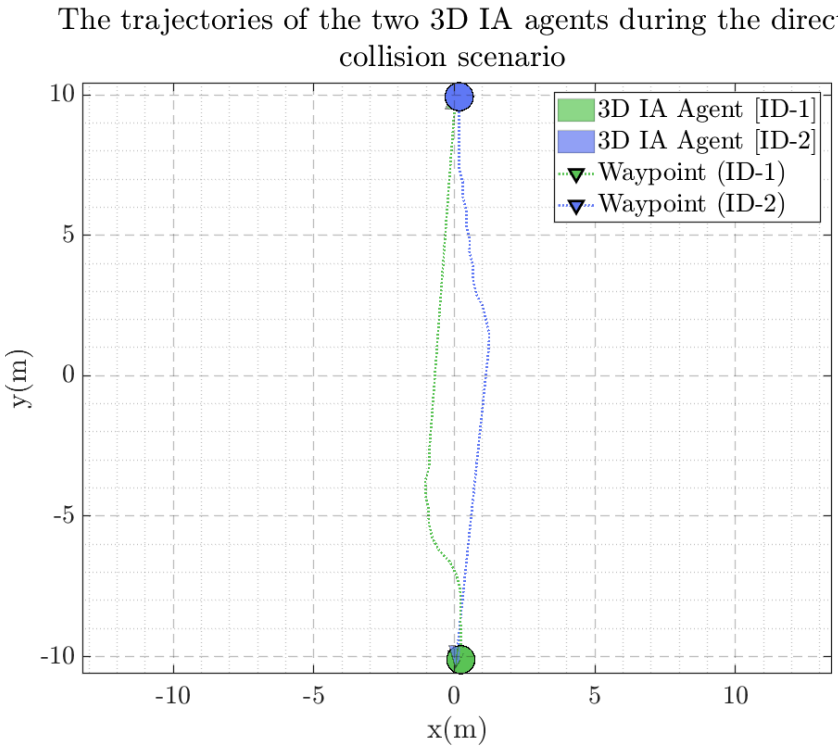


Figure 7.7: *Example trajectories of two MAVs negotiating a conflict with geometrically opposing flight paths at the same altitude. The agents are shown to proceed from their initial position (circles) and proceed to their assigned waypoints (triangles). Here both MAVs can be seen to resolve a direction of pass using their uncertainty in obstacle trajectory and continue to enlarge their separation until there is no opportunity for collision in the future.*

excite the change in trajectory of the two agents. This helps to disambiguate the direction of pass since $||\boldsymbol{r}_m|| \neq 0$. Once the direction of pass is established, $[\boldsymbol{U}_i]$ acts to extend the miss interval $[\boldsymbol{r}_m]$ until the collision condition $\boldsymbol{r}_{ca} = ||\boldsymbol{r}_m|| > \overline{r}_{safe}$ is satisfied.

In Figure 7.7 the two agents are shown to move from their initial positions and proceed to their assigned waypoints. Both agents are shown to successfully generate a non-zero miss distance $|||[\boldsymbol{r}_m]|||$ from initially co-linear trajectories. Due to the uncertainty in the trajectory measurements introduced in Section 7.1, adjustments to the trajectory of agents 1 and 2 can be seen up until the time of closest approach $t = 4.25s$. Here, the algorithm is shown to be able to exceed the minimal separation condition shown in Figure 7.8, with a value of $1.9m$. It can be deduced from Figure 7.9 that once $\underline{r}_{safe}$ has been generated, it is maintained through the point of closest approach $\boldsymbol{r}_{ca}$. This is ensured by the convergence condition $\overline{\tau}_j > 0$ that if convergent behaviour can exist then the agent will select from a new optimal region $[\boldsymbol{U}_{k,j}]$.

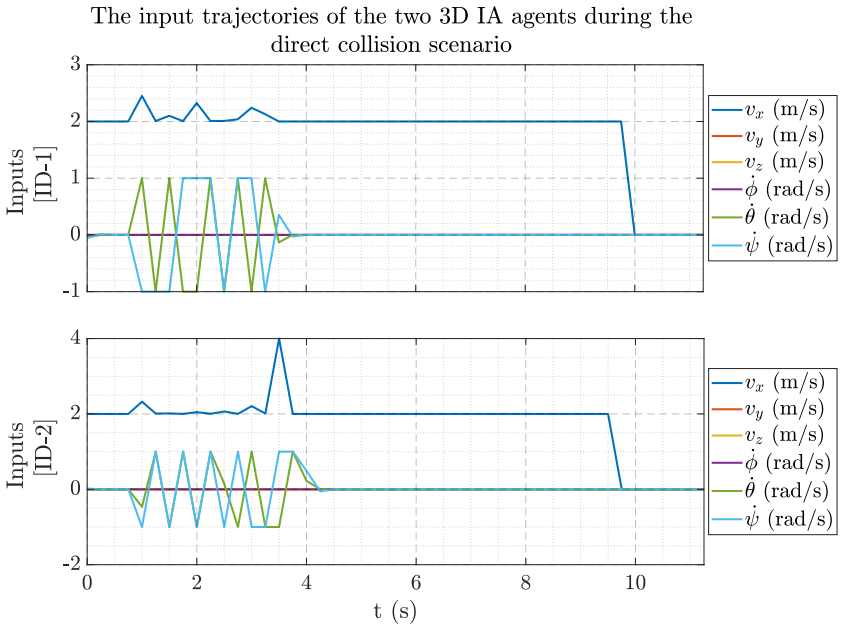


Figure 7.8: *The input trajectories of agents* 1 *and* 2 *during the conflicting flight path example. Both agents can be seen to oscillate as the direction of pass is resolved, with fewer adjustments made as resolution trajectory is found. As* r_{ca} *is reached, both agents accelerate onto their unimpeded waypoint trajectory.*

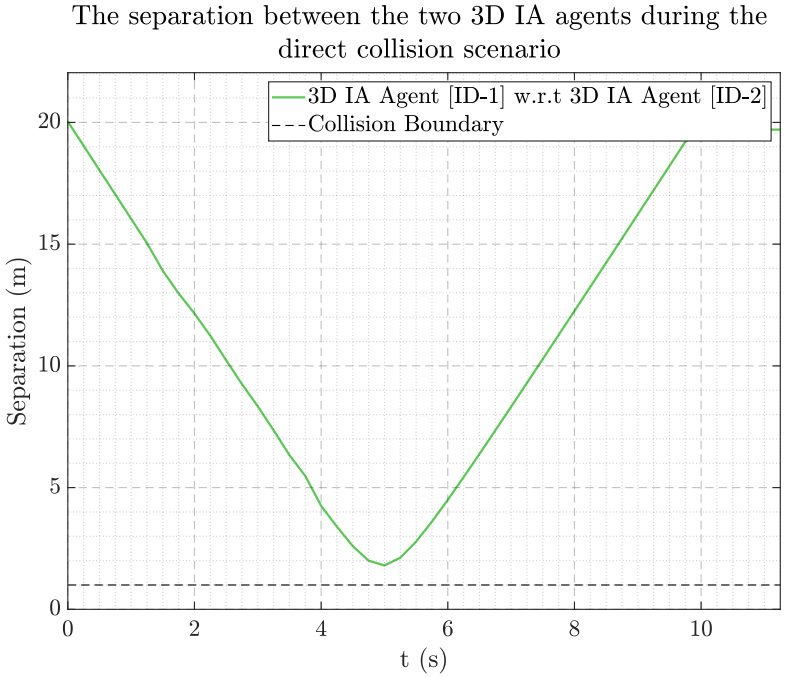


Figure 7.9: *The separation between agents* 1 *and* 2 *through the conflicting flight path example. Despite the initial zero miss distance condition, the IA algorithm can be seen to generate a separation of* 1.9m *at the point of closest approach.*

Initial Condition	Position (m)	Velocity (ms^{-1})	Euler Pose (rad)
MAV 1 (alpha001)	[-10.00,0.00,0.00]	[2.00, 0.00,0.00]	[0.00,0.00, 0.00]
MAV 2 (beta001)	[5.00, 8.66,0.00]	[-1.00,-1.73,0.00]	[0.00,0.00, 2.09]
MAV 3 (gamma001)	[5.00,-8.66,0.00]	[-1.00,1.73,0.00]	[0.00,0.00,-2.09]

Table 7.5: *The initial conditions for three agents defining the concentric collision scenario, centred around the origin with a radius of* $10m$ *and an initial agent velocity of* $2ms^{-1}$.

7.6.4 Multi-agent Concentric Conflict

In Section 6.5.7 it is shown how the representation of the geometric constraints as intervals allows us to consider multiple obstacles [63, 66]. To demonstrate this, we introduce a scenario where three agents are involved in the concentric encounter defined by the initial conditions in Table 7.5. Similar to the scenarios presented in [63, 111], agents 1 to 3 proceed from their initial positions toward their antipodal waypoint position $W_{1:3}$ (shown in Figure 7.10). By the position of the waypoints, a collision mutual to all agents is ensured. Each agent is assumed to be able to observe the trajectory of the other agents in accordance to the sensor model presented in Section 7.1 once within a defined range d_{nei}. Once initialised, each agent is then tasked with computing a mutual avoidance trajectory in the presence of the two other agents.

It is clear from Figure 7.12 that each agent was successful in mitigating collision with their respective aggressors. This is demonstrated by a minimal separation of $3m$ at $t = 5.6s$ between agents 2 and 3 whereby the agents revert to their nominal cruise conditions. Some oscillation can again be seen initially as the agents resolve their direction of pass and change from the combined optimal avoidance volume $[\boldsymbol{U}*]$. This is also reflected in the agent control inputs (see Figure 7.11) where the agents are seen to exert significant changes in pitch and yaw rate, $\dot{\theta}$ and $\dot{\psi}$, to keep their heading within the optimal region.

7.6.5 Scalability Analysis

A statistical analysis of the presented scenarios was undertaken to assess the algorithm's performance in a one thousand iteration Monte-Carlo simulation. In each scenario presented in Sections 7.6.2 to 7.6.4, the global positions and velocities of each object is perturbed by the noise signals $\boldsymbol{p}_{i,0} = \mathcal{N}(\boldsymbol{p}_i, \mathbf{I}_{3\times3} \cdot \sigma_p^2)$ and $\boldsymbol{v}_{i,0} = \mathcal{N}(\boldsymbol{v}_i, \mathbf{I}_{3\times3} \cdot \sigma_v^2)$ respectively. The event and temporal statistics observed over one thousand Monte Carlo independent runs are presented in Table 7.6.

Observing Table 7.6, we see that interval avoidance approach was able to attain safe separation in each of the presented cases emulating typical UAV encounters. The mean

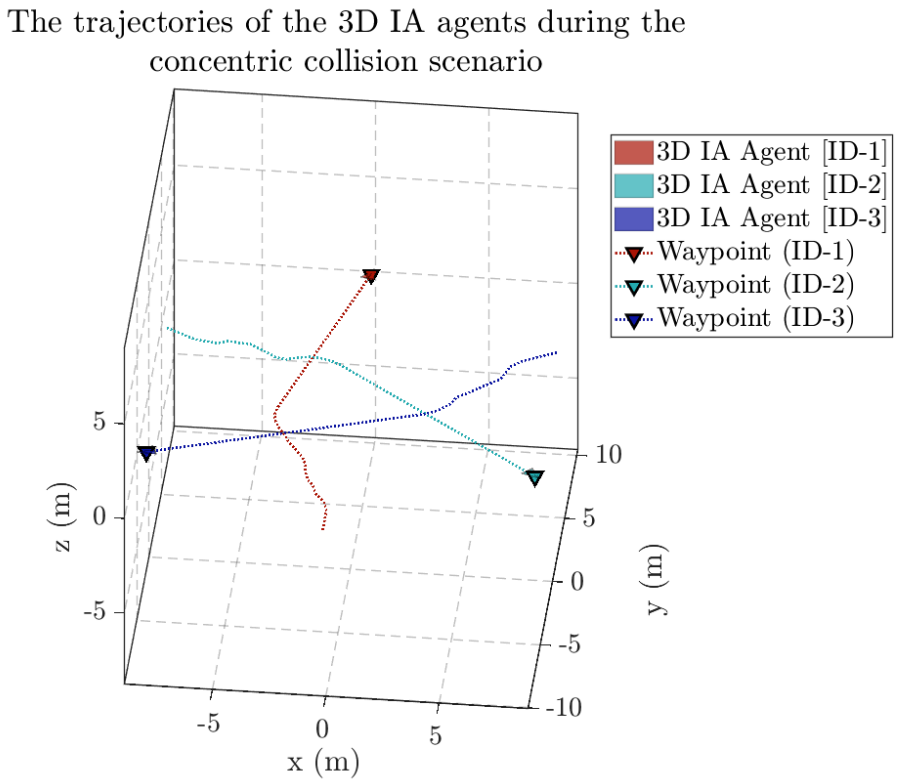


Figure 7.10: *An isometric view of a concentric collision scenario involving three agents operating at the same altitude. Each agent quickly resolves a direction of pass and begins to generate a non-zero miss distance with the other agents. The agents can be seen to oscillate as the shape of the mutual avoidance region is altered as the priority of the obstacle is re-evaluated.*

Mean Data	Overlapping Flight Paths	Direct Collision	Concentric Conflict
Agents	2.000	2.000	3.000
Collisions	0.000	0.000	0.000
Waypoints	1.982	1.988	2.964
Minimum Separation (m)	1.775	1.839	3.050
Maximum Separation (m)	29.103	29.182	25.873
Mean Computation Time (ms)	0.950	1.030	1.990
Minimum Computation Time (ms)	0.025	0.026	0.031
Maximum Computation Time (ms)	2.820	3.440	5.130

Table 7.6: *A comparison of the mean data from a 1000 cycle Monte-Carlo analysis for the three presented scenarios. The event, separation and computational time statistics in the scenarios are presented in which. Here the 3D IA algorithm is shown able to be effective in avoiding collision in each scenario, generating a minimum of* $1.775m$ *separation under the effects of obstacle and agent trajectory uncertainty.*

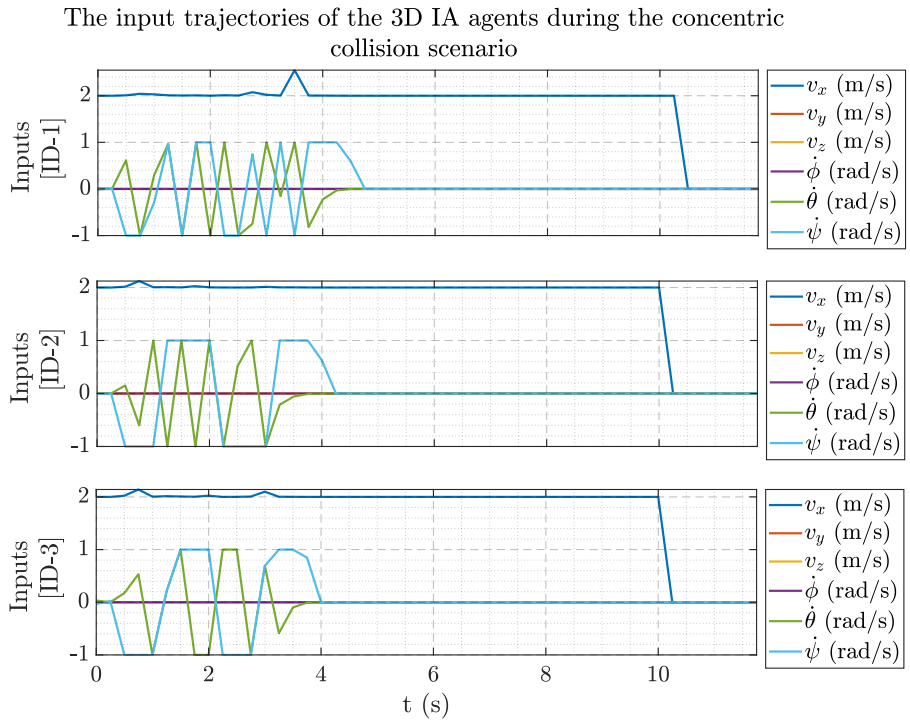


Figure 7.11: *The input trajectories of agents* 1 *to* 3 *during the three agent concentric collision scenario. The agents are again shown to make a series of adjustments in order to maintain a trajectory within the region of mutual avoidance. As the agents are no longer confined to a plane, the agents are able to command a heading and pitch rate simultaneously in order to increase their respective miss-distances more quickly.*

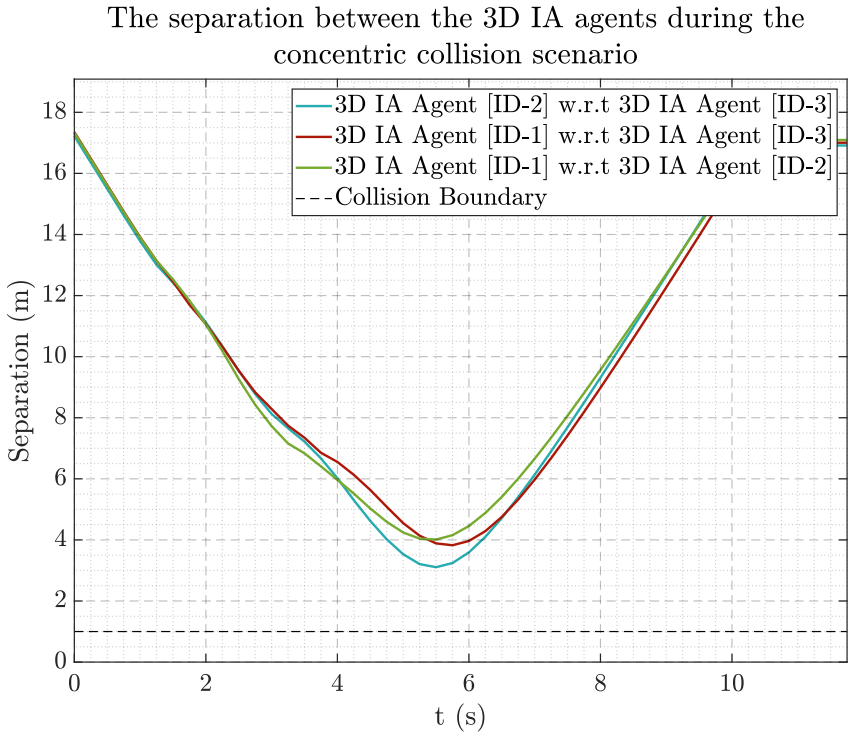


Figure 7.12: *The separations maintained between the three agents during the 3D concentric collision scenario. A minimum separation of* $3m$ *between agents* 2 *and* 3 *can be seen at the time of closest approach* $t = 5.6s$.

number of waypoints is shown to be on average less than the number of total waypoints in each condition. This suggests that in some cases the resolved avoidance trajectory, prevented the waypoint from being achieved within the fixed $60s$ time period.

The mean minimum separation is shown to well exceed the collision boundary of $r_i + r_j = 1m$ in each of the cases. In the most geometrically complex case, the direct collision scenario, the separation generated is 184% of the required safe distance. The mean computation times are shown to be similar between both the flight path conflict and direct collision scenarios. It is clear that the minimal miss distance condition in the direct collision scenario is the cause of a marginal increase in each of the timing parameters. This demonstrates that on average a greater number of computational cycles is necessary to resolve the conflict. The most significant separation can be seen in the multi-agent scenario, where the mean maintained separation was $3.050m$. The compromise seems to be when considering the computation times, which is seen to increase by a factor of 1.9 with the addition of a secondary obstacle. Further investigation may be necessary to illustrate the effects of increasing obstacle number on the algorithm's computation time.

Similar to the analysis presented in Section 6.6.4, the performance of the 3D IA algorithm is statistically evaluated with respect to increasing agent population under the conditions presented in Section 7.6. The performance of the two IA derivatives can be seen compared with respect to mean collision rate and mean computation time in Figures 7.13 and 7.14 respectively. Examining the effects of increasing agent number on the mean rate of collision in Figure 7.13, it is clear that the addition of the third dimension allows the 3D method to be more effective in maintaining safe separation in multi-agent conflicts with a greater number of agents. As seen in Chapter 6, the 2D IA method was demonstrated to be effective in mitigating collision in scenarios with up to $n = 6$ agents. Here however, it is demonstrated that the 2D method begins break down in scenarios with higher agent numbers but more significantly when the encounters are symmetrical (indicated by the higher collision rates on even numbers). In the same conditions however, the 3D method is shown to perform more predictably than the 2D IA approach as the number of agents exceeds $n = 6$. From this point the mean rate of collision is observed to be lower than the 2D IA method up until a population of $n = 10$ is reached. Here both methods are shown to be unable to determine avoidance trajectories that satisfy the complete obstacle set in order maintain safe separation.

Observing the relationship between computational time and agent number in Figure 7.14, its clear that the time required to compute the 3D IA approach scales similarly to the 2D IA approach. The effect of considering the third dimension is shown to result

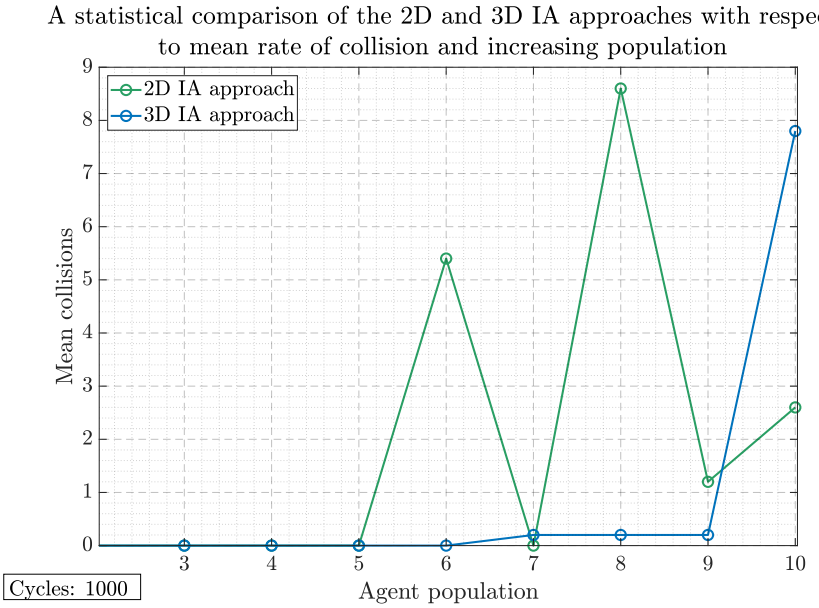


Figure 7.13: *A comparison of the mean rates of collision between the 2D(green) and 3D(blue) IA derivatives over 1000 Monte-Carlo cycles. The error bars indicate the variation between mean-minimum and mean-maximum computation times. Here, the availability of the third dimension is shown to allow the IA method to continue to avoid collision in scenarios with a greater number of agents. In addition to this, the Behaviour predictability extends beyond 6 agents, although collisions become more likely once ten agents are involved in the conflict.*

in an increase in computational overhead for each novel obstacle. This is seconded by the variability in computation time; described by the error-bars in Figure 7.14. The increase in computational cost is also shown to increase the margin between the mean minimum and maximum computation times as the cost of resolving an avoidance solution for any one obstacle is marginally higher. The 3D IA approach is shown to be effective in conflict scenarios involving more agents than the approach presented in Chapter 6, whilst also encompassing the effects of sensor corruption and trajectory uncertainty. The addition of the third dimension is however shown to extend the performance ceiling of the 2D IA method to $n = 9$. This is achieved by making use of trajectories outside of the XY plane, which also allows the solution space to better represent the array of trajectories available to modern UAS and UAVs.

7.7 Conclusions

In this chapter we extend the concept of an interval-geometric approach to reflexive avoidance to facilitate avoidance in 3D. It has been shown how the approach incorporates obstacle trajectory uncertainty and sensor confidence in the generation of optimal avoidance trajectories in both the singular and multiple agent cases. The proposed method

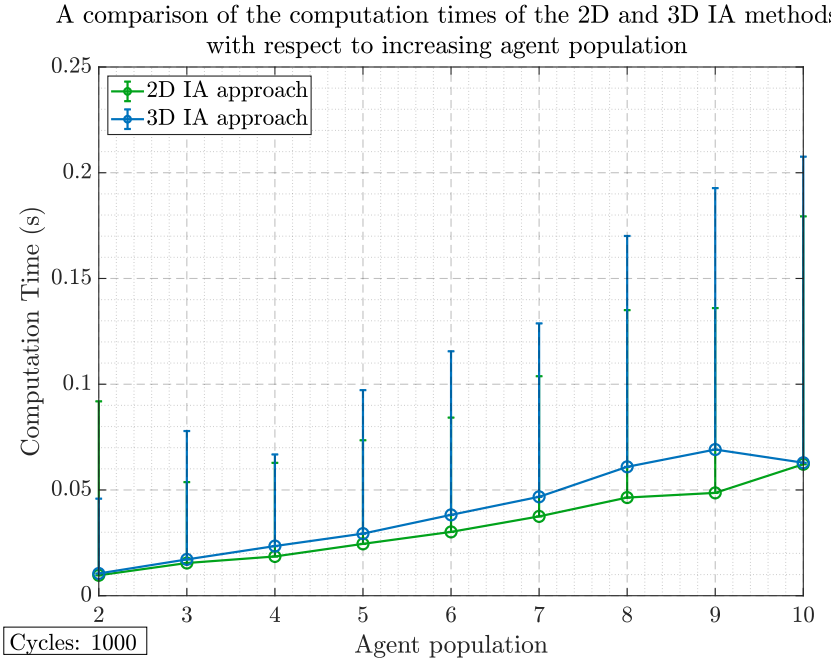


Figure 7.14: *A comparison of the variation in mean computation times between the 2D and 3D IA algorithms with respect to increasing agent population. The addition of the third dimension is shown to result in a increase in computation time that scales similarly to the 2D IA approach.*

is presented initially under the assumption that the MAVs behave as a 6DOF single integrator, as a reduced representation of the complex descriptions in Chapter 3.

Kinematic constraints are introduced to limit the MAV's performance window by emulating saturation of the aircraft's actuators. Such constraints confine the achievable body axis rates in order to better represent the dynamic limitations of a physical system without further assumptions on the propulsive mechanism.

The proposed 3D interval avoidance method is demonstrated in three representative aerial conflicts: a typical flight path conflict, a direct collision and a three agent concentric collision. In each case, the proposed method is shown to maintain an inter-agent proximity well over the required safe separation. This is done so under the assumption of corrupted measurement devices, which are assumed derived from a camera and range-finder local to the agent with their respective measurement uncertainties. The resulting avoidance trajectories are therefore determined without filtration, prediction or prior agent knowledge.

The results of the preliminary analysis are seconded by a 1000 cycle Monte-Carlo analysis in Section 7.6.5. Here, the proposed algorithm performed consistently in maintaining safe separation across each of the presented scenarios. Although infrequent, the presented analysis demonstrated the possibility of agents being unable to obtain their designated waypoints in their prescribed time. Investigation into the selection of optimal trajectories from the solution region is therefore a clear next step in the development of

the IA algorithm.

A comparison between the 2D and 3D IA approaches is given as part of the statistical analysis in which the performance of 2D IA approach is shown to be significantly improved by the algorithms ability to select trajectories in 3D. The expanded solution space is also shown to make the behaviour of the 3D IA method more predictable beyond six-agent conflicts. The performance of method however is shown to deteriorate in collisions involving closer to ten agents where trajectories cannot be found that satisfy the complete obstacle set, whilst also tolerating uncertainty in their geometric parameters. Examining the computational scalability of the 3D method with respect to the 2D method, a minor increase in computation time can be observed in response to increasing obstacle number. The consideration of each obstacle and its trajectory in 3D is shown to result in computation times that are otherwise comparable to the 2D method and scale similarly. The increased cost of computing the avoidance region for each obstacle is also reflected in the increased variability in computation time, which is also shown to be correlated to the number of obstacles.

As scalability is a well established concern in decentralised systems, the presented analyses of the 2D and 3D IA methods demonstrates that there is still work to be done in improving the performance of the approach for further use in multi-agent conflicts. As the current implementation is based on the INTLAB toolbox for Matlab [207], an immediate opportunity exists for increasing efficiency for collision avoidance through the development a c++ toolbox to support parallelisation. Similarly, the current agent assumptions were shown to be sufficient for a preliminary demonstration of the presented method. Continued development in this area, building on the works of Chapters 3 and 4, may be used to demonstrate the effectiveness of the algorithm in the context of more realistic MAV descriptions.

Chapter 8

Conclusions

In this thesis, several of the challenges related to modelling and control of multiple unmanned aerial systems (UAS) are presented. In this thesis there were two principle aims. The first was to enhance and formalise the approach to assembling UAS dynamic behaviours through the use of symbolic parameters. The challenges here stem from the variety of systems, number of propulsive groups and the techniques used to achieved low-level control.

The second aim is orientated towards higher-level autonomous behaviour; focusing on the challenges of implementing non-cooperative collision avoidance on real-world coordinated systems. Collision scenarios are often ambiguous and complex when there are multiple kinematic obstacles, limited samples available, and trajectory measurements are distorted. In addition to this, phenomena such as dead(live) lock present additional challenges that any autonomous system must address.

In Chapter 3, a symbolic framework for the derivation of non-linear dynamical behaviours is presented. It is shown how using this technique it is possible to assemble generalise UAS descriptions for the purpose of control and stability analysis. It is shown how, without parameterisation, symbolic representations of key control parameters such as the associated state space model, stability conditions and more can be defined without the assumption of a known operating point. To demonstrate the application of symbolic model definitions for linear control design, a linearisation condition is selected and applied in the context of a Quadcopter system in a straight and level (S&L) hover scenario (see Section 3.4.3). Under this assumption it is then shown how symbolic applications of linear quadratic regulation (LQR) and model predictive control (MPC) techniques may be applied to achieve control over the aircraft's attitude rates.

Utilising the symbolic description of the Quadcopter's plant, it is also speculated

how symbolic prediction matrices may be used to define more interesting control parameters. The possibility of time-variant/adaptive models is discussed for active adjustment of model assumptions and linearisation conditions across the prediction horizon. Following the demonstration of the proposed framework using the "aligned" and "Xflyer" Quadcopter configurations, the concept is applied to several other known configurations representing the state-of-the-art. This includes the derivation of the non-linear dynamics of two other thrust vectoring MAVs known as the Tricopter and the over-actuated Deltacopter or "Delta" configuration.

The dynamic interactions of vectoring components of these two configurations are used to provide context for the novel UAS referred to as the "Polycopter" in Chapter 4. This unique UAS topology presents a number of interesting motion characteristics due to the way the thrust from it's three radial nacelles may be vectored independently. Unlike previous systems, the Polycopteris morphic in that it actively servos the relative pitch and roll of it's three nacelle assemblies to augment it's centre of gravity. This presents a number of opportunities for inherent control over the vehicle's stability properties, disturbance rejection and authority in complex manoeuvres.

In these works, two derivations of the dynamics of the Polycopterare presented for the first time. The first demonstrates a reduced model under the assumptions of "small nacelle angles" where the inertia of the airframe may be assumed constant and time invariant. This is synonymous to systems where only the rotor-disk is deflected. The second derivation examines the complex interactions of the Polycopterby considering the nacelle assemblies as serial-link kinematic chains. Here, a recursive Newton-Euler method is introduced to define the nacelles as a sequence of accelerating rigid-bodies. The resulting interaction of the nacelles with the body is presented in relation to the thrust produced from their respective rotors to allow a more accurate description of the thrust vectoring properties of the Polycopterto be formed. These preliminary models are presented and discussed in light of the interesting opportunities highlighted for enhanced disturbance rejection, aerodynamic stability refinement and adaptive control.

In Chapter 6, focus is shifted to examine the challenges surrounding sense detect and avoidance (SDA) collision avoidance in coordinated unmanned aerial vehicles (UAVs) operating at a fixed altitude. A technical review of several popular approaches to non-cooperative collision avoidance is presented. The results from this analysis highlight the need for reliable yet scalable techniques in the presence of trajectory uncertainty and multi-agent conflicts. A comparative analysis of the established velocity obstacle (VO), reciprocal velocity obstacle (RVO), hybrid-reciprocal velocity obstacle (HRVO) and op-

timal reciprocal collision avoidance (ORCA) techniques is presented. The performance of the methods are compared with respect to increasing agent populations, collision rate, computational time and tolerance to obstacle trajectory uncertainty. Of the VO derivatives, the analysis showed that both the HRVO and ORCA algorithms are most effective in the presented conditions; the ORCA algorithm benefiting from it's linear representation of the constraint set and the HRVO algorithm demonstrating an increased tolerance to sensor corruption.

The comparative analysis of the state-of-the-art is used to provide context for the proposed IA algorithm. The formulation of the planar conflict scenario using interval geometry is presented. Here it is demonstrated how a planar region containing the optimal avoidance trajectories may be propagated from the collective obstacle and agent sensor uncertainties. Using some of the unique properties of intervals, it is shown how avoidance of multiple obstacles can be achieved by collating sets of optimal regions and, using their intersection, defining a region of mutual avoidance. The intersection is computed sequentially based on obstacle proximity initially, however it is postulated how strategy might be applied to encode different resolution behaviour.

An analysis of the performance of the proposed interval avoidance (IA) method is then presented in several typical planar encounters with reference to current methods. It is shown by way of a one thousand cycle Monte-Carlo analysis that, under the assumptions of corrupt sensor measurements, the IA algorithm is able to maintain safe separation for collisions involving multiple agents. The scalability of many of the selected geometric approaches is examined alongside the IA method under the presented sensor conditions. Here it is shown that the IA algorithm is more effective in avoiding collision in conflicts involving up to six agents. Hereafter, the performance of the current IA formulation is shown to deteriorate as the scenarios become more complex. Examination of the computational statistics of the presented algorithms demonstrated that many of the VO derivatives are more computationally scalable, despite their inability to tolerate trajectory distortion.

An interesting feature of the IA method is identified in the analysis of the algorithm's performance. It is shown that by exploiting the representation of the trajectory uncertainty intervals, problematic conditions such as dead lock or direction of pass ambiguity may be handled intrinsically. In each case, the optimal region is shown to define a set of possible trajectories, to which default behaviour protocols may be applied.

In Chapter 7 the IA algorithm is developed further to facilitate collision avoidance in 3D space. The chapter begins by defining an 3D interval sensor model taking meas-

urements from an on-board camera and rangefinder. It is then shown from this sensor model how trajectory intervals may be propagated in 3D using knowledge of the sensor's noise characteristics without filtration or predictive techniques. The avoidance problem demonstrated in Chapter 6 is then re-framed to allow the representation of the IA problem about a set of arbitrary conflict planes. These planes are shown geometrically to represent the set of IA problems solved simultaneously through the use of geometric intervals. As the trajectories are contained within 3D regions, the solution of IA problem naturally takes the form of a 3D region describing the set optimal avoidance trajectories for any given obstacle. Finally, it is shown how these regions may be combined using principles from Chapter 6 to allow regions of mutual avoidance to be defined in scenarios involving multiple obstacles.

The performance of the 3D IA method is examined in a series of representative conflict scenarios. The results of each scenario are presented in the form of a one thousand cycle Monte-Carlo analysis where the agent separation, computational and collision statistics were observed. In each case, the 3D IA implementation is shown to successfully avoid collision with the obstacle set in the presence of both agent and obstacle trajectory uncertainty. A comparison formed between the 3D and 2D IA methods demonstrated that the 3D algorithm is able to maintain separation with a higher number of obstacles. This is shown due to the larger search space described by the 3D interval of candidate velocities, which is also shown to make the behaviour of the 3D IA method more predictable. This is most evident in conflict scenarios beyond six agents, which is demonstrated to be the limit of 2D IA method in Chapter 6. Continued analysis of the scalability of the current 3D IA implementation did however later show the performance of the method to be limited to conflicts with less than ten agents.

An examination of the mean computational time demonstrates near identical in the presented scalability analyses; with a minor additional cost of computation per neighbouring obstacle. This increase in cost is also reflected in the variability of the 3D IA algorithms computation time, which is shown to be correlated with the number of considered obstacles. The implementation of the IA methods using the Intlab toolbox for Matlab® may however may present further opportunity for enhancing their scalability.

8.1 Direction of Future Research

Establishing an effective level of autonomy in coordinated aerial systems remains a complex task. Following the works presented in this thesis, a number of opportunities have been identified as avenues for future research. Below a non-exhaustive list is presented, grouped by the focus areas of this thesis:

Symbolic Modelling

- The generation of dynamic behaviours using symbolic UAS descriptions is a concept that is initially presented in these works in Chapter 3. Further work in this area would likely see the refinement of symbolic component descriptions, as the mechanics behind many aspects of aerial morphism are investigated.

- Based on the assumptions made in Section 3.3, there exists an opportunity to further develop the fidelity of the symbolic process. Future work may examine more precise propulsive models and rotor descriptions, introducing phenomena such as blade-flapping in [273] or motor electro-dynamic behaviour in [61, 176].

- In Section 3.4.3 it is shown how symbolic descriptions of UAVs may be used to generate key control parameters. Utilising the representation of the UAV's dynamics, there exists an opportunity to develop procedures for more sophisticated techniques, such as feedback linearisation [100, 165] and plant representations for non-linear predictive control methods or identification of unique model properties such as differential flatness [40, 81].

- The concept of the Polycopteris introduced for the first time in Chapter 4. Because of this, there are numerous opportunities in examining the unique aerodynamic and stability characteristics of the system brought about the vehicles ability to augment it's CG position.

- The uniqueness of the configuration presents some interesting avenues for controller designs that are able to exploit the over-actuated nature of the Polycopter. Avenues in which conventional multicopter designs struggle represent immediate points of interest namely: fuselage pose control, disturbance rejection, aerodynamic optimisation and active stability augmentation.

- Due to it's ability to thrust vector each nacelle, examination of the Polycopterunder the effects of the "ground" and "ceiling" present some immediate opportunities for

airflow management in complex and turbulent situations. An enhanced ability to modify the centre of thrust in landing or perching [243] exercises are also postulated to be exciting potential area of research with this concept.

Interval Avoidance (IA & 3DIA)

- Due to the popularity of presented VO derivatives, representation of complex obstacle descriptions is well established. In this thesis, the presented scenarios are in "open" air; analogous to air-traffic encounters. To better represent the capabilities of micro aerial vehicles (MAV)s operating at low altitude, interval representations of more complex obstacles, such as doorways and walls would be an immediate avenue for research.

- It is shown in the presented analysis in Section 6.5 that by utilising some of the properties of intervals, use of the supremum and infimum of this region allows specific behaviour to be applied to a given dimension. There is an opportunity to investigate further strategies for extracting escape velocities from the optimal region. This may allow protocols for more complex, cluttered, scenarios.

- In Section 6.6.4 the performance of the IA algorithm relative to the VO derivatives is shown. It is clear from the temporal statistics of the IA algorithm that optimisation is needed. The current implementation of the IA and 3D IA methods is based on the Intlab toolbox for Matlab® [207], which is known to be inefficient. This toolbox is originally intended for the estimation of rounding errors through the use of intervals, and is optimised for these operations. Development of a parallel toolbox implementing the IA method would support future work on interval based avoidance.

- The performance of the 2D IA algorithm is presented in context of the several of the state of the art geometric approaches to collision avoidance and compared to the 3D IA approach in Chapter 7. To enhance the evaluation of the 3D IA method, further investigation should be done into recent developments in 3D geometric avoidance such as the 3D-RVO tools presented in [230]. Such a comparison would provide a greater context for the analysis of the 3D IA approach, it's performance and its scalability in multi-agent conflicts.

- In Section 2.2.3 the concept of abstracting agent behaviours to "dynamic descriptors" is introduced for heterogeneous swarm simulation in OpenMAS [66]. This is initially used to facilitate the assumption of single integrator dynamics in Chapters 6 and 7

as the IA algorithm is proposed. While this assumption is well established in the literature, introduction of more complex system dynamics, similar to those Chapters 3 and 4, would better represent many of the challenges in coordinated UAV systems. Validation of the IA algorithm in higher fidelity scenarios would therefore better assess the applicably of the approach on board real-world systems.

Bibliography

[1] E. Aarts and J. Korst. *Simulated annealing and Boltzmann machines.* New York, NY; John Wiley and Sons Inc., 1988.

[2] Z.S. Abo-Hammour, O. Alsmadi, S.I. Bataineh, M.A. Al-Omari, and N. Affach. Continuous genetic algorithms for collision-free cartesian path planning of robot manipulators regular paper. *International Journal of Advanced Robotic Systems*, 8(6):14–36, 2011.

[3] V.I. Abramenko, S.I. Gopasiuk, M.B. Ogir', and V.B. Iurchishin. A robotic aircraft that follows terrain using a neuromorphic eye. In *Proceedings of the International Conference of Intelligent Robotic Systems*, volume 8, pages 50–55, Lausanne, Switzerland, 1992. IEEE.

[4] DARPA Public Affairs. 1st success seen in system designed to help aircraft automatically avoid mid-air collisions. In *Defense Advanced Projects Agency (DARPA) News Release*, Arlington, Virginia, 2016.

[5] B.M. Albaker and N.A. Rahim. A survey of collision avoidance approaches for unmanned aerial vehicles. In *In Proceedings of the International Conference for Technical Postgraduates*, pages 1–7. IEEE, 2010.

[6] P. Albertos and A. Sala. *Multivariable control systems: an engineering approach.* Springer, Valencia, Spain, 2004.

[7] G. Alefeld and G. Mayer. Interval analysis: theory and applications. *Journal of Computational and Applied Mathematics*, 121(1):421–464, 2000.

[8] Q. Ali, N. Gageik, and S. Montenegro. A review on distributed control of cooperating mini UAV's. *International Journal of Artificial Intelligence and Applications (IJAIA)*, 5(4):1–13, 2014.

[9] D. Allerton. *Principles of flight simulation.* Number 1 in Aerospace Series. John Wiley & Sons Ltd., Sheffield, 2009.

[10] T. Alpcan and I. Shames. An information-based learning approach to dual control. *Transactions on Neural Networks and Learning Systems*, 26(11):2736–2748, 2015.

[11] H. Alturbeh. *Collision avoidance systems for UAS operating in civil airspace.* PhD thesis, Cranfield University, 2014.

[12] M.H. Amoozgar and A. Chamseddine. Fault-tolerant fuzzy gain-scheduled PID for a quadrotor helicopter testbed in the presence of actuator faults. In *Proceedings of the IFAC Conference on Advances in PID Control*, pages 1–6, Brescia, Italy, 2012. IFAC.

[13] B.D. Anderson and J.B. Moore. *Optimal control: linear quadratic methods.* Courier Corporation, 2007.

[14] J.K. Archibald, J.C. Hill, F.R. Johnson, and W.C. Stirling. Satisficing negotiations. *Transactions in Systems, Man and Cybernetics- Part C Applications Review*, 36(1):4–18, 2006.

[15] L.M. Argentim, W.C. Rezende, P.E. Santos, and R.A. Aguiar. PID, LQR and LQR-PID on a quadcopter platform. In *Proceedings of the International Conference on Informatics, Electronics and Vision*, pages 1–6, May 2013.

[16] B. Armstrong, O. Khatib, and J. Burdick. The explicit dynamic model and inertial parameters of the puma 560 arm. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pages 510–518. IEEE, 1986.

[17] A. Aswani, P. Bouffard, and C. Tomlin. Extensions of learning-based model predictive control for real-time application to a quadrotor helicopter. In *Proceedings of the American Control Conference*, pages 4661–4666, Montreal, Canada, 2012. IEEE.

[18] A. Aswani, H. Gonzalez, S.S. Sastry, and C. Tomlin. Provably safe and robust learning-based model predictive control. *Automatica*, 49(5):1216–1226, 2013.

[19] T. Avant, U. Lee, B. Katona, and K. Morgansen. Dynamics, hover configurations, and rotor failure restabilization of a morphing quadrotor. In *Annual American Control Conference*, pages 4855–4862, Millwaukee, USA, 2018. ACC.

[20] N. Baba and N. Kubota. Collision avoidance planning of a robot manipulator by using genetic algorithm. a consideration for the problem in which moving obstacles

and/or several robots are included in the workspace. In *Proceedings of the 1st IEEE Conference on Evolutionary Computation*, volume 1, pages 714–719, Kashihara City, Japan, 1994. IEEE.

[21] R.J. Bamberger, D.P. Watson, D.H. Scheidt, and K.L. Moore. Flight demonstrations of unmanned aerial vehicle swarming concepts. *Johns Hopkins APL Technical Digest (Applied Physics Laboratory)*, 27(1):41–55, 2006.

[22] T. Basa and G.J. Olsder. *Dynamic noncooperative game theory*. Society for Industrial and Applied Mathematics, London, 1998.

[23] A.M. Bayen, Santhanam S., Mitchell I., and Tomlin C.J. A differential game formulation of alert levels in etms data for high-altitude traffic. In *American Institute of Aeronautics and Astronautics*, pages 1–12, Austin, Texas, 2003.

[24] R. Beard, D. Kingston, T.W. McLain, and D. Nel. Decentralized cooperative aerial surveillance using fixed-wing miniature UAVs. *Proceedings of the IEEE*, 94(7):1306–1324, 2006.

[25] J. Berg, M. Lin, and D. Manocha. Reciprocal velocity obstacles for real-time multi-agent navigation. Technical report, University of North Carolina, North California, 2008.

[26] J. Berg, M. Lin, and D. Manochag. Reciprocal velocity obstacles for real-time multi-agent navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1928–1935. IEEE, 2008.

[27] J. Berg, J. Snape, S.J. Guy, and D. Manocha. Reciprocal collision avoidance with acceleration-velocity obstacles. Technical report, University of North Carolina at Chapel Hill, North Carolina, 2011.

[28] D. Bershadsky, S. Haviland, and E.N. Johnson. The semi-coaxial multirotor. In *Proceedings of the AHS International 74th Annual Forum & Technology Display*, volume May, pages 1–9, Phoenix, Arizona, 2018.

[29] A. Bicchi and L. Pallottino. On optimal cooperative conflict resolution for air traffic management systems. *Transactions on Intelligent Transport Systems*, 1(4):221–231, 2000.

[30] K. Bilimoria, B. Sridhar, and G. Chatterji. Effects of conflict resolution manoeuvres and traffic density on free flight. In *Guidance, Navigation, and Control Conference*, volume 7, pages 1–9, 1996.

[31] R. Bosch and K. Cohen. Tilting-rotor quadcopter for aggressive flight manoeuvres using differential flatness based flight controller. In *Proceedings of the ASME Dynamic Systems and Control Conference*, pages 1–10, Virginia, USA, 2017. ASME.

[32] S. Bouabdallah, A. Noth, and R. Siegwart. PID vs LQ control techniques applied to an indoor micro quadrotor. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 2451–2456. IEEE, September 2004.

[33] S. Bouabdallah and R. Siegwart. Backstepping and sliding-mode techniques applied to an indoor micro quadrotor. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2247–2252. IEEE, April 2005.

[34] S. Bouabdallah and R. Siegwart. Full control of a quadrotor. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 153–158. IEEE, Oct 2007.

[35] P. Bouffard. On-board model predictive control of a quadrotor helicopter: design, implementation, and experiments. Technical report, University of California, Berkeley, 2012.

[36] D. Brescianini and R. D'Andrea. An omni-directional multirotor vehicle. *Mechatronics*, 55(July):76–93, 2018.

[37] G. Cai, J. Dias, and L. Seneviratne. A survey of small-scale unmanned aerial vehicles: recent advances and future development trends. *Unmanned Systems*, 2(2):175–199, 2014.

[38] C. Carbone, U. Ciniglio, F. Corraro, and S. Luongo. A novel 3D geometric algorithm for aircraft autonomous collision avoidance. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 1580–1585, San Diego, 2006. IEEE.

[39] C.L. Castillo. *Fault-tolerant adaptive model predictive control using joint Kalman filter for small-scale helicopter*. Doctorial thesis, University of South Florida, 2008.

[40] A. Chamseddine, T. Li, Y. Zhang, C.A. Rabbath, and D. Theilliol. Flatness-based trajectory planning for a quadrotor unmanned aerial vehicle test-bed considering

actuator and system constraints. In *The American Control Conference (ACC)*, pages 920–925. IEEE, 2012.

[41] D.E. Chang, S.C. Shadden, J.E. Marsden, and R. Olfati-Saber. Collision avoidance for multiple agent systems. In *Proceedings of the 42nd IEEE Conference on Decision Control*, pages 539–543, Hawaii USA, 2003. IEEE.

[42] H. Chao, Y. Cao, and Y. Chen. Autopilots for small unmanned aerial vehicles: a survey. *International Journal of Control, Automation and Systems*, 8:36–44, 2010.

[43] Z. Chao, S.L. Zhou, L. Ming, and W.G. Zhang. UAV formation flight based on nonlinear model predictive control. *Mathematical Problems in Engineering*, 2012(261367):15, 2012.

[44] O. Chatain. Cooperative and non-cooperative game theory. Technical report, MEC, Paris, 2014.

[45] H. Chen, V.P. Jilkov, and X.R. Li. On threshold optimization for aircraft conflict detection. In *Proceedings of the 18th International Conference on Information Fusion (Fusion 2018)*, pages 1198–1204. IEEE, 2015.

[46] Juntao Chen and Quanyan Zhu. Resilient and decentralized control of multi-level cooperative robotic networks to maintain connectivity under adversarial attacks. *ArXiv*, abs/1505.07158, 2015.

[47] G. Chmaj and H. Selvaraj. Distributed processing applications for UAV/drones: a survey. In *Progress in Systems Engineering*, pages 449–454. Springer, 2015.

[48] M. Choi, A. Rubenecia, T. Shon, and H. Choi. Velocity obstacle based 3D collision avoidance scheme for low-cost micro UAVs. *Sustainability (Switzerland)*, 9(7):1–23, 2017.

[49] S.J. Chung, A.A. Paranjape, P. Dames, S. Shen, and V. Kumar. A survey on aerial swarm robotics. *Transactions on Robotics*, 34(4):837–855, 2018.

[50] P.J. Conroy. The development of an aerial robotics laboratory highlighting the first experimental validation of optimal reciprocal collision avoidance. *ProQuest Dissertations and Theses*, 1545302(August):101, 2013.

[51] M.V. Cook. *Flight dynamics principles*. Elsevier, Cranfield, 2 edition, 2007.

[52] A.J. Cope, C. Sabo, K. Gurney, E. Vasilaki, and J.A.R. Marshall. A model for an angular velocity-tuned motion detector accounting for deviations in the corridor-centering response of the bee. *PLoS Computational Biology*, 12(5):1–22, 2016.

[53] Council of the European Union. Fr 3019: automatic dependent surveillance-broadcast (ads-b) out equipment and use - 91.225, 2016.

[54] Z.A. Daniels, L.A. Wright, J.M. Holt, and S. Biaz. Collision avoidance of multiple UAS using a collision cone-based cost function. Technical report, Auburn University, Auburn, 2007.

[55] P. Das, R. Ghoshal, D. Kumar Kole, and R. Ghosh. Measurement of displacement and velocity of a moving object from real time video. *International Journal of Computer Applications*, 49(13):12–16, 2012.

[56] L. Davies, R.C. Bolam, Y. Vagapov, and A. Anuchin. Review of unmanned aircraft system technologies to enable beyond visual line of sight (BVLOS) operations. In *Proceedings of the 10th International Conference on Electrical Power Drive Systems*, pages 1–6. IEEE, 2018.

[57] J. Dawes, R. Dorf, C. Tam, and L. Ng. Design of deadbeat robust systems. In *Proceedings of the 3rd Conference on Control Applications*, pages 3–4, California, 1994. IEEE.

[58] M.M. De Almeida and G.V. Raffo. Nonlinear control of a tilt-rotor UAV for load transportation. *International Federation of Automatic Control*, 48(19):232–237, 2015.

[59] D. De Martini, G.V. Gramazio, A. Bertini, C. Rottenbacher, and T. Facchinetti. Design and modelling of a quadcopter with double axis tilting rotors. *Unmanned Systems*, 05(03):169–180, 2017.

[60] D. Delahaye, N. Durand, J. Alliot, and M. Schoenauer. Genetic algorithms for air traffic control systems. *International Conference on the Optimal Real Time Management of Large Scale Transport Systems (INFAUTOM)*, September:1–13, 1996.

[61] N. A. Demerdash and T. W. Nehl. Dynamic modeling of brushless DC motors for aerospace actuation. *Transactions on Aerospace and Electronic Systems*, 16(6):811–821, Nov 1980.

[62] L. Derafa, A. Ouldali, T. Madani, and A. Benallegue. Four rotors helicopter yaw and altitude stabilization. In *Proceedings of the World Congress on Engineering*, volume 1, pages 4–8, Alger, 2007.

[63] J.A. Douthwaite, A. De Freitas, and L.S. Mihaylova. An interval approach to multiple unmanned aerial vehicle collision avoidance. In *Proceedings of the 11th Symposium Sensor Data Fusion: Trends, Solutions, and Applications (SDF)*. IEEE, September 2017.

[64] J.A. Douthwaite, L.S. Mihaylova, and S.M. Veres. Enhancing autonomy in VTOL aircraft based on symbolic computation algorithms. In L. Alboul, D. Damian, and J.M. Aitken, editors, *Towards Autonomous Robotic Systems*, pages 99–110, Cham, 2016. Springer International Publishing.

[65] J.A. Douthwaite, S. Zhao, and L.S. Mihaylova. A comparative study of velocity obstacle approaches for multi-agent systems. In *Proceedings of the 12th UKACC International Conference on Control*. IEEE, September 2018.

[66] J.A. Douthwaite, S. Zhao, and L.S Mihaylova. Matlab® open source multi-agent simulator (OpenMAS). `https://github.com/douthwja01/OpenMAS`, 2018.

[67] J.A. Douthwaite, S. Zhao, and L.S. Mihaylova. Velocity obstacle approaches for multi-agent collision avoidance. *Unmanned Systems*, 7(1):55–64, 2019.

[68] W.B. Dunbar and R.M. Murray. Model predictive control of coordinated multi-vehicle formations. In *Proceedings of the 41st Conference on Decision and Control*, pages 4631–4636. IEEE, 2002.

[69] W.B. Dunbar and R.M. Murray. Distributed receding horizon control for multi-vehicle formation stabilization. *Automatica*, 42(4):549–558, 2006.

[70] Z.T. Dydek, A.M. Annaswamy, and E. Lavretsky. Adaptive control of quadrotor UAVs: a design trade study with flight evaluations zachary. *Transactions on Control Systems Technology*, 21(4):1400–1406, 2013.

[71] E. Dyer. *Modelling and control of an omni-directional UAV*. Master's thesis, McMaster university, 2018.

[72] J. Engel, J. Sturm, and D. Cremers. Camera-based navigation of a low-cost quadrocopter. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2815–2821. IEEE, October 2012.

[73] H. Erzberger. The automated airspace concept. Technical report, NASA Ames Reseach Center, California, 2001.

[74] ESDU. Quaternion representation of aeroplane attitude. Technical report, The Royal Aeronautrical Society, 2002.

[75] ESDU. Introduction to aerodynamic derivatives, equations of motion and stability. Technical Report 86021c, The Royal Aeronautrical Society, 2003.

[76] European Aviation Safety Agency. European aviation safety agency european technical standard order. Technical report, European Aviation Safety Agency, 2012.

[77] Eurocontrol Experimental European Organisation for the Safety of Air Navigation. Extended flight rules (EFR) to apply to the resolution of encounters in autonomous airborne seperation. Technical Report September, EUROCONTROL , European Organisation For Safety of Air Navigation, Bretigny-sur-Orge, 1996.

[78] R. Falconi and C. Melchiorri. Dynamic model and control of an over-actuated quadrotor UAV. In *Proceedings of the 10th IFAC Symposium on Robot Control International Federation of Automatic Control*, volume 45, pages 192–197. IFAC, 2012.

[79] J. Farrell, M. Sharma, and M. Polycarpou. Backstepping-based flight control with adaptive function approximation. *Journal of Guidance, Control, and Dynamics*, 28(6):1089–1102, 2005.

[80] Federal Aviation Administration. National airspace system (NAS) plan. Technical report, US Department of Transportation, 1987.

[81] J. Ferrin, R. Leishman, R. Beard, and T. McLain. Differential flatness based control of a rotorcraft for aggressive maneuvers. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pages 2688–2693. IEEE, 2011.

[82] P. Fiorini and Z. Shiller. Motion planning in dynamic environments using velocity obstacles. *The International Journal of Robotics Research*, 17(7):760–772, 1998.

[83] P. Foehn and D. Scaramuzza. Onboard state dependent LQR for agile quadrotors. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 6566–6572. IEEE, 2018.

[84] K.M. Ford, N. Suri, K. Ko, P. Jisl, P. Benda, and P. Michael. A game-based approach to comparing different coordination mechanisms. In *Proceedings of the*

IEEE International Conference on Distributed Human-Machine Systems, pages 20–25. IEEE, 2008.

[85] T.I. Fossen. Mathematical models for control of aircraft and satellites. Technical Report 3, Norwegian University of Science and Technology, 2013.

[86] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *Robotics & Automation Magazine*, 4(1):23–33, 1997.

[87] T. Fraichard and H. Asama. Inevitable collision states - a step towards safer robots? In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 18, pages 2–9, Las Vagas, October 2008. IEEE.

[88] N. Franceschini. Towards automatic visual guidance of aerospace vehicles: from insects to robots. *Acta Futura*, 3(January):15–34, 2008.

[89] freepngimages.com. White dove, 2019. Accessed October 13th, 2019.

[90] M.A. Garratt. *Biologically inspired vision and control for an autonomous flying vehicle*. PhD thesis, The Australian National University, 2007.

[91] S.S. Ge and Y.J. Cui. Dynamic motion planning for mobile robots using potential field method. *Auton. Robots*, 13(3):207–222, 2002.

[92] Gertner, P. No more whirly-splat. `http://media.economist.com/sites/default/files/imagecache/original-size/20111210_STP501.jpg`, 2011. [Accessed April 25th, 2019].

[93] C. Geyer, S. Singh, and L. Chamberlain. Avoiding collisions between aircraft : state of the art and requirements for UAVs operating in civilian airspace. Technical Report January, Carnegie Mellon University Pittsburgh, Pittsburgh, 2008.

[94] A. Giese, D. Latypov, and N.M. Amato. Reciprocally-rotating velocity obstacles. Technical report, Texas A&M University, Texas, 2013.

[95] C. Goerzen, Z. Kong, and B. Mettler. *A survey of motion planning algorithms from the perspective of autonomous UAV guidance*, volume 57. Springer, 2010.

[96] A. Golas, R. Narain, S. Curtis, and M. Lin. Hybrid long-range collision avoidance for crowd simulation. *Transactions on visualization and computer graphics*, 20(7):1022–1034, 2013.

[97] J.H. Graham and J.M. Zurada. A neural network approach for safety and collision avoidance in robotic systems. *Reliability Engineering and System Safety*, 53(3):327–338, 1996.

[98] A. Grancharova, E.I. Grøtli, D. Ho, and T.A. Johansen. UAVs trajectory planning by distributed MPC under radio communication path loss constraints. *Journal of Intelligent Robotic Systems*, 79(1):115–134, 2015.

[99] B Grigorov. A generalized newton-euler algorithm for dynamic simulation of robot-manipulators with revolute joints. *Recent*, 14(2):99–105, 2013.

[100] G. Guerout, P.C. Brechard Alarcia, A. Savvaris, and A. Tsourdos. Development of a novel polyhedral multirotor: a dodecahedral aerial vehicle. In *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, volume 4, pages 989–994, 2018.

[101] S. Gupte, P. Mohandas, and J.M. Conrad. A survey of quadrotor unmanned aerial vehicles. In *Proceedings of the IEEE Southeastcon*, pages 1–6. IEEE, 2012.

[102] S.J. Guy, J. Chhugani, C. Kim, N. Satish, M. Lin, D. Manocha, and P. Dubey. Clearpath: highly parallel collision avoidance for multi-agent simulation. *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, page 12, 2009.

[103] J. Hagelbäck. *Multi-agent potential field based architectures for real-time strategy game bots*. Blekinge Institute of Technology, Blekinge, 1 edition, 2012.

[104] J.C. Hill, F.R. Johnson, J.K. Archibald, R.L. Frost, and W.C. Stirling. A cooperative multi-agent approach to free flight. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems.*, pages 1083–1090. ACM, 2005.

[105] Y. Ho, A.E. Jr. Bryson, and S. Baron. Differential games and optimal pursuit-evasion strategies. Technical Report 457, US Office of Naval Research, Massachusetts, 1964.

[106] J. Hoekstra, R. Gent, and R.C. Ruigrok. Designing for safety: the "free flight" air traffic management concept. *Reliability Engineering & System Safety*, 75:215–232, 2 2002.

[107] R. Holdsworth. *Autonomous in-flight path planning to replace pure collision avoidance for free flight aircraft using automatic dependent surveillance broadcast.* Doctoral thesis, Swinburne University, 2003.

[108] M. Horacio. *Nonlinear control systems: analysis and design.* Wiley, Alberta, Canada, 2003.

[109] S.B. Hottman, K.R. Hansen, and M. Berry. Literature review on detect, sense, and avoid technology for unmanned aircraft systems. Technical Report September, Federal Aviation Administration, U.S. Department of Transportation, Washington, DC, 2009.

[110] J. Hu and A. Lanzon. An innovative tri-rotor drone and associated distributed aerial drone swarm control. *Robotics and Autonomous Systems*, 103(1):162–174, 2018.

[111] I. Hwang, J. Kim, and C. Tomlin. Protocol-based conflict resolution for air traffic control. *Air Traffic Control Quarterly.*, 15(1):1–34, 2007.

[112] I. Hwang and C. Tomlin. Protocol-based conflict resolution for finite information horizon. In *Proceedings of the AACC American Control Conference*, pages 3–8, Anchorage, 2002.

[113] ICAO. *Annex 2: rules of the air.* Number July in 1. ICAO Publications, 10th edition, 2005.

[114] R.P. Isaacs. Games of pursuit. Technical report, The RAND Corporation, Santa Monica, California, 1951.

[115] L. Jaulin. Robust set-membership state estimation ; application to underwater robotics. *Automatica*, 45(1):202–206, 2009.

[116] F. Javidi Niroumand, A. Fakharian, and M.S. Seyedsajadi. Fuzzy integral backstepping control approach in attitude stabilization of a quadrotor UAV. In *In proceedings of the 13th Iranian Conference on Fuzzy Systems, IFSC 2013*, pages 1–6. IEEE, 2013.

[117] Y.I. Jenie, E. Kampen, C.C. de Visser, J. Ellerbroek, and J.M. Hoekstra. Three-dimensional velocity obstacle method for uncoordinated avoidance maneuvers of unmanned aerial vehicles. *Journal of Guidance, Control and Dynamics*, 39(10):2312–2323, 2016.

[118] Z. Jiang, J. Han, Y. Wang, and Q. G. Enhanced LQR control for unmanned helicopter in hover. In *Proceedings of the 1st International Symposium on Systems and Control in Aerospace and Astronautics*, pages 1438–1443, Shenyang, China, 2006.

[119] V.P. Jilkov, J.H. Ledet, and X.R. Li. Improved conflict resolution method for unmanned aircraft sense-and-avoid. In *Proceedings of the International Conference on Information Fusion*, pages 2067–2074, Heidelberg, Germany, 2016. IEEE.

[120] V.P. Jilkov, X.R. Li, and J.H. Ledet. An efficient algorithm for aircraft conflict detection and resolution using list Viterbi algorithm. In *Proceedings of the 18th International Conference on Information Fusion*, pages 1709–1716, Washington, DC, 2015. IEEE.

[121] E. Joelianto, E.M. Sumarjono, A. Budiyono, and D.R. Penggalih. Model predictive control for autonomous unmanned helicopters. *Aircraft Engineering and Aerospace Technology*, 83(6):375–387, 2011.

[122] T.A. Johansen. Toward dependable embedded model predictive control. *IEEE Systems*, 11(2):1208–1219, 2017.

[123] S. John and K.L. John. Case study : efficient algorithms for air traffic management decision support tools. In *Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pages 1–4, Bonn, 2015. IEEE.

[124] F.R. Johnson, J.C. Hill, J.K. Archibald, R.L. Frost, and W.C. Stirling. A satisficing approach to free flight. *Proceedings of the IEEE Conference on Networking, Sensing and Control*, 5:123–128, 2005.

[125] G. Jonker, F. Dignum, and J. Meyer. Achieving cooperation among selfish agents in the air traffic management domain using signed money. In *Proceedings of the 6th International Conference on Autonomous Agents and Multi-Agent Systems*, volume 5, pages 1–3, Hawaii USA, 2007. IFAAMAS.

[126] JPDO Aircraft Working Group. Nextgen avionics roadmap version 2.0. Technical report, Joint Planning and Development Office, Washington DC, 2011.

[127] B. Kada and Y. Ghazzawi. Robust PID controller design for an UAV flight control system. In *Proceedings of the World Congress on Engineering and Computer Science*, volume 2, pages 1–6, San Francisco, 2011.

[128] S. Kahne and I. Frolow. Air traffic management: evolution with technology. Technical report, MITRE Corporation's Center for Advanced Aviation System Development in McLean, 1996.

[129] M. Kamel, K. Alexis, M. Achtelik, and R. Siegwart. Fast nonlinear model predictive control for multicopter attitude tracking on SO(3). In *Proceedings of the IEEE Conference on Control and Applications*, pages 1160–1166, Sydney, 2015. IEEE.

[130] Wallace E. Kelly. Conflict detection and alerting for separation assurance systems. In *Proceedings of the Conference AIAA/IEEE Digital Avionics Systems*, volume 2, pages 1–8. IEEE, 1999.

[131] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Autonomous robot vehicles*, pages 396–404. Springer, 1986.

[132] K. Khoud, S. Bouallegue, and A. Mounir. Design and co-simulation of a fuzzy gain-scheduled PID controller based on particle swarm optimization algorithms for a quad tilt wing unmanned aerial vehicle. *Transactions of the Institute of Measurement and Control*, 40(14):3933–3952, 2018.

[133] W. H. Kirchner and M. V. Srinivasan. Freely flying honeybees use image motion to estimate object distance. *Naturwissenschaften*, 282(1):281 – 282, 1989.

[134] M.J. Kochenderfer and J.P. Chryssanthacopoulos. Robust airborne collision avoidance through dynamic programming. Technical report, Lincoln Laboratory Massachusetts Institute of Technology, Massachusetts, 2011.

[135] P.V. Kokotovic. The joy of feedback: nonlinear and adaptive. In *Control Systems*, volume 12, pages 7–17, California, 1992. IEEE.

[136] J. Košecká, C. Tomlin, G. Pappas, and S. Sastry. Generation of conflict resolution manoeuvres for air traffic management. *Proceedings of the International Conference on Intelligent Robot Systems*, 3:1598–1603, 1997.

[137] J.K. Kuchar and L.C. Yang. Prototype conflict alerting system for free flight. *Journal of Guidance Control and Dynamics*, 20(4):768–773, 1997.

[138] J.K. Kuchar and L.C. Yang. A review of conflict detection and resolution modelling methods. *Transactions on Intelligent Transportation Systems*, 1(4):179–189, 2000.

[139] K. Kunz, S. M. Huck, T. H. Summers, and J. Lygeros. Fast model predictive control of miniature helicopters. In *Proceedings of the IEEE European Control Conference*, pages 1377 – 1382. IEEE, 2013.

[140] A.R. Lachner. Collision avoidance as a differential game: real-time approximation of optimal strategies using higher derivatives of the value function. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, volume 3, pages 2308–2313. IEEE, 1997.

[141] A.R. Lachner, D.R. Maroney, and A.D. Zeitlin. Unmanned aircraft collision avoidance–technology assessment and evaluation methods. In *Proceedings of the 7th Air Traffic Management Research & Development Seminar*, pages 1–10, Virginia, 2007. The MITRE Corporation.

[142] A.R. Lachner, A.D. Zeitlin, and D.R. Maroney. Airspace integration alternatives for unmanned aircraft. Technical Report February, The MITRE Corporation, 2010.

[143] S. Ladha, D.K. Kumar, P. Bhalla, A. Jain, and R.K. Mittal. Use of LIDAR for obstacle avoidance by an autonomous aerial vehicle. In *Paper presentation at IARC-2012*, pages 1–6, Dubai, 2012. Birla Institute of Technology and Science Pilani.

[144] C.F. Lee and D.H. Wolpert. Product distribution theory for control of multi-agent systems. In *Proceedings of the Third International Joint Conference on Autonomous Multi-Agent Systems*, pages 522–529, Washington, DC, USA, 2004. IEEE Computer Society.

[145] T. Lee and Y. Kim. Nonlinear adaptive flight control using backstepping and neural networks controller. *Journal of Guidance Control and Dynamics*, 24(4):675–682, 2001.

[146] X. Lei, S.S. Ge, and J. Fang. Adaptive neural network control of small unmanned aerial rotorcraft. *J. Intelligent Robotic Systems*, 75(2):331–341, 2014.

[147] T. Lester, S. Cook, and K. Noth. USAF airborne sense and avoid (ABSAA) airworthiness and operational approval approach. Technical report, MITRE Corporation, Bedford, Massachusetts, 2014.

[148] A. Levy, C. Keitel, S. Engel, and J. McLurkin. The extended velocity obstacle and applying ORCA in the real world. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume June, pages 16–22, Washington, Seattle, 2015. IEEE.

[149] C.Y. Li, W.X. Jing, and C.S. Gao. Adaptive backstepping-based flight control system using integral filters. *Aerospace Science and Technology*, 13(2-3):105–113, 2009.

[150] R. Licitra, A.J. Neale, E.A. Doucette, and J.W. Curtis. Adversarial aircraft diversion and interception using missile herding techniques. In *SPIE Defense and Commercial Sensing*, volume May 2019, page 80, Baltimore, Maryland, United States, 2019. SPIE Digital Library.

[151] C. Liu, W. Chen, and J. Andrews. Explicit non-linear model predictive control for autonomous helicopters. In *Loughborough University's Institutional Repository*, pages 1–21, Loughborough, 2012. Institution of Mechanical Engineers (IMechE).

[152] R. Lopes, P. Santana, G. Borges, and J. Ishihara. Model predictive control applied to tracking and attitude stabilization of a VTOL quadrotor aircraft. In *Proceedings of the 21st International congress of Mechanical Engineering*, volume 5, pages 176–185, 2011.

[153] R. Lozano. *Unmanned aerial vehicles: embedded control*. John Wiley & Sons, 2013.

[154] S. Lupashin and R. D'Andrea. Adaptive open-loop aerobatic maneuvers for quadrocopters. In *Proceedings of the 18th World Congress of the International Federation of Automatic Control*, volume 44, pages 2600–2606. IFAC, 2011.

[155] J.M. Maciejowski. *Predictive control with constraints*. Pearson Education, Harlow, Essex, 1 edition, 2002.

[156] J.A. Marin, R. Radtke, D. Innis, D.R. Barr, and A.C. Schultz. Using a genetic algorithm to develop rules to guide unmanned aerial vehicles. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 1055–1060. IEEE, 1999.

[157] M Mark, S. Lupashin, and R.D Andrea. Quadrocopter ball juggling. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 5113–5120, San Francisco, CA, USA, 2011. IEEE.

[158] D. Martini, G.V. Gramazio, A. Bertini, C. Rottenbacher, and T. Facchinetti. Design and modelling of a quadcopter with double axis tilting rotors. *Unmanned Systems*, 5(3):169–180, 2017.

[159] S.H. Mathisen, T.I. Fossen, and T.A. Johansen. Non-linear model predictive control for guidance of a fixed-wing UAV in precision deep stall landing. *American Institute of Aeronautics and Astronautics*, 16(1):1–16, 2016.

[160] J. McCalmont, J. Utt, and M. Deschenes. Detect and avoid technology demonstration. In *AIAA's 1st Technical Conference and Workshop on Unmanned Aerospace Vehicles*, Portsmouth, Virginia, 2002. AIAA.

[161] Eurocontrol Media and News. Closing the gaps: a report on what still needs to be done for the single european sky, 2nd August 2016.

[162] P.K. Menon, G.D. Sweriduk, and B. Sridhar. Optimal strategies for free-flight air traffic conflict resolution. *Journal of Guidance, Control, and Dynamics*, 22(2):202–211, 1999.

[163] A.W. Merz. Maximum-miss aircraft collision avoidance. *Dynamics and Control*, 1(1):25–34, 1991.

[164] I. Mitchell, A. Bayen, and C. Tomlin. A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games. *Transactions on Automatic Control*, 50(7):947–957, 2005.

[165] M. Kara Mohamed and A. Lanzon. Effect of unmodelled actuator dynamics on feedback linearised systems and a two stage feedback linearisation method. In *Proceedings of the 52nd IEEE Conference on Decision and Control*, pages 841–846. IEEE, 2013.

[166] R.E. Moore, R.B. Kearfott, and M.J. Cloud. *Introduction to interval analysis*. Society for Industrial and Applied Mathematics, Philadelphia, 1 edition, 2009.

[167] Y. Mulgaonkar, G. Cross, and V. Kumar. Design of small, safe and robust quadrotor swarms. *Proceedings of the IEEE International Conference on Robotics and Automation*, June(1):2208–2215, 2015.

[168] R.M. Murray. Recent research in cooperative control of multi-vehicle systems. *ASME Journal of Dynamic Systems, Measurement, and Control*, 1(1):1–27, 2007.

[169] R.R. Negenborn and J.M. Maestre. Distributed model predictive control: an overview and roadmap of future research opportunities. *Control Systems*, 34(August):87–97, 2014.

[170] M. Neunert, C. De Crousaz, F. Furrer, M. Kamel, F. Farshidian, R. Siegwart, and J. Buchli. Fast nonlinear model predictive control for unified trajectory optimization and tracking. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume June, pages 1398–1404. IEEE, 2016.

[171] D. Nicolas and A. Jean-Marc. Collision avoidance using neural networks learned by genetic algorithms. In *IEA/AIE*, volume 9, 2014.

[172] M. Nieuwenhuisen, M. Schadler, and S. Behnke. Predictive potential field-based collision avoidance for multicopters. *International Archives of the Photogrammetry Remote Sensing and spatial Information Sciences*, 1:1–6, 2013.

[173] I.K. Nikolos, K.P. Valavanis, N.C. Tsourveloudis, and A.N. Kostaras. Evolutionary algorithm based off-line path planner for UAV navigation. *Transactions on systems, man, and cybernetics*, 43(6):143–150, 2003.

[174] M.S. Nolan. *Fundamentals of air traffic control*. Wadsworth Publishing Company, 1994.

[175] K. Nonami, F. Kendoul, S. Suzuki, W. Wang, and D. Nakazawa. *Autonomous flying robots: unmanned aerial vehicles and micro aerial vehicles*. Springer, Tokyo, 1st edition, 2010.

[176] Y Obulesh, S. Babu Ch, and A. Rao. Mathematical modelling of a BLDC motor with closed loop speed control using PID controller under various loading conditions. *ARPN Journal of Engineering and Applied Sciences*, 7:1321–1328, 10 2012.

[177] M. Odelga, P. Stegagno, and H. Bulthoff. A fully actuated quadrotor UAV with a propeller tilting mechanism: modelling and control. In *Proceedings of the International Conference on Advanced Intelligent Mechatronics*, volume September, pages 306–311, Banff, Alberta, 2016. IEEE.

[178] Office of the Secretary of Defense. Airspace integration plan for unmanned aviation. Technical Report November, Department of Defense, Washington, DC, USA, 2004.

[179] Kestral Onboard. Kestrel autopilot system. Technical report, Procerus Technologies, 2010.

[180] A. Oosedo, S. Abiko, S. Narasaki, A. Kuno, A. Konno, and M. Uchiyama. Flight control systems of a quad tilt rotor unmanned aerial vehicle for a large attitude

change. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 2326–2331. IEEE, 2015.

[181] J.G. Ortega and E.F. Camacho. Mobile robot navigation in a partially structured static environment, using neural predictive control. *Control Engineering Practice*, 4(12):1669–1679, 1996.

[182] H. Otsuka, D. Sasaki, and K. Nagatani. Reduction of the head-up pitching moment of small quad-rotor unmanned aerial vehicles in uniform flow. *International Journal of Micro Air Vehicles*, 10(1):85–105, 2018.

[183] R.A. Paielli and H. Erzberger. Conflict probability estimation for free flight. In *Proceedings of the 35th Aerospace Sciences Meeting and Exhibit*, volume October. National Aeronautics and Space Administration, 1997.

[184] R.A. Paielli, H. Erzberger, D. Chiu, and K.R. Heere. Tactical conflict alerting aid for air traffic controllers. *AIAA Journal of Guidance, Control and Dynamics*, 32(1):1–19, 2009.

[185] G. Pappas, C. Tomlin, J. Lygeros, D. Godbole, and S. Sastry. Next generation architecture for air traffic management systems. In *Proceedings of the IEEE Conference on Decision and Control*, volume 3, pages 2405–2410. IEEE, 1997.

[186] A. A. Paranjape, S. Chung, K. Kim, and D. H. Shim. Robotic herding of a flock of birds using an unmanned aerial vehicle. *Transactions on Robotics*, 34(4):901–915, Aug 2018.

[187] J.W. Park, H.D. Oh, and M.J. Tahk. UAV collision avoidance based on geometric approach. In *Proceedings of the SICE Annual Conference*, pages 2122–2126. SICE, 2008.

[188] T. Paul, T.R. Krogstad, J.T. Gravdahl, and J. Tommy. Modelling of UAV formation flight using 3D potential field. *Simulation Modelling Practice and Theory*, 16(9):1453–1462, 2008.

[189] M. Pechoucek and D. Sislák. Agent-based approach to free-flight planning, control, and simulation. *IEEE Intelligent Systems*, 24:14–17, 01 2009.

[190] L.P. Perera, J.P. Carvalho, and C. Guedes. Autonomous guidance and navigation based on the COLREGS rules and regulations of collision avoidance . In *Advanced*

Ship Design for Pollution Prevention, pages 205–216, Split, Croatia, 2010. Taylor & Francis Group.

[191] A. Pierson and M. Schwager. Bio-inspired non-cooperative multi-robot herding. *Proceedings of the IEEE International Conference on Robotics and Automation*, June(1):1843–1849, 2015.

[192] The Joint Planning and Development Office (JPDO). Unmanned aircraft systems (UAS) comprehensive plan. Technical report, Federal Aviation Administration (FAA), Arlington, Virginia, 2013.

[193] J.T. Platts. Autonomy in umanned air vehicles. *The Aeronautical Journal*, page 97, 2006.

[194] P. Poksawat, L. Wang, and A. Mohamed. Gain scheduled attitude control of fixed-wing UAV with automatic controller tuning. *Transactions on Control Systems Technology*, 1(May):1063–6536, 2017.

[195] G. Portelli, F. Ruffier, and N. Franceschini. Evidence for ventral optic flow regulation in honeybees. In *Cinquième Conférence Plénière Française de Neurosciences Computationnelles*, pages 1–4. HAL Archives, 2011.

[196] G. Portelli, J. Serres, F. Ruffier, and N. Franceschini. Modelling honeybee visual guidance in a 3-d environment. *Journal of Physiology-Paris*, 104(1):27 – 39, 2010. Computational Neuroscience, from Multiple Levels to Multi-level.

[197] S. Ramasamy and R. Sabatini. A unified approach to cooperative and non-cooperative sense-and-avoid. In *Proceedings of the International Conference on Unmanned Aircraft Systems*, pages 765–773. IEEE, 2015.

[198] W. Ren and E.M. Atkins. Nonlinear trajectory tracking for fixed wing UAVs via backstepping and parameter adaptation. *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, pages 1–11, 2005.

[199] C.W. Reynolds. *Flocks, herds and schools: a distributed behavioural model*, volume 21. ACM, 1987.

[200] F. Riaz, M.A. Niazi, M. Sajid, S. Amin, N.I. Ratyal, and F. Butt. An efficient collision avoidance scheme for autonomous vehicles using genetic algorithm. *Journal of Applied Environmental and Biological Sciences*, 5(8):70–76, 2015.

[201] A. Richards and J. How. Decentralized model predictive control of cooperating UAVs. *Proceedings of the Conference on Decision and Control*, 4:4286–4291, 2004.

[202] A. Richards and Jonathan P. How. Model predictive control of vehicle manoeuvres with guaranteed completion time and robust feasibility. *Information and Software Technology*, 51(4):769–784, 2009.

[203] E. Rimon and D.E. Koditschek. Exact robot navigation using artificial potential functions. *Transactions on Robotics and Automation*, 8(5):501–518, 1992.

[204] J. Rong, S. Geng, J. Valasek, and T. Ioerger. Air traffic conflict negotiation and resolution using an onboard multi-agent system. In *Proceedings of the 21st Digital Avionics Systems Conference*, volume 2, pages 782–794, Oct 2002.

[205] F. Ruffier and N. Franceschini. Octave: a bioinspired visuo-motor control system for the guidance of micro-air-vehicles. In *Proceedings of SPIE*, page 1, Canary Islands, Spain, 2003. SPIE.

[206] F. Ruffier, S. Viollet, S. Amic, and N. Franceschini. Bio-inspired optical flow circuits for the visual guidance of micro air vehicles. *Proceedings of the International Symposium on Circuits and Systems*, 3:846–849, 2003.

[207] S.M. Rump. INTLAB - INTerval LABoratory. In T. Csendes, editor, *Developments in Reliable Computing*, pages 77–104. Kluwer Academic Publishers, Dordrecht, 1999. `http://www.ti3.tuhh.de/rump/`.

[208] M. Ryll, H.H. Bulthoff, and P.R. Giordano. A novel overactuated quadrotor unmanned aerial vehicle: modeling, control, and experimental validation. *Transactions on Control Systems Technology*, 23(2):540–556, 2015.

[209] R. Sabatini, A. Gardi, and M.A. Richardson. LIDAR obstacle warning and avoidance system for unmanned aircraft. *International Journal of Mechanical, Aerospace, Industrial and Mechatronics Engineering*, 8(4):704–715, 2014.

[210] I. Sadeghzadeh, A. Mehta, A. Chamseddine, and Y. Zhang. Active fault tolerant control of a quadrotor UAV based on gain scheduled PID control. In *Proceedings of the 25th IEEE Canadian Conference on Electrical and Computer Engineering*, pages 1–4. IEEE, April 2012.

[211] Safety and Airspace Regulation Group. Unmanned aircraft system operations in UK airspace – guidance. Technical Report 5, Civil Aviation Authority, West Sussex, 2015.

[212] S. Saha, A. Natraj, and S. Waharte. A real-time monocular vision-based frontal obstacle detection and avoidance for low cost UAVs in GPS denied environment. In *Proceedings of the IEEE International Conference on Aerospace Electronics and Remote Sensing Technology*, pages 189–195. IEEE, Nov 2014.

[213] S. Salazar-Cruz, R. Lozano, and J. Escareño. Stabilization and nonlinear control for a novel trirotor mini-aircraft. *Control Engineering Practice*, 17(8):886–894, 2009.

[214] M. Santos, V. López, and F. Morata. Intelligent fuzzy controller of a quadrotor. In *Proceedings of IEEE International Conference on Intelligent Systems and Knowledge Engineering*, pages 141–146. IEEE, 2010.

[215] V.G. Santos, M.F.M. Campos, and L. Chaimowicz. On segregative behaviors using flocking and velocity obstacles. In *Distributed Autonomous Robotic Systems*, pages 121–133. Springer, 2014.

[216] M. Schwager, P. Dames, D. Rus, and V. Kumar. A multi-robot control policy for information gathering in the presence of unknown hazards. *Springer Tracts in Advanced Robotics*, 100:455–472, 2017.

[217] K. Sears. Automatic collision avoidance technology / fighter risk reduction project, 2013.

[218] P. Segui-Gasco, Y. Al-Rihani, H.S. Shin, and A. Savvaris. A novel actuation concept for a multi rotor UAV. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 74(1-2):173–191, 2014.

[219] Z. Shiller, F. Large, S. Sekhavat, and C. Laugier. Motion planning in dynamic environments: obstacles moving along arbitrary trajectories. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 4, pages 3716–3721. IEEE, 2001.

[220] D.H. Shim, H.J. Kim, and S. Sastry. Control system design for rotorcraft-based unmanned aerial vehicles using time-domain system identification. In *Proceedings of the IEEE Conference on Control Applications*, volume 1, pages 808–813. IEEE, 2000.

[221] J. Shin and H.J. Kim. Nonlinear model predictive formation flight. *Transactions on Systems, Man and Cybernetics*, 39(5):1116–1125, 2009.

[222] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.

[223] A. Sikander and R. Prasad. Soft computing approach for model order reduction of linear time invariant systems. *Circuits, Systems, and Signal Processing*, 34(11):3471–3487, 2015.

[224] H.A. Simon. A behavioural model of rational choice. *The Quarterly Journal of Economics*, 69(1):99–118, 1952.

[225] D. Sislak. *Autonomous collision avoidance in air-traffic domain*. Doctorial thesis, Czech Technical University in Prague, 2010.

[226] D. Šišlák. *Agent-based approach to air-traffic modelling, simulation and collision avoidance*. Habilitation thesis, Czech Technical University In Prague, 2013.

[227] D. Sislak, M. Rehak, M. Pechoucek, D. Pavlicek, and M. Uller. Negotiation-based approach to unmanned aerial vehicles. In *Proceedings of the IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications*, pages 279–284. IEEE, 2006.

[228] D. Sislak, P. Volf, A. Komenda, J. Samek, and M. Pechoucek. Agent-based multilayer collision avoidance to unmanned aerial vehicles. In *Proceedings of the International Conference on Integration of Knowledge Intensive Multi-Agent Systems*, pages 365–370. IEEE, 2007.

[229] D. Šišlák, P. Volf, M. Pěchouček, N. Suri, D. Nicholson, and D. Woodhouse. Optimization-based collision avoidance for cooperating airplanes. In *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, volume 2, pages 375–378. IEEE, 2009.

[230] J. Snape and D. Manocha. Navigating multiple simple-airplanes in 3D workspace. In *Proceedings of the International Conference on Robotics and Automation*, volume 1, pages 3974–3980, Anchorage, USA, May 2010. IEEE.

[231] M. Sniedovich. Dynamic programming and principles of optimality. *Journal of Mathematical Analysis and Applications*, 65(3):586–606, 1978.

[232] D.P. Spanos and R.M. Murray. Motion planning with wireless network constraints. In *Proceedings of the American Control Conference*, pages 87–92, Portland, Oregon, 2005. IEEE.

[233] M. V Srinivasan. Visual control of navigation in insects and its relevance for robotics. *Current opinion in neurobiology*, 21(4):535–543, 2011.

[234] R.S. Stansbury, M.A. Vyas, and T.A. Wil. A survey of UAS technologies for command, control, and communication. In K.P. Valavanis, P. Oh, and L A Piegl, editors, *Unmanned Aircraft Systems*, pages 61–77. Springer, 2010.

[235] R.F. Stengel. Toward intelligent flight control. *Transactions on Systems, Man and Cybernetics*, 23(6):1699–1717, 1993.

[236] Jack Stewart. Google tests drone deliveries in project wing trials. *BBC World Service Radio*, 2014.

[237] J. Stowers, M. Hayes, and A. Bainbridge-Smith. Biologically inspired UAV obstacle avoidance and control using monocular optical flow & divergence templates. In *Proceedings of the 5th International Conference on Automation, Robotics and Applications*, pages 378–383, Wellington, New Zealand, 2011. IEEE.

[238] S. Sugathan, B.V. Sowmya Shree, M.R. Warrier, and C.M. Vidhyapathi. Collision avoidance using neural networks. *IOP Conference Series- Materials Science and Engineering*, 263(5):1–9, 2017.

[239] T. N. Sullivan, M. A. Meyers, and E. Arzt. Scaling of bird wings and feathers for efficient flight. *Science Advances*, 5(1), 2019.

[240] M. Suzuki and K. Uchiyama. Three-dimensional formation flying using bifurcating potential fields. In *Proceedings of the AIAA Conference on Guidance, Navigation and Control*, volume 22, pages 1753–1759. AIAA, 2009.

[241] Y. Tanabe, S. Saito, N. Ooyama, and K. Hiraoka. Study of a downwash caused by a hovering rotor in ground effect. *34th European Rotorcraft Forum*, 1:589–599, 2008.

[242] Y. Tanabe, S. Saito, N. Ooyama, and K. Hiraoka. Investigation of the downwash induced by rotary wings in ground effect. *International Journal of Aeronautical and Space Sciences*, 10(1):20–29, 2009.

[243] J. Thomas, M. Pope, G. Loianno, E.W. Hawkes, M.A. Estrada, H. Jiang, M.R. Cutkosky, and V. Kumar. Aggressive flight for perching on inclined surfaces. *Journal of Mechanisms and Robotics*, 8(October):1–10, 2015.

[244] J.L. Thorsson and O. Steinert. *Neural networks for collision avoidance learning and genetic algorithms for active safety functions.* Master's thesis, Chalmers University of Technology, 2016.

[245] Bret W. Tobalske. Biomechanics of bird flight. *Journal of Experimental Biology*, 210(18):3135–3146, 2007.

[246] C.J. Tomlin, J. Lygeros, and S. Shankar. A game theoretic approach to controller design for hybrid systems. *Proceedings of the IEEE*, 88(7):949–970, 2000.

[247] A.A. Tunik and O.I. Nadsadna. Robust digital gain-scheduling control of the UAV altitude. In *Proceedings of the IEEE 4th International Conference Actual Problems of Unmanned Aerial Vehicles Developments*, pages 175–179, Kiev, Ukraine, 2017. IEEE.

[248] G. Vachtsevanos, T. Liang, and J. Reimann. An intelligent approach to coordinated control of multiple unmanned aerial vehicles. In *Proceedings of the American helicopter society 60th annual forum*, pages 1–9, Atlanta, 2004. AHS.

[249] G. Vachtsevanos, L. Tang, G. Drozeski, and L. Gutierrez. From mission planning to flight control of unmanned aerial vehicles: strategies and implementation tools. *Annual Reviews in Control*, 29:101–115, 2005.

[250] J. Van Den Berg, S.J. Guy, M. Lin, and D. Manocha. Reciprocal n-body collision avoidance. In *Robotics research*, volume 70, pages 3–19. Springer Tracts in Advanced Robotics, 2011.

[251] J. van den Berg, J. Snape, S. Guy, and D. Manocha. Reciprocal collision avoidance with acceleration-velocity obstacles. In *Proceedings of the International Conference on Robotics and Automation*, pages 3475–3482. IEEE, 2011.

[252] R. Vidal, O. Shakernia, H.J. Kim, D.H. Shim, and S. Sastry. Probabilistic pursuit-evasion games: theory, implementation, and experimental evaluation. *Transactions on Robotics Automation*, 18(5):100–107, 2002.

[253] E. Vries and K. Subbarao. Cooperative control of swarms of unmanned aerial vehicles. *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 18(49):115–117, 2011.

[254] J.P. Wangermann and R.F. Stengel. Principled negotiation between intelligent agents: a model for air traffic management. *Artificial Intelligence in Engineering*, 12(3):177 – 187, 1998.

[255] J.P. Wangermann and R.F. Stengel. Optimization and coordination of multiagent systems using principled negotiation. *Journal of Guidance, Control, and Dynamics*, 22(1):43–50, 1999.

[256] P. Wen and T.W. Lu. Decoupling control of a twin rotor mimo system using robust deadbeat control technique. *IET Control Theory and Applications*, 2(11):999–1007, 2008.

[257] K. Wesselowski and R. Fierro. A dual-mode model predictive controller for robot formations. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 5–10, Hawaii, 2003. IEEE.

[258] J. Wilkerson, J. Bobinchak, M. Culp, J. Clark, T. Halpin-Cha, K. Estabridis, and G. Hewer. Two-dimensional distributed velocity collision avoidance. Technical Report February, Naval Air Warfare Center Weapons Division, 2014.

[259] D. Wilkie, J. Berg, and D. Manocha. Generalized velocity obstacles. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pages 5573–5578, North Carolina, 2009. IEEE.

[260] S. Wollkind, J. Valasek, and T.R. Ioerger. Automated conflict resolution for air traffic management using cooperative multiagent negotiation. *AIAA Conference on Guidance Navigation and Control*, 4992(August):1–11, 2004.

[261] A. Wu. *Guaranteed avoidance of unpredictable, dynamically constrained obstacles using velocity obstacle sets.* Masters thesis, Massachusetts Institute of Technology, 2011.

[262] Y. Xi, D. Li, and S. Lin. Model predictive control — status and challenges. *Acta Automatica Sinica*, 39(3):222–236, 2013.

[263] J. Xiao, Z. Michalewicz, L. Zhang, and K. Trojanowski. Adaptive evolutionary planner/navigator for mobile robots. *Transactions on Evolutionary Computation*, 1(1):18–28, 1997.

[264] B. Yang and M. Liu. Attack-resilient connectivity game for UAV networks using generative adversarial learning. *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, 3:1743–1751, 2019.

[265] Y. Yang and Y. Yan. Attitude regulation for unmanned quadrotors using adaptive fuzzy gain-scheduling sliding mode control. *Aerospace Science and Technology*, 54:208–217, 2016.

[266] D. Yeo, E. Shrestha, D.A. Paley, and E. Atkins. An empirical model of rotorcraft UAV downwash model for disturbance localization and avoidance. *AIAA Atmospheric Flight Mechanics Conference, 2015*, 1(January):1–15, 2015.

[267] L.A. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *Transactions on Systems, Man and Cybernetics*, 3(1):28–44, 1973.

[268] L.A. Zadeh. Soft computing and fuzzy logic. *IEEE Software*, 11(6):48–56, 1994.

[269] U. Zengin and A. Dogan. Cooperative target pursuit by multiple UAVs in an adversarial environment. *Robotics and Autonomous Systems*, 59(12):1049 – 1059, 2011.

[270] S. Zerbst and O. Duvel. *3D game engine programming.* Number One in Game Development Series. Premier Press, Boston, 2004.

[271] A. Zhahir, A. Razali, and M. Ajir. Current development of UAV sense and avoid system. In *Proceedings of the IOP Conference Series: Materials Science and Engineering*, volume 152, Universiti Putra Malaysia, 2016. IOP Publishing.

[272] H. Zhang, Z. Cheng, G. Chen, and C. Li. Model predictive flocking control for second-order multi-agent systems with input constraints. *Transactions on Circuits and Systems*, 62(6):1599–1606, 2015.

[273] X. Zhang, X. Li, K. Wang, and Y. Lu. A survey of modelling and identification of quadrotor robot. *Abstract and Applied Analysis*, 2014:1–16, 2014.

[274] M. Zhao, T. Anzai, F. Shi, X. Chen, K. Okada, and M. Inaba. Design, modeling, and control of an aerial robot dragon: a dual-rotor-embedded multilink robot with the

ability of multi-degree-of-freedom aerial transformation. *Robotics and Automation Letters*, 3(2):1176–1183, April 2018.

[275] M. Zhao, K. Kawasaki, X. Chen, S. Noda, K. Okada, and M. Inaba. Whole-body aerial manipulation by transformable multirotor with two-dimensional multilinks. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 5175–5182. IEEE, 2017.

[276] S. Zhao, F. Lin, K. Peng, B.M. Chen, and T.H. Lee. Distributed control of angle-constrained cyclic formations using bearing-only measurements. *Systems and Control Letters*, 63(1):12–24, 2014.

[277] S. Zhao and Z. Sun. Defend the practicality of single-integrator models in multi-robot coordination control. In *Proceedings of the 13th IEEE International Conference on Control and Automation*. IEEE, August 2017.

[278] B. Zhu, L. Xie, D. Han, X. Meng, and R. Teo. A survey on recent progress in control of swarm systems. *Science China Information Sciences*, 60(7):70–201, 2017.

[279] G. Zlotkin and J. Rosenschein. Negotiation and task sharing among autonomous agents in cooperative domains. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, volume 2, pages 912–917, 1989.

[280] A.F. Şenkul and E. Altuğ. System design of a novel tilt-roll rotor quadrotor UAV. *Journal Intelligent and Robotic Systems: Theory and Applications*, 84(1-4):575–599, 2015.