MANCHESTER
1824
The University of Manchester

Novel Navigation Structures for Aerial Unmanned Traffic Management

A thesis submitted to the University of Manchester for the degree of

Doctor of Philosophy

in the Faculty of Science and Engineering, School of Engineering

2023

Samuel Knox

Department of Mechanical Aerospace and Civil Engineering

Contents

Word Count: 29247

List of figures

List of tables

Nomenclature

Variables

λ	Mean free path
r	Rule in set α
τ_r	Rule update rate
$\mathbf{a}$	Acceleration
$\mathbf{d}$	Direction to next waypoint
$\mathbf{F}$	Force
$\mathbf{v}_{\mathrm{drift}}$	Drift velocity
Θ	Velocity variance
θ_C	Corner angle
Θ_d	Velocity direction variance
Θ_G	Generalised velocity variance
θ_h	Helix angle
φ	Strip angle
a_{max}	Maximum acceleration
D_e	Environment diameter
K^r	Gain of rule r
K_{vv}	Covariance matrix of velocity

l_c	Turning radius
L_j	Junction length
L_s	Switching length
m	Gradient
N^d	Number of dimensions used in path
N^i	Number of inflows
R	Radius of orbit
R^2	Coefficient of determination
r_{min}	Minimum distance between agent center points
t^d	Deviation free time
t^t	True time
t_i	Inflow rate
U	Uniform random number
u_{ref}	Reference speed
x^N	Nearest point on path cross-section
x_{ij}	Relative position between agent i and agent j
Z	Collision rate
α	Set of all rules
τ	Mean free time
$\mathbf{D}$	Rule Direction
$\mathbf{v}_i$	Velocity of agent i
$\mathbf{v}_i^g$	Total goal velocity for agent i

$\mathbf{v}_{ij}$	Relative velocity between agent i and agent j
$\mathbf{x}_i$	Position of agent i
A	Area
a, b	Weighting variables in sigmoid function
D	List of directions
D_c	Collision Distance
d_c	Conflict Distance
d_j	Characteristic junction size
M	Rule magnitude
N	Number of agents
$r\ \theta\ \phi$	Polar coordinates
u	Speed
Y	List of waypoints
$\bar{v}$	Mean speed
ρ	Density
$L\ H\ W$	Length, Height, Width
V	Volume

Superscrips

$\mathcal{A}$	Velocity-based attraction rule
$\mathcal{E}$	External velocity rule
$\mathcal{F}$	Flow rule
$\mathcal{G}$	Geometry rule

$\mathcal{M}$	Migration Rule
$\mathcal{OS}$	Orthogonal separation rule
$\mathcal{P}$	Procession rule
$\mathcal{R}$	Repulsion rule
$\mathcal{SS}$	Simple separation rule
θ	Angle
cs	Cross-section
E	Error
e	Earth axes
g	Goal
N	Nearest point
R	Repulsion
SF	Side force
w	Waypoint
x	Position
σ	Sigmoid

Subscripts

CF	Cross flow
i	Agent i
UR	Uniform Random

Operators

$'$	Cross section reference frame

Δ	Change in
$\sigma(\cdots)$	Standard deviation
$\langle\cdots\rangle$	Mean

Abstract

A video abstract is found here: PhD Video Abstract - https://youtu.be/sAQeCa4lurc

The aim of this thesis is to advance distributed control in unmanned traffic management (UTM) systems using novel rule-based navigation structures. Current UTM systems typically use a centralised volume reservation approach to maintain separation between agents. Navigation structures for a distributed control UTM exist in the literature, such as aerial routing networks, zones, and layers; however, there are limited theoretical, analytical, or experimental data to allow a critical evaluation of alternatives. The three main contributions in this work are the design, implementation, and evaluation of aerial routing networks, novel deconfliction strategies, and the development of a congestion metric in point and field measurements.

Aerial routing networks are composed of paths and intersections. Path geometries examined in this work include lanes (1D), ribbons (2D), and corridors (3D). A novel topology is introduced in the form of a helical tube (2.5D). Intersections are modelled using combinations of converging and diverging junctions. Aerial routing networks are implemented as static objects with predefined topology or as dynamic objects where paths and intersections are emergent. A novel orthogonal separation rule is proposed to increase efficiency over classical separation rules. Navigation structures are evaluated using transit efficiency, dimensionless conflict rate, and dimensionless collision rate. To measure risk in aerial systems, methods that are improved from traffic theory and gas kinematics are compared for use as a congestion metric.

Numerical experiments were performed using the real-time simulation environment Unity. For predefined aerial routing networks, rules for proposed paths were implemented in code, with the goal of achieving rules that were suitable for purpose with minimum complexity. Converging and diverging junctions of all types of paths were tested at a range of junction angles. Emergent aerial routing networks and deconfliction strategies were tested using variable

angle cross-flow, destination opposite flow, and uniform random flow test cases. Analysis of the congestion metric was performed in a test case that varied the closing velocity and density. A procedurally generated city was also used in example full-scale implementations.

Rule-based algorithms effectively controlled agents within predefined aerial routing networks. Out-of-plane ribbon junctions had the lowest dimensionless conflict rates compared to other 2D paths. Helical tube junctions had lower dimensionless conflict rates compared to 3D paths in diverging junctions, but similar conflict rates in converging junctions. However, helical tubes exhibit reduced capacity compared to corridors. Emergent aerial routing networks reduced dimensionless collision rates by approximately 98% and 79% in the contra flow and destination opposite test flows, respectively. A reduction in transit efficiency was expected due to the increase in path length; however, transit efficiency increased due to a reduction in collision avoidance, thus allowing consistently higher speeds. Using the orthogonal separation rule, collision rates of 0 with transit efficiencies greater than 0.99 were found in the destination opposite test case. A gas kinematic method for computing congestion was found to have the strongest relationship with the collision rate. This method was then used to measure congestion in several flow test cases, including an entire city test case.

Practical application of aerial routing network concepts was evaluated using experiments with 8 multirotor drones controlled with a bespoke multi-agent framework. The experiments showed that the aerial routing networks performed consistently with the simulations. The key challenge of the implementation was the typical communication latency between agents of 0.2 s and communication dropouts. Communication latency/dropouts led to greater position uncertainty; therefore, cruise speed, minimum separation, and rule update rates were finely balanced to ensure safety.

This thesis has shown how both pre-defined and emergent navigation structures can be used to successfully improve the transit efficiency and safety of UTM systems. Contributions are made to the design of rule-based algorithms for navigation structures and novel collision avoidance methods, and the development of analytical, numerical, and experimental methods for critical evaluation of navigation structure performance. The work has shed light on some novel aspects of structure and behaviour for UTM systems, and provides clear signposts for future discovery in this rich opportunity space.

Declaration of originality

I hereby confirm that no portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright statement

Acknowledgements

First, I would like to give my sincerest thanks to **Bill Crowther** and **Peter Hollingsworth** for their continual support and guidance throughout my thesis. Without their valuable contributions, this thesis would have been completed in half the time. Also within the University of Manchester, I would like to thank **Verdon Crann** and **Peyman Amiri** for their contributions in bringing my theoretical work to practical experiments. Furthermore, I would like to thank **Ben Parslew** for his support in the earlier years of this work and **Kieran Wood** for his guidance in the latter year.

Next, a friend and a colleague who I could simply not have completed this work without, **Paul Broadley**. I cannot put into words the value of the camaraderie that we have developed over the years. However, I can say with certainty, our friendship is something I will always treasure.

Thirdly, and most importantly, my kind, infinitely supportive and unconditionally loving parents. Their unwavering support has kept me in a positive state of mind throughout the entirety of my research career. I am forever in debt for their kindness.

Finally, my deepest gratitude to all of my other friends and family who have always been there for me.

Chapter 1

Introduction

This chapter outlines the background, aims and contributions of this work.

1.1 Background

The field of Unmanned Aerial Vehicle (UAV) Traffic Management (UTM) is in its infancy. Over the past several years there have been many developments in vehicles, control architectures, regulations, communication methods, etc, which have enabled many low-density UTM experiments. In the near future, there is predicted to be a significant increase in the use of UAVs [1]. This will pose a challenge in ensuring conflict-free flights in these increasingly dense environments. There have been many proposals for the future architecture of UTM that fall into two main categories, centralised and distributed [2]. Centralised methods have two major shortcomings, a single point of failure and a large computational overhead for rapid re-planning during an emergency [3], [4]. Distributed methods are being explored as another option for control in a UTM [5]–[7]. The major shortcoming of distributed methods is the fact that the flight paths do not conflict prior to flight. This kind of problem is well known in the field of multi-agent systems [8].

This work organises traffic management systems into the following categories: navigation structures, traffic management design, and agent control. Navigation structures are components that can be used in a traffic management design. Traffic management design is the process of implementing components in a particular way to improve the performance of a traffic management system. Agent control describes how agents navigate the traffic system, including pathfinding, control methods, and emergency protocols. Figure 1.1 presents these components for a road traffic system.

In UTM systems that rely on distributed control methods, free flight is currently the most commonly used topology [9], [10]. Free flight (also known as basic flight and full mix) uses a

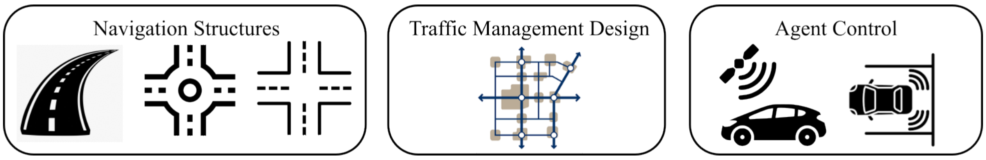
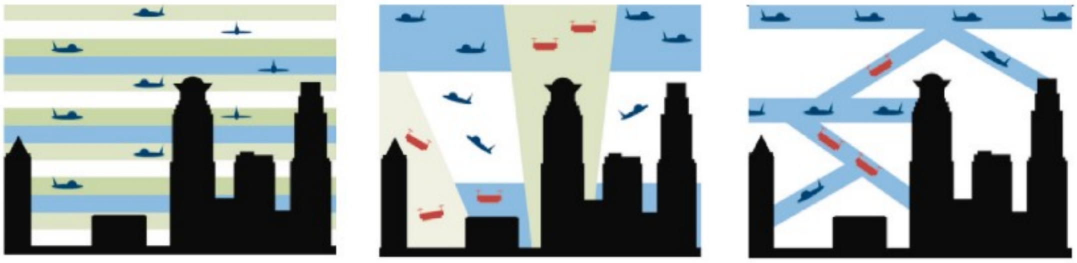


Figure 1.1. Components of a road traffic management system. Navigation structures like roads, freeways, roundabouts, etc are organised together in a traffic management design. Agent control describes the control methods and path finding of an agent



(a) Layers (b) Zones (c) Aerial Routing Networks

Figure 1.2. Examples of three types of pre-defined navigation structures researched in the Metropolis Case Study (Source: Hoekstra et al. (2015) [6])

distributed control method that does not have navigation structures [1], [6]. Agents are responsible for ensuring separation between each other while navigating to their destinations.

Within distributed control environments, several navigation structures have been proposed. In the body of work by Hoekstra et al. (2015) [6], Metropolis presents the concept of zones, layers, and tubes, Figure 1.2. Other proposals include slightly different implementations of zones, layers, and tubes, and others use combinations of each. Many of these works are merely proposals, and those that collect data are generally exploratory, leading to incomplete comparisons [11]. This is due to approaches that attempt to improve on all components of a traffic management system.

One of the several contributions of this work is on the evaluation of navigation structures. Various types of navigation structures are presented in Figure 1.3. In particular, this work focusses on aerial routing networks. These have many different names in the literature, including sky highways, sky lanes, corridors, tubes, skyways, airways, and aerial lanes. To evaluate navigation structures, a simplistic control method is developed. This control method is fit for evaluating navigation structures, but it is not proposed to be used in a final implementation. As

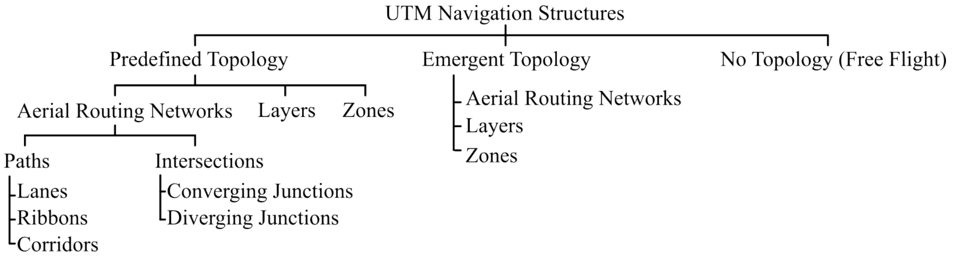


Figure 1.3. Hierarchy of airspace architecture terminology used in this thesis. Pre-defined navigation structures are defined globally and agents conform to the same architecture. Emergent navigation structures are not pre-defined and form due to common behaviours exhibited by agents. Free flight is exhibited when there are no navigation structures.

this field is very much in its infancy, metrics, terminology, test cases, and experimental methods are drawn from other fields and developed for use in this work.

Aerial routing networks are considered as either pre-defined, or emergent topology. Pre-defined topologies have a global geometry that is shared by all agents. Emergent topologies are generated dynamically based on inter-agent interactions.

This work also proposes a novel collision avoidance method and an improved congestion metric. The novel collision avoidance method falls under the agent control component of a traffic system. The improved congestion metric falls under traffic management design.

This thesis is composed of the following chapters.

- Chapter 2 - Literature review: Review of the state of current UTM literature, particularly aerial routing networks, guidance navigation and control systems, and measuring collision risk.
- Chapter 3 - Theory: Outline of the theory for pre-defined and emergent aerial routing networks, collision avoidance methods, and measuring congestion.
- Chapter 4 - Research Methods: Define the experimental setup within the simulation environments and the experimental procedures.
- Chapter 5 - Simulation Results: Presents results from simulation experiments
- Chapter 6 - Experimental Demonstration: Experiments and results of practical implementations of the proposed theory
- Chapter 7 - Advanced Concepts: Development of advanced intersections using helical tubes
- Chapter 8 - Conclusions and Future Work

1.2 Aim

The aim of this thesis is

To improve the performance of UTM systems through implementation and evaluation of novel navigation structures, control concepts and performance metrics

1.3 Contributions

The contributions of this work are threefold:

- The development, implementation, and evaluation of aerial routing networks using rule-based algorithms.
- The development of a novel collision avoidance method that improves efficiency in distributed control UTM systems.
- The advancement of a congestion metric used to measure the risk of collision within a UTM.

Chapter 2

Literature Review

This chapter provides an assessment of relevant preceding work in the field of UTM systems. The aim is to present the current state-of-the-art concepts and identify areas that require further efforts to advance the body of work. However, first, a brief history of air traffic management and traffic theory is presented. Following this, the reviewed literature is divided into three sections. The first section presents a review of navigation structures with a particular focus on aerial routing networks. The second section presents various methods of control available to autonomous UAV operations and focuses on distributed control methods. The third section reviews methods of measuring the risk of collision in multi-agent systems. The three sections presented have little overlap and are presented independently; hence, a summary of each is provided at the end of each section to highlight the key points in each field.

2.1 Brief History of Air Traffic Management

From late 1609, Hugo Grotius began to fight the doctrine that the ocean is private property, and free passage was not a right [12]. In his book *Mare Liberum*, which literally translates to *The Freedom of the Seas*, Grotius postulates that the sea should be international territory and that free passage was the right of any sea fairing trade. In the early 1900s, a similar debate was growing about the ownership of airspace. In 1917, a pioneer of the public airspace ideology, De Montmorency, promoted Hugo Grotius' doctrine for the sea to be adapted for airspace [13]. As air travel became more commonplace, the Convention on International Civil Aviation of 1944, known commonly as the Chicago Convention, saw the creation of new rights for all participating nations, titled *Freedoms of the Air* [14]. This set of nine rights allows a nation's airlines to have specific privileges to move within another nation's airspace. These rights remain the basis of aviation legislation to this day.

Before the Chicago convention, it became apparent that some level of control was required to ensure safe travel in aviation. The first Air Traffic Control (ATC) tower was built at Croydon Airport, London, in 1920 and was charged with providing information to pilots and sequencing aircraft for arrivals and departures through the use of radio telephony[1]. Since then, the concepts of air traffic control and airspace design have come a long way, however, the principles of traffic controllers sequencing traffic in busy corridors and into busy airports are still commonplace [15]. With the introduction of autonomous aerial vehicles, research has started on building a new model of airspace design and control to allow for the new era of aircraft.

Moving towards Autonomous Aerial Traffic Management (AATM), in 2015, NASA/FAA outlined its intentions to begin studies on a UTM system to enable low-altitude airspace operations safely [16]. Soon after, the initial Concept of Operations (shorthand ConOps) V1 was published with four Technology Capability Levels (TCL), which outlined a time scale for demonstrating various capabilities. This was a pivotal document in the industry and is widely adopted as a first draft for UTM policy.

[1] Croydon Airport - https://www.historiccroydonairport.org.uk/interesting-topics/air-traffic-control/

Since this initial UTM ConOps, many regulatory bodies have constructed frameworks for how AATM can be implemented, including the European Aviation Safety Agency (EASA), International Civil Aviation Organisation (ICAO), Federal Aviation Authority (FAA) in the United States and other civil aviation authorities in New Zealand, India, the United Kingdom and Switzerland [17]. In the United Kingdom, a proposal for a UTM framework was published along with several demonstrations performed to show the application of this framework [18]. This work outlines a very similar framework suggested by the FAA and NASA, however develops the ideas of several services are required for a UTM. These services include deconfliction and monitoring services. The general consensus between all authorities is that the majority of deconfliction should happen pre-flight, however tactical deconfliction strategies are available if required. In this thesis, this ideology is rejected for high-density environments and a framework for tactical deconfliction is outlined.

It should be noted that the importance of tactical deconfliction is outlined in [18], however, during the flight trials presented in this work, deconfliction was performed manually. In the recommendations by this work, tactical deconfliction should become automated.

Following the NASA ConOps, SESAR[2] (Europe) released a blueprint introducing U-space, a framework to allow all aircraft, including autonomous vehicles, to operate smoothly in all environments [19]. A detailed comparison between the United States and European approaches to UTM is presented in [20]. Several more traffic management systems were proposed from other government-led initiatives and industry-led initiatives, which are reviewed in [11].

More recently, there has been a shift to the concept of Urban Air Mobility (UAM) which is likely due to NASA releasing a ConOps V1 for UAM in 2020 [21]. These trends of interest are clearly visible in a keyword search in Web of Science, as seen in Figure 2.1. The keywords searched for UTM were: ‘Unmanned Traffic Management’, ‘Unmanned Aerial System (UAS) Traffic Management’ or ‘UAV Traffic Management’. The keywords searched for UAM were ‘Urban Air Mobility’. It can be seen that after the UTM proposal in 2015 there is a gradual increase until the completion of TCL 4 and the release of ConOps v2 by NASA/FAA in 2020, Figure 2.1. After this, there is a change of gear as the focus is directed towards UAM, after the

[2] Single European Sky ATM Research

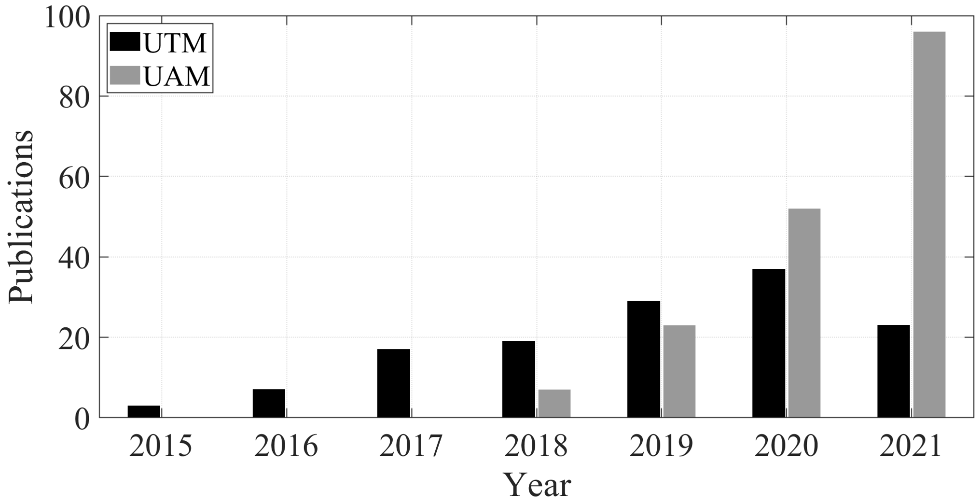


Figure 2.1. Recent trends in literature in the fields of UTM and UAM.

release of NASA/FAA ConOps v1 for UAM in 2020.

2.2 Modelling Traffic Systems

The seminal work on the modelling of traffic behaviour was carried out by Greenshields using a camera shown in Figure 2.2 (a) measuring the spacing of the cars and comparing it to their respective velocities. His results present a convincing argument that there is indeed a correlation between spacing and velocity, as seen in Figure 2.2 (c).

Following this, several more articles rigorously defined the variables involved in traffic modelling, including *flow*, *velocity*, and *concentration* (more commonly known as *density* in more recent work [23]) [24], [25]. These variables present several characteristic figures that apply to most traffic flows, reproduced in Figure 2.3.

Traffic flow is visually presented in a useful way using a position and time graph, where each line represents a vehicle, Figure 2.4. From this figure, it is possible to define the three aforementioned variables within a specified area of the figure. For the remainder of this section, the term *area* is defined as the product of dx and dt, as shown in Figure 2.4.

The flow in a given area is simply the number of vehicles that travel through the area and the

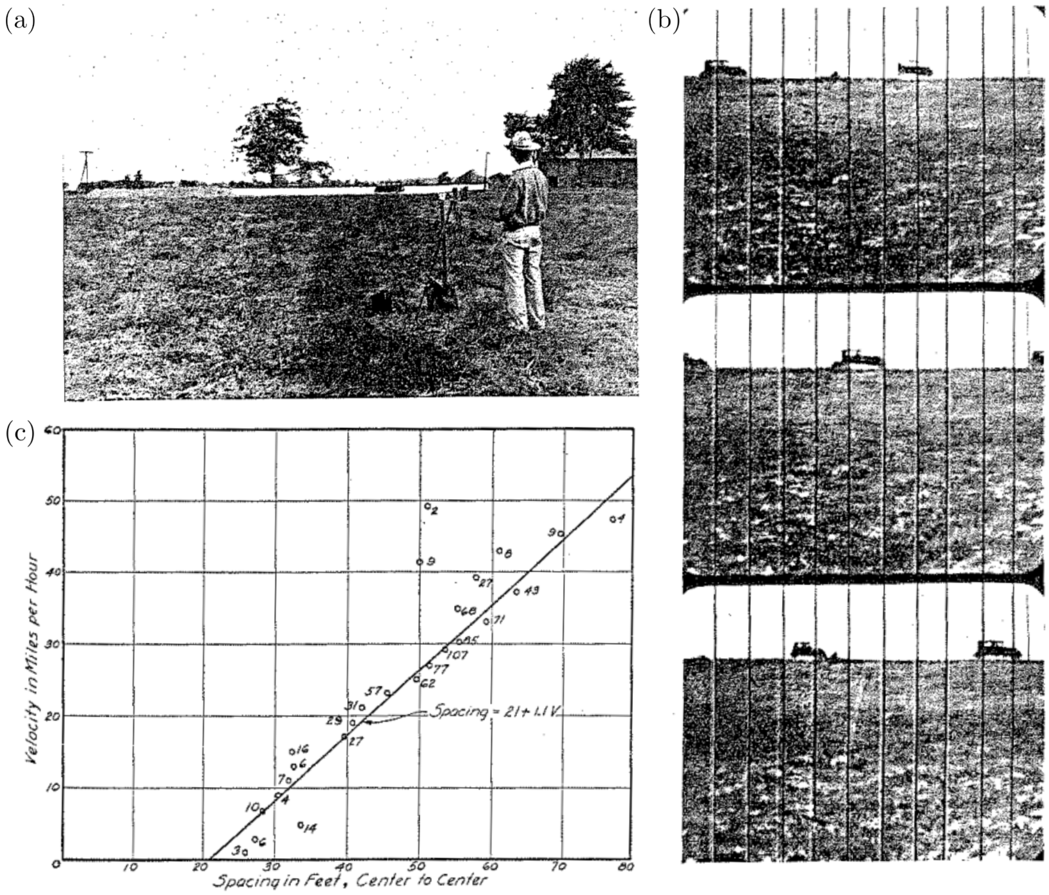


Figure 2.2. Greenshields seminal work on traffic behaviour. (a) Greenshields with his camera overlooking the road. (b) Sample images were captured in regular time intervals. Lines are projected onto the images afterwards using a projector to measure distances. (c) Results present a linear relationship between velocity and spacing. (Edited from [22])

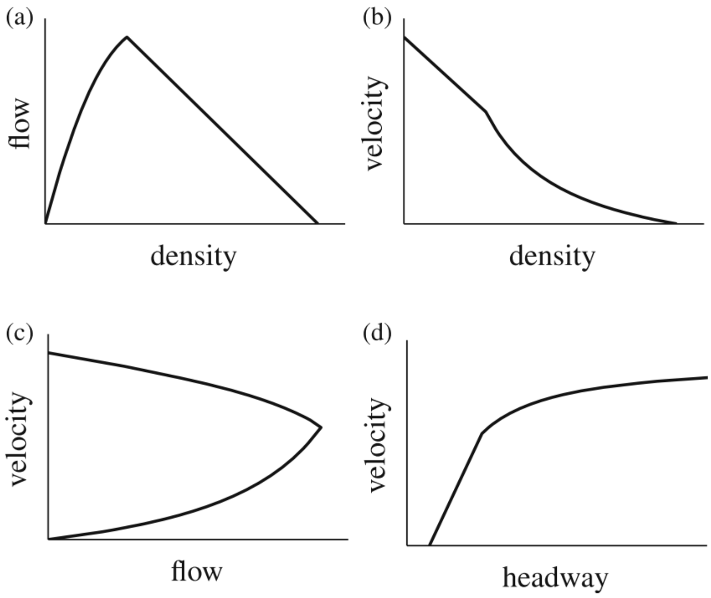


Figure 2.3. Characteristic traffic flow plots presenting relationships between the key metrics used in traffic theory. (Source: [23])

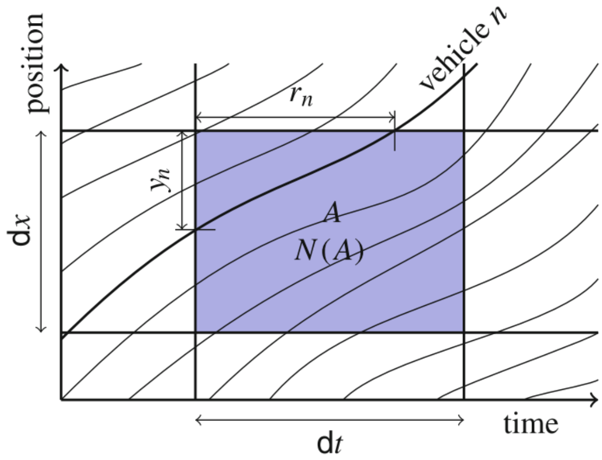


Figure 2.4. Diagram presenting variables used to quantify traffic theory metrics using position and time of vehicles along a road. (Source: [23])

distance they travel y_n.

$$q_{area} = \frac{\sum_{n=1}^{N} y_n}{\mathrm{d}x \ \mathrm{d}t} \tag{2.1}$$

The density in a given area is the number of agents and the amount of time they spend in the area.

$$\rho_{area} = \frac{\sum_{n=1}^{N} r_n}{\mathrm{d}x \ \mathrm{d}t} \tag{2.2}$$

The velocity in the given area is the total distance travelled divided by the total time spent.

$$v_{area} = \frac{q_{area}}{\rho_{area}} = \frac{\sum_{n=1}^{N} y_n}{\sum_{n=1}^{N} r_n} \tag{2.3}$$

It can be seen that these quantities were measured in a discrete fashion, resulting in averages over a given area. In 1955, Whitham and Lighthill wrote a seminal article describing the propagation of kinematic waves through traffic flows [24] by reducing the area size so that these waves can be resolved. Later, in 1991, methods were devised to make these variables continuous in space and time, allowing for local and instantaneous measurements [26].

To find the flow in a given location at a given time, a cross-section is first taken by taking the limit of d$x \rightarrow 0$. Subsequently, the time interval d$t \rightarrow 0$ is decreased to find the local and instantaneous flow.

$$q(x,t) = \lim_{\mathrm{d}t \to 0} \lim_{\mathrm{d}x \to 0} \frac{\sum_{n=1}^{N} y_n}{\mathrm{d}x \ \mathrm{d}t} = \lim_{\mathrm{d}t \to 0} \frac{N(x, [t, t + \mathrm{d}t])}{\mathrm{d}t} \tag{2.4}$$

Similarly, to find the density, a point in time is found taking the limit of $\mathrm{d}t \to 0$. Subsequently, the length $\mathrm{d}x \to 0$ is found to find a local and instantaneous density.

$$\rho(x,t) = \lim_{\mathrm{d}x \to 0} \lim_{\mathrm{d}t \to 0} \frac{\sum_{n=1}^{N} r_n}{\mathrm{d}x \ \mathrm{d}t} = \lim_{\mathrm{d}t \to 0} \frac{N([x, x + \mathrm{d}x]), t}{\mathrm{d}x} \tag{2.5}$$

Finally, velocity is defined simply as the division of flow by density.

$$v(x,t) = \frac{q(x,t)}{\rho(x,t)} \tag{2.6}$$

These types of continuous quantities are commonplace in traffic research today, where traffic flows are now considered analogous to gas flows in some fields of work [27], [28]. In many works, macroscopic quantities are used to measure the properties of a flow, for example, traffic pressure [26], [27], [29]. This quantity will be referenced later as a candidate for measurement of safety within a UTM.

2.3 Types of Navigation Structures in AATM

Within a distributed AATM, three main types of navigation structure are proposed. These are layers, aerial routing networks, and zones, as seen in Figure 1.2. Additional terminologies are found in the literature for other types of navigation structures; however, these are generally special cases of the navigation structures mentioned above. These navigation structures will be mentioned in the appropriate sections.

It should be noted that many articles in this field are proposals that come from national and corporate entities without much scientific backing. Nevertheless, these types of articles are included for completeness. The literature in this review is divided into four categories:

- Proposals - General reports that propose a navigation structure
- Exploratory simulation - Simulation of the performance of a navigation structure
- Comparative simulation - Simulation of a navigation structure in comparison to another navigation structure
- Practical experiment - Practical experimentation of a navigation structure

It should be noted that, although most research is performed to advance a field, predominantly for advancement of science/knowledge, there are many stakeholders in this research area [17]. This means that the expansion of this research area can be considered to be less coherent than other studies. Considerations made in this field include social, political, economic, safety and industrial. The majority of literature in the *proposal* category are advanced for a particular consideration. Literature that fall into the other categories, are generally performed purely to advance the field.

The purpose of this is to differentiate literature that simply proposes a navigation structure from literature that has found definitive conclusions of a navigation structure. Furthermore, there is a growing body of literature reviews that compare and contrast these proposals. These works are useful but plentiful. Therefore, to synthesise something novel, this review of navigation structures will introduce layers and zones, however, focus predominantly on aerial routing networks.

In the following sections, each of these navigation structures will be reviewed, followed by a discussion on the impacts of navigation structures on the performance of a traffic management system.

2.3.1 Layers

Layers are used to segment airspace by altitude into bands [6]. This kind of layering system has been used regularly in general aviation, known as the semicircular rule, where aircraft flying with a heading of between 0 and 179 degrees fly at certain altitudes and aircraft flying with a heading of between 180 and 359 degrees fly at other altitudes [30]. However, more recently, the use of the semicircular rule has come into question; the Federal Aviation Authority (US) still enforces its use ([31]), whereas the Civil Aviation Authority (UK) actively encourages pilots *not* to follow these principles [32].

Layers have been shown to be beneficial to the performance of an AATM. In the Metropolis project [7], it was shown that using layers has the lowest number of conflicts in simulation relative to using aerial routing networks, zones, or no structure. In [2], a simulation using a layering system with several hundred agents in a volume of $1.2km^3$ was able to deconflict all agents. It is clear that in these experiments layers were beneficial to the system; however, it should be noted that the comparisons are made to specific traffic management designs using layers, aerial routing networks, and zones. Therefore, it cannot be concluded that layers are the superior form of navigation structure due to one particular traffic management design performing better than another.

There are several proposed ways in which agents can be assigned to different layers. In [33], it is proposed that there are only two layers between ground level and 500 ft above ground level, similar to the semicircular rule, one for eastbound traffic and the other for westbound traffic. Similarly, in [34], 40 layers are proposed between 75 feet and 1050 feet that separate agents travelling in different directions and those turning. It should be noted that this article does not consider the current legislation limiting the altitude of UAVs to a maximum altitude of 400 feet (in most countries), which is considered in many other articles. This is important because there is currently much conflict between general aviation and UTM operations, which may cause the continuation of a maximum altitude of 400 ft for some time [35]. In [36], proposed layers are speed-based, with 0 to 200 feet reserved for low-speed vehicles and 200 to 400 feet reserved for high-speed vehicles. In [37], layers are separated based on vehicle type, low-level flight is reserved for light autonomous aerial vehicles, mid-level flight is reserved for heavy autonomous aerial vehicles, and upper levels are reserved for crewed aerial vehicles. In [2], agents are assigned to layers based on the duration of the journey, with shorter trips using lower layers and longer trips using higher layers. It should be noted that these are proposed methods with no experimental data to evaluate their performance.

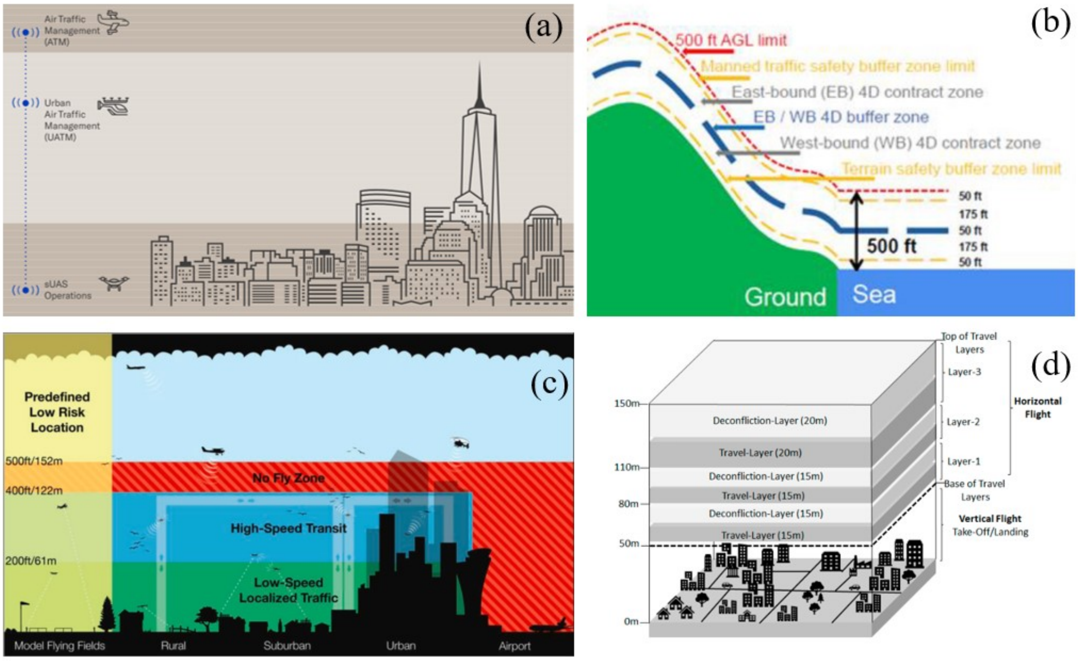
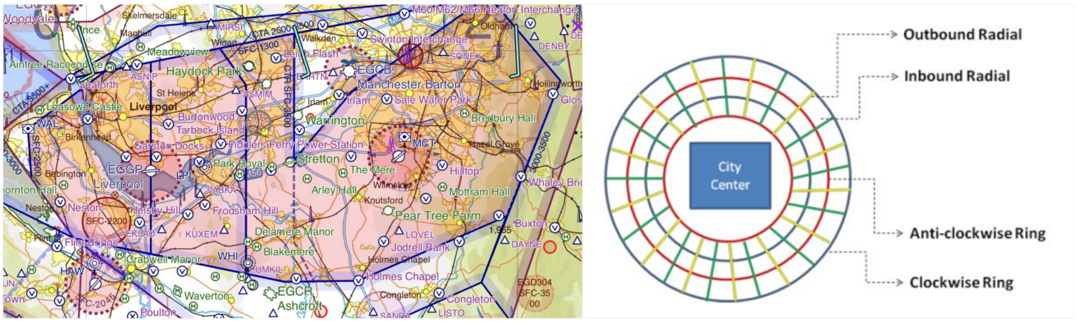

(a)
Air Traffic Management (ATM)
Urban Air Traffic Management (UATM)
sUAS Operations
(b)
500 ft AGL limit
Manned traffic safety buffer zone limit
East-bound (EB) 4D contract zone
EB / WB 4D buffer zone
West-bound (WB) 4D contract zone
Terrain safety buffer zone limit
50 ft
175 ft
50 ft
175 ft
50 ft
500 ft
Ground
Sea
(c)
Predefined Low Risk Location
500ft/152m
400ft/122m
200ft/61m
No Fly Zone
High-Speed Transit
Low-Speed Localized Traffic
Model Flying Fields
Rural
Suburban
Urban
Airport
(d)
Top of Travel Layers
Layer-3
Layer-2
Layer-1
Base of Travel Layers
Horizontal Flight
Vertical Flight Take-Off/Landing
150m
110m
80m
50m
0m
Deconfliction-Layer (20m)
Travel-Layer (20m)
Deconfliction-Layer (15m)
Travel-Layer (15m)
Deconfliction-Layer (15m)
Travel-Layer (15m)

Figure 2.5. Layers in Literature. (a) Assigned based on aircraft type [37]. (b) Assigned based on direction [33]. (c) Assigned based on speed [36]. (d) Assigned based on journey length [2].

There are some novel innovations in the field of layers, one of which is of particular interest. In [38], the number of layers changes with the density of agents in a given area, allowing vertical separation when appropriate, thus ensuring separation. This method is interesting because it is a form of adaptive navigation structure as the structure changes depending on the density of agents. There is little literature on other types of adaptive navigation structures, however, seems like a very interesting concept.

2.3.2 Zones

Zones segment airspace into volumes that can be traversed be particular agents. Zones are one of the most commonly used airspace navigation structures in crewed aviation, with the segmentation into different classes of airspace [39]. Zones can be used to separate agents based on any characteristic, for example, type of aircraft, speeds, priority, or operation type. The only proposed use of zones in the literature for AATM so far is in [7], where zones are created radially and concentrically around a city centre, as seen in Figure 2.6. The segmentation is based on travel direction, where agents moving radially inward to the city use the inward zones, and vice versa. Similar to the radial zones, the concentric zones alternate clockwise and counter-



(a) Manchester Airspace (b) Radial/Concentric Zones

Figure 2.6. Zones in Aerial Traffic Management. (a) Zones in crewed airspace around Manchester are segmented. (b) Zones described in [7]

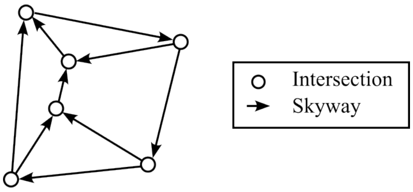


Figure 2.7. Aerial routing networks as directed graphs with paths and intersections defined in this work.

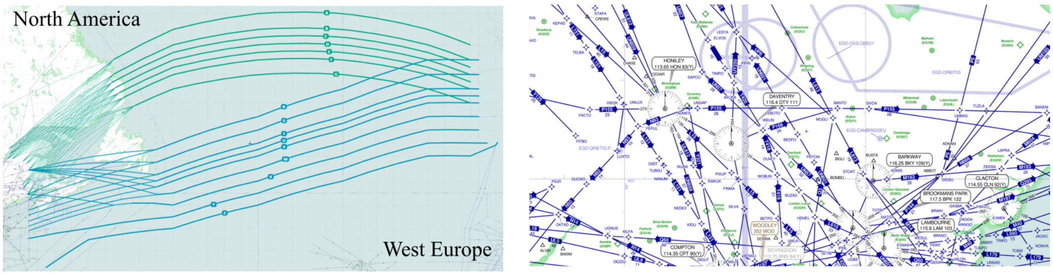
clockwise. It can be seen from the figure, the zones are very similar to aerial routing networks and restrict the volume of traversable space considerably, hence increasing the density within zones. This increase in density is likely the reason for the results showing zones were detrimental to the system and caused the most conflicts with respect to the other structures.

There is another example of a structure which is very similar to zones that are referred to as ‘cells’ [40]. Cells segment airspace for agents of similar requirements on airspace usage.

2.3.3 Aerial Routing Networks

Terminology

There are many terms used to describe aerial routing networks in the literature. These include sky highways, sky lanes, corridors, tubes, skyways, airways, and aerial lanes. In this work, we use the term *aerial routing network* to describe a network of *paths* and *intersections*. These form a directed graph as seen in Figure 2.7.



(a) North Atlantic Tracks

(b) Fixed Routes

Figure 2.8. Aerial routing networks used in air traffic control of crewed aircraft. (a) North Atlantic tracks on 16th September 2022 used to guide aircraft in regions of no radar coverage. Tracks are updated twice daily, green westbound and blue eastbound. (b) Fixed routes found all over the world with radar coverage. Waypoints (4 pointed stars) are connected with routes (blue lines). All routes and tracks have several altitudes at which they can be traversed to maintain separation. (Source: SkyVector)

Aerial Routing Networks Outside of AATM

The concept of paths has been used in civil and military aviation for many years. Generally speaking, a path is constructed between a series of waypoints defined by ground location and altitude. In crewed aviation, aerial routing networks are used readily, Figure 2.8. A notable set of paths are the North Atlantic tracks, which are updated twice a day to take advantage of current wind conditions and preferred routes by airlines [41]. These tracks are used to organise traffic as radar coverage is not possible in these areas. In Figure 2.8 (a), green tracks are westbound and placed to avoid the jetstreams (which flow eastward), and the blue tracks are eastbound and placed in the jetstream, to take advantage of the up to 115.7 m/s (416.5 km/h; 258.8 mph) tailwind [42]. In other regions, fixed routes can be found between fixed waypoints, as seen in Figure 2.8 (b).

The main purpose of using aerial routing network is to help manage the flow of traffic and to assist in aircraft separation [43]. Aerial routing networks are in use in almost all civil aviation; however, the National Air Traffic Services (NATS) in the UK and the Federal Aviation Authority (FAA) in the US have begun discussing the potential of aircraft being able to freely traverse the skies, allowing aircraft to fly optimal routes [44], [45]. This is interesting as it shows a move away from aerial routing networks and towards a free-flight model.

Table 2.1. Categorisation of publications with meaningful reference to aerial routing networks.

Category	Citation
Proposal	[1], [21], [37], [46], [47]
Exploratory Simulation	Scheduled Aerial Routing Networks [48]–[51] Corridrone [52]–[56] Others [57]–[59]
Comparative Simulation	Distributed Scheduling [3], [60] Metropolis [6], [7]
Practical Experiment	[61]

Literature Review Method of Aerial Routing Networks in AATM

An extensive search of aerial routing network was performed using Google Scholar with various combinations of: (*lane, tube, corridor, skyway, airway*) and (*UTM, U-Space, UAV Traffic Management, UAS Traffic Management*). Relevant papers were found and forward/backward citation searching was performed to find all documents relating to aerial routing networks. In total there were 22 publications that make a meaningful reference to the concept of aerial routing networks. These 22 publications have been tabulated into the four previously defined categories in Table 2.1.

It should be noted that several publications make distinctions between different types of aerial routing network structures, like lanes, planes and corridors; however, few publications outline a method for implementing them. Of the few publications that discuss the implementation, the explanation is limited to figures and description. This is a gap in the current body of research.

Description of Aerial Routing Network in Literature

Aerial routing networks have been proposed by many researchers as useful navigation structures for AATM. Within UAM, corridors are proposed to be reserved volumes that are primarily used for the movement of heavy autonomous aerial vehicles [21]. In [46], counter flow corridors are proposed, similar to a dual carriage way, to allow the flow of traffic along main arterials. The corridors are proposed to be stacked vertically or horizontally. In [1], corridors are proposed to

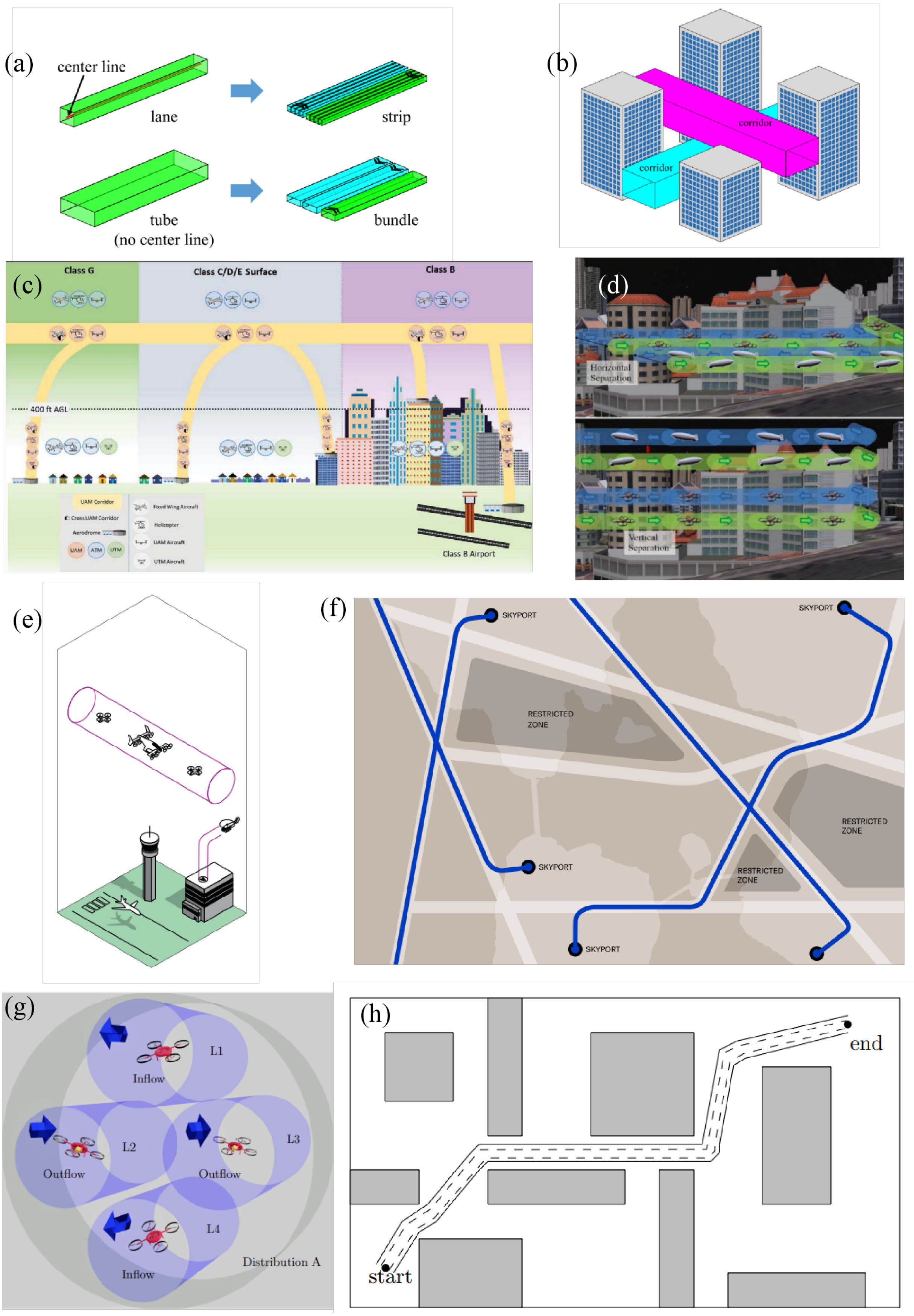


Figure 2.9. Review of skyways in literature. (a) Visual definition of lane, strip, tube and bundles [59]. (b) Visual definition of corridors [59]. (c) Proposed corridor utilisation in FAA UAM ConOps [21]. (d) Proposed corridor utilisation in Singapore [46]. (e) Proposed corridor utilisation by Airbus [1]. (f) Proposed corridors and restricted space by Embraer [37]. (g) Internal structure to skyway proposed by Corridrone [52]. (h) Routing of skyways between obstacles [54].

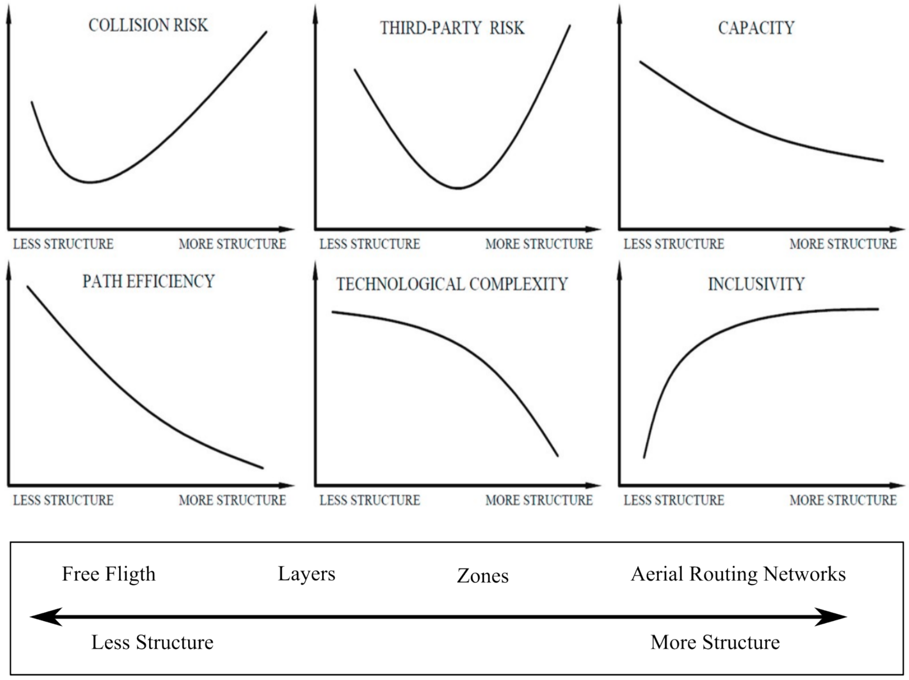


Figure 2.10. Hypothesis of results of navigation structure on the performance of an AATM (Adapted from: [11]).

be incorporated into a centralised system, where agents have pre-deconflicted paths to merge and unmerge from a corridor, but uses distributed control while in the corridor. In [37], a fixed network of corridors is proposed that agents can navigate between. In [52], the aerodynamic interactions between agents in a corridor are considered and an internal structure is proposed. In [54], a method is proposed to define corridors to navigate around obstacles. In summary, there are many propositions for the implementation of aerial routing networks, however, none of these works characterise or evaluate the performance in a meaningful way.

A review of AATM navigation structures [11] has presented hypotheses on the impact of navigation structures on several performance metrics, shown in Figure 2.10. It is hypothesised that with an increase in the constraints of a navigation structure (visually shown at the bottom of Figure 2.10), capacity and path efficiency are reduced, and collision and third-party risk are increased. On the other hand, technological complexities are reduced, and therefore the airspace is more inclusive to different types and capability aircrafts. These hypotheses appear to be based more on intuition than on conclusive evidence. For example, there is only one study with experimental results for the capabilities of zones [7], and only two studies with experimental results for aerial routing networks [6], [59], both of which have specific results for a limited range of traffic management designs. The majority of studies reviewed are concepts of operation

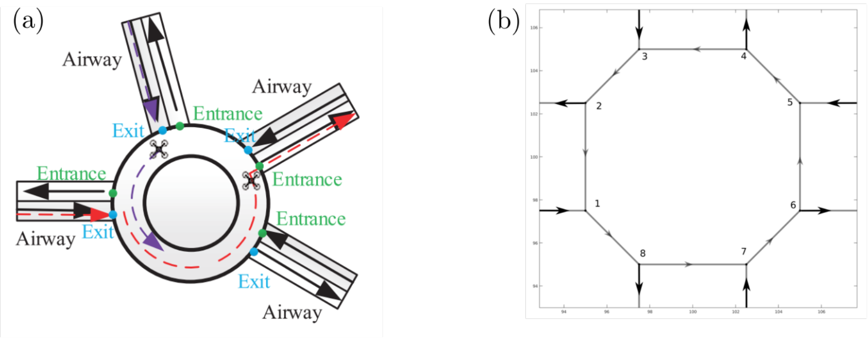


Figure 2.11. Roundabouts in AATM from Literature. (a) Continuous roundabout utilised in [61]. (b) Roundabout graph used with scheduler in [49]

proposals that do not have experimental data to support any performance conclusions.

Description of Intersections in Literature

There are two main types of intersections found in the literature, roundabouts and crossings. In [61], roundabouts are used when three or more paths intersect by implementing a central rotary hub to coordinate traffic flow, as seen in Figure 2.11 (a). Simulations and experiments showed that roundabouts allowed agents to coordinate successfully with each other. A repulsive force was used to ensure separation between agents. In [49], a roundabout is presented in a similar experiment to [61] however organises the roundabout into a graph and a centralised scheduler is used to ensure deconfliction, as seen in Figure 2.11 (b).

The other main intersection are crossings where paths intersect. In [59], several intersections are proposed, with various methods to allow flow to and from all directions. From Figure 2.12 (a) is a level crossing that uses a scheduling protocol to allow agents to flow in all directions, similar to a traffic light intersection, (b) is a stacked crossing that allows free flow straight and in one other direction, in the case of this example, turning right and a scheduled turn in the other direction, and (c) is also a stacked crossing that allows free movement in all directions. This method works when 4 directions are considered; however, if another direction is included, this intersection cannot be considered. Furthermore, free flow is available in (c), except for the strips on the top and bottom of the stack. The results of the experiments conclusively showed that the throughput and system-wide speed were significantly higher for the stacked crossing (c) than

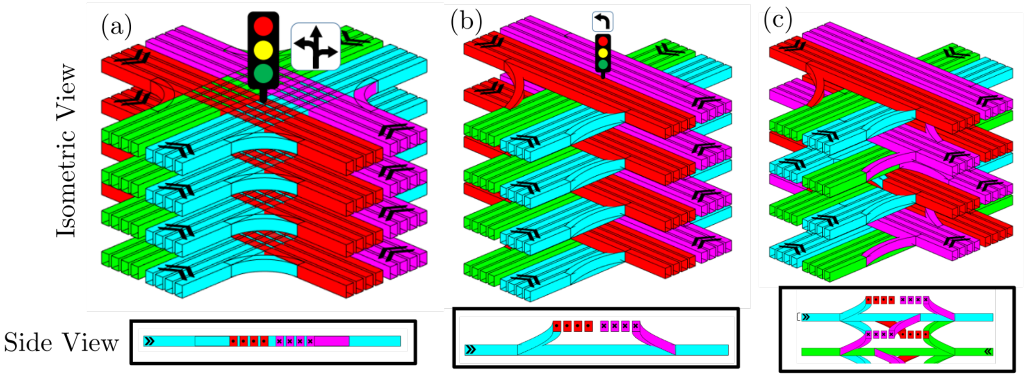
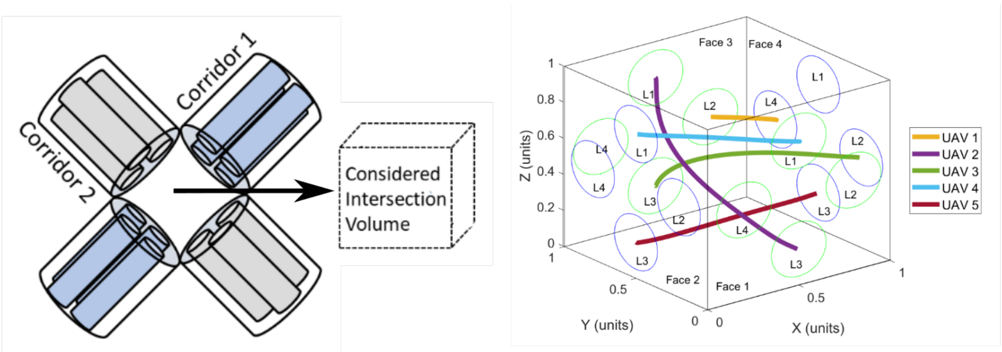


Figure 2.12. Intersections proposed in [59]. (a) Level crossing intersection using a signalling method for all movements. (b) Stacked crossing with slip lane to turn right and a signal to turn left. (c) Stacked crossing with slip lanes to turn right and left.



(a) Topology

(b) Agent Trajectory

Figure 2.13. Intersection design proposed in [56]. (a) The topology of the surrounding paths in a Corridrone formation. (b) The trajectories of agents within the considered intersection volume.

when using a level crossing (a). No results were presented for stacked crossings with free flow in one direction (b). This result is intuitive, as highway interchanges in road traffic allow for much higher throughput than a level traffic crossing.

A similar crossing presented in [56] is based on the Corridrone lane structure [52]. The proposed intersection is controlled by a centralised path planning algorithm which computes pre-defined conflict-free paths for agents to follow. No simulation results are presented for throughput.

Agent Control within Aerial Routing Networks

In [56], multiple parallel 1D paths (lanes) run parallel to each other. A lane-changing algorithm is proposed by finding a neighbouring lane where a transition is feasible. When a feasible lane change is found, a Bezier curve is generated for the agent to follow to change lane. In [59], a

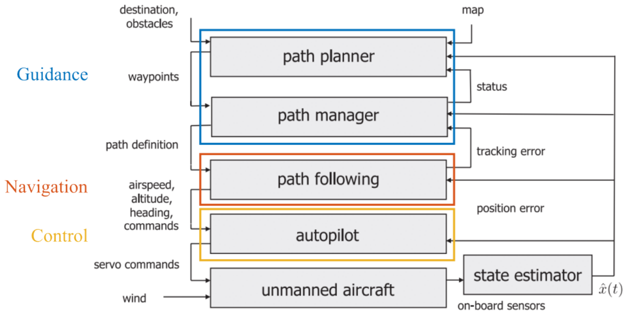


Figure 2.14. Block diagram of UAV flight management system. (Adapted from: [62])

similar approach to lane-changing is considered; however, agents in a lane make changes to their velocities to make space for the incoming agent. If space is not available or velocities cannot be matched, the lane change is rejected.

To the knowledge of the author, there are no publications specifying the methods of how an agent will merge into or unmerge from a skyway. This is likely due to perceived triviality of the problem[3], however, in the authors' opinion should not be overlooked.

2.3.4 Summary

Three types of navigation structures have been reviewed. There exists a limited body of work on the use of zones within an AATM. This appears to be inconsistent with the fact that zones are readily used within crewed air traffic management. There exist many proposals for use of layers, however, there is a limited body of exploratory research in the field to support this. Similarly, aerial routing networks are frequently proposed within many bodies of work; however, there exist few exploitative studies on their performance.

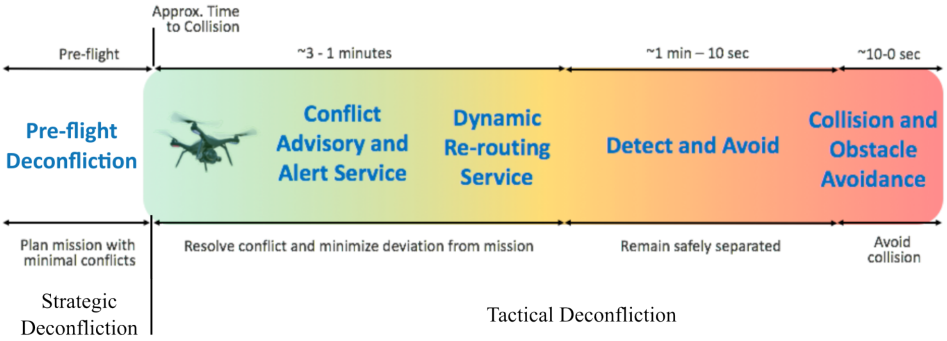

Approx. Time to Collision
Pre-flight
~3 - 1 minutes
~1 min – 10 sec
~10-0 sec
Pre-flight Deconfliction
Conflict Advisory and Alert Service
Dynamic Re-routing Service
Detect and Avoid
Collision and Obstacle Avoidance
Plan mission with minimal conflicts
Resolve conflict and minimize deviation from mission
Remain safely separated
Avoid collision
Strategic Deconfliction
Tactical Deconfliction

Figure 2.15. Methods of ensuring safety within a UTM as outlined in [64]. Strategic deconfliction minimising conflicts pre-flight. Several layers to tactical deconfliction to minimise risk. (Adapted from: [64])

2.4 Guidance, Navigation and Control

2.4.1 Overview

Flight management systems in AATM requires several components. Often, these components are split into Guidance, Navigation and Control (GN&C). A survey of various GN&C methods for UAVs is found in [63]. A useful visualisation of the components of GN&C in a block diagram is shown in Figure 2.14. These components are generally for a single agent. For multiple agents, deconfliction is also required. In this review, the focus is on the guidance component of GN&C, which includes deconfliction between agents.

In [64], deconfliction is categorised into two groups, strategic and tactical deconfliction, Figure 2.15. Strategic deconfliction aims to minimise the risk of collision pre-flight. Tactical deconfliction is performed in-flight and has many layers to reduce risk. In the following sections, methods for both types of deconfliction are reviewed.

2.4.2 Strategic Deconfliction in Guidance Systems

Strategic deconfliction, also known as pre-flight Conflict Detection and Resolution (CDR) [65], generally uses a process called cooperative multi-agent path finding (MAPF) to ensure that all agent paths are deconflicted before flight. MAPF is a well-defined field for Multi-Agent Systems

[3] In [59], this concept is glossed over in a single sentence.

(MAS) with many solutions; a good review of methods for UAVs is [66]. Many of these methods have been tested in simulation, including rule based algorithms [67], search based algorithms [68] and reduction based algorithms [69]. One of the challenges of MAPF in an AATM is the fact that it must be continuous [70]. It is possible to achieve an optimal solution when all starting points and destinations are known before agents begin travelling; however, the challenge is apparent when agents have already begun travelling and a future path must be planned. In one study, improved solutions were found when the MAPF algorithm was able to schedule starting times and vary the agent's velocity during flight [71]. Other challenges to MAPF include the large computing overhead for rapid re-planning if an agent diverges from its course or due to an emergency. Furthermore, MAPF does not scale well: it is classified as NP hard, which means that the problem is always at least as hard as the hardest problem and generally harder [72]. In conclusion, in a successful MAPF system, safety and efficiency are high; however, there are many challenges to overcome.

2.4.3 Tactical Deconfliction in Guidance Systems

Tactical deconfliction methods, also known as in-flight CDR methods, can be categorised into centralised, distributed, and independent [73]. A centralised method has a central entity that performs the path planning for agents in real-time and communicates the results to all agents in the environment, similar to crewed ATC [74]. Distributed methods have agents communicating with each other and path planning computed on-board to avoid a collision, similar to T-CAS [75]. Independent methods use on-board sensors to detect and path plan[76]. In this work, we focus on distributed methods of tactical deconfliction.

One of the first control methods for tactical deconfliction in aerial MAS was designed for animating a flock of birds. This article is the seminal work in what is now known as *Reynolds Flocking*, named after the author of the paper [77]. The control method combines a distributed tactical deconfliction method within the guidance system by using a set of linearly summed rules. Flocking has since become a field of its own, with rules being adapted for different purposes [78]–[86]. A particularly interesting implementation of flocking is in a practical demonstration

Figure 2.16. Long exposure of flocking experiment in confined environments (Source: [86]).

of the rules in [87], where flying wings demonstrated the ability to flock with short range communications. Another practical demonstration was presented in [86], where agents were forced to flock in confined environments, as seen in Figure 2.16.

Several more methods of tactical deconfliction methods include the following. In [88], V2V communications are used to detect a potential conflict and categorise the conflict to determine which agent has the right-of-way. Once the right-of-way has been identified, an evasive manoeuvre is carried out by one or both agents. Using artificial potential fields [9], simulates obstacles which applies an artificial force on the agent resulting in an evasive manoeuvre. A vector-sharing resolution [89], which analytically calculates evasive manoeuvres for both agents in a potential conflict to ensure minimum separation. The Metropolis study used a Modified Voltage Potential (MVP) algorithm, which calculates the minimum required heading change to resolve the conflict [6]. Finally, optimal reciprocal collision avoidance (ORCA) [10], which calculates the maximum safe velocity for agents to travel toward a target without causing a conflict. ORCA is designed predominantly for environments with very high density. All algorithms have shown some levels of success, however, are optimised for different problems and have varying levels of complexity and, therefore, computation cost.

2.4.4 Summary

There are many methods to ensure safety with a UTM. There are several research groups that are working on the challenges in MAPF for use within a UTM. If successful, this method can ensure high levels of efficiency and safety within a UTM. Within the reviewed literature, there are significantly more works considering in-flight deconfliction. This may be due to in-flight deconfliction being a simpler problem or researchers believe that, even when most of the MAPF problems are solved, it will still be less effective than in-flight deconfliction.

2.5 Measuring Risk of Collision in Multi-Agent Systems

2.5.1 Introduction

Measuring safety is a general problem for all Multi-Agent Systems (MAS). Safety is a difficult concept to measure, especially since there are many hazards to consider. To simplify this, the only hazard reviewed in this section is collisions between agents. *Collision* is defined as when two agents physically collide.

It is useful to break risk into *severity* and *probability*, which is commonly done. Severity is constant in most cases, including loss of vehicles or harm to people and property damage. Probability can change dramatically, and hence is reviewed in detail. The names of metrics used to measure the probability of collision commonly found in the literature include *probability of collision*, *collision rate*, *number of collision*, *expected number of collision* or *risk of collision*.

The risk of collision has historically been divided into geometric and causation [90]. The geometric probability is based on the geometry of the situation. Causation is the probability of not avoiding the collision. In this review, focus is placed on the geometric probability. All terminology is visually described in Figure 2.17.

The motivation to explore this field is exemplified when considering the number of collisions found in various MAS. In the commercial maritime industry, between 2011 and 2020, there

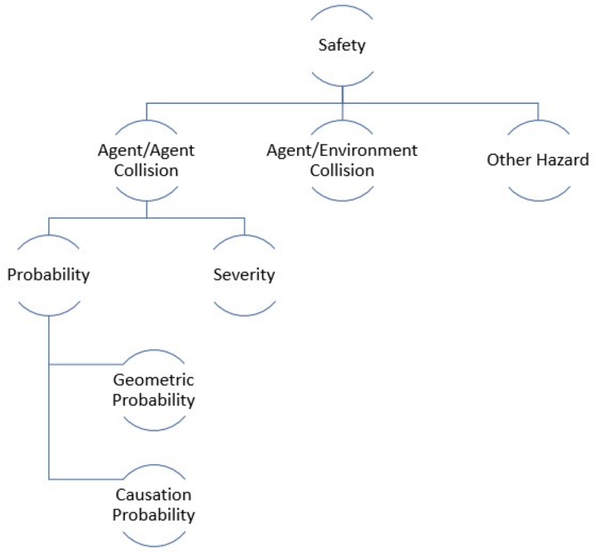


Figure 2.17. Hierarchy of terminology in safety for MAS defined in this work.

were 3288 reported collisions between vessels weighing more than 100 tonnes, 27 of which resulted in the loss of a ship [91]. In general aviation, in the US alone between 1991 and 2001, there were 15.6 mid-air collisions per year on average [92]. In road traffic, it is estimated that there are between 20-50 million non-fatal injuries worldwide every year due to road collisions [93]. Within commercial aviation, between 1996 and 2015, there have been 13 notable mid-air collisions involving commercial aircraft [94].

In what follows, the various methods for measuring geometric probability of conflict is reviewed for different MAS. Important conclusions show that there is an intuitive relationship that is yet to be quantified in the literature.

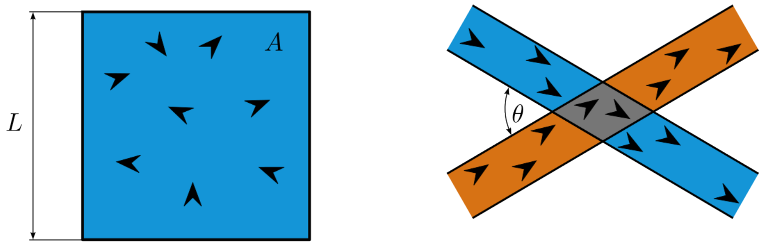
2.5.2 Measuring Geometric Probability of Conflict in MAS

All equations used from literature have been adapted to use a standard set of variables. All variables are defined in Table 2.2.

Within this body of work, there are generally two types of environments which are examined for geometric probability of conflict, referred to here as *Uniform Random* and *Cross Flows*. Uniform random is the environment where agents move freely independent of other agents. Cross flows is when there is a crossing of flows and collisions are measured in the intersection. These are represented visually in Figure 2.18.

Table 2.2. Standard variables for equations.

Variable	Definition	Units
N	Number of agents	
d_{col}	Collision boundary diameter	m
$\bar{v}$	Average velocity of all agents	ms^{-1}
ρ	Number density	m^{-3}
V	Volume	m^3
L	Characteristic Length	m
P	Probability of conflict	
v_i	Velocity of flow i	ms^{-1}
v_{ij}	Relative velocity between flow i and j	ms^{-1}
θ	Angle between flows	$^\circ$
E	Expected number of conflicts	
k	Specific motion of the agents [95]	
n	Number of pairs of agents	
A	Area of environment (2D)	m^2
W	Width of fairway	m
Q	Inflow rate	s^{-1}
f	Inflow distribution function	
x, y, z	Position of agent	m



(a) Uniform Random

(b) Cross Flow

Figure 2.18. Environments considered in geometric probability of conflict. (a) Uniform random has agents moving in any direction around a certain area/volume. (b) Cross flow has agents flowing along paths that intersect (grey region). Collisions are measured within the grey region only.

Within the following derivations, it is noted that there are common terms in equations, specifically the use of density and velocity. Intuitively, the density of a traffic system should have an impact on the expected number of collisions, however, the level of coordination in the agents velocity also, should have an impact on the expected number of collisions.

Starting with uniform random environments, [96] is a maritime study that derives from first principles, the expected number of collisions per unit time between agents travelling with uniformly random velocity distributions.

$$E = \rho L \frac{8d_{col}}{\pi} \tag{2.7}$$

Where E is the expected number of collisions, ρ is the agent number density, L is the characteristic length of the fairway being examined, and d_{col} is collision diameter of a vessel, assumed to be circular. The assumption that all agents are travelling in uniformly random directions with constant speed allows for the integration over all relative velocities, and as a result yields the factor of π. The conclusion from this equation is that the expected number of conflicts per unit time and area is proportional to the density of agents for uniformly random motion.

In a study of UAVs [97], the expected number of collisions is derived from a kinematic model of an ideal gas in [98]. The derivation for a 2D experiment of flying fixed-wing aircraft is presented as:

$$E = kn(n-1)\frac{1}{A}\frac{d_{col}}{2}\sqrt{2}\bar{v} \tag{2.8}$$

Where n is the number of*pairs* of agents, due to the pairwise nature of collisions, A is the area where agents are free to traverse, and k is a variable which is dependent of the specific motion of the agents. Small values of k denote strong coordination between agents, with perfectly coordinated constraints reducing k to 0. In this equation, motion is *not* assumed to be uniformly random, and direction has an impact on the expected number of conflicts. This principle is loosely related to crossing flows, as seen in Figure 2.18

There are several studies of cross flows as they are fairly common in traffic systems. In [96], there is further consideration into the probability of collisions when considering crossing flows of maritime vessels. This model assumes uniform distribution of agents across the width of the fairway.

$$P = \frac{d_{col}W\rho_1}{v_2}\left(\frac{v_2}{\sin\theta} - \frac{v_1}{\tan\theta} + v_1\right) \tag{2.9}$$

Where ρ_1 is the density of flow 1, v_1 and v_2 are the velocities of flow 1 and 2, respectively, W is the width of the fairway and θ is the crossing angle of the two flows. From this equation, it can be seen that the probability of a conflict is dependent on the crossing angle and the density.

One of the founding articles on maritime collisions presents a similar equation however assumes a normal distribution across the width of the fairway, as shown in Figure 2.19. The

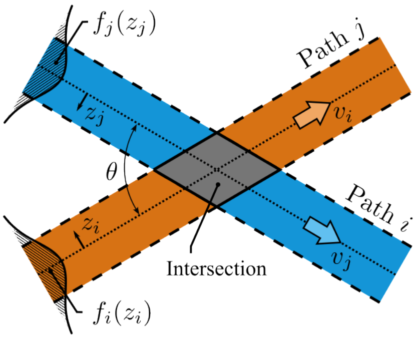


Figure 2.19. Crossing flows with agents distributed normally along the width of the fairway. Two paths, blue and orange, cross to make an intersection in the grey region. Adapted from [99]

derived equation for expected number of collisions per unit time is

$$E = \sum_i \sum_j \iint_{z_i, z_j} \frac{Q_i Q_j}{v_i v_j} f_i(z_i) f_j(z_j) v_{ij} d_{col} dA \tag{2.10}$$

Where i and j as subscripts are the identifiers for the respective flows, Q is the inflow rate of agents per unit time, f is the distribution of agents across the width of the path and v_{ij} is the relative velocity between the flows. The relative velocity is dependent on the angle between the flow, showing that as the angle increases, so too does the relative velocity and hence the expected number of collisions.

Two other studies which present findings which show that velocity coordination has an impact on expected number of conflicts. In [92], attention was brought to the fact that there is a difference in collision frequency of civil aircraft dependent of closing angle, categorised into head-on, obtuse angle, acute angle and head-tail. In [100], four types of flight were considered, from uniform random to a highly coordinated stream of agents.

A common pattern found within this literature is intuitively there is a relationship between the coordination of agent motion and the collision rate. However, there is yet to be found a way of effectively quantifying this coordination between agents.

2.5.3 Summary

In the review of geometric probability of collisions in MAS, there are many models used to quantify the risk of collision. The models have a range of approaches, however, generally always share a density and velocity term. It is intuitive that density has a role to play in collision rate. The velocity term is more complex as it must incorporate the coordination between agents. An attempt at quantifying coordination is in [97], however the coefficient is found experimentally.

Chapter 3

Theory

In this chapter, the theory developed in this work is presented. In the introduction of this thesis, traffic management systems were categorised into navigation structures, traffic management design and agent control. In this work, contributions are made to each of these categories. The first contribution is the design of a practical way to implement aerial routing networks. Rule-based algorithms were selected as a simple and effective way to implement both pre-defined and emergent aerial routing networks. The second contribution is the development of a novel deconfliction method, orthogonal separation, that acts in a more efficient manner than current common practices. Orthogonal separation is also constructed using rule-based algorithms. Finally, congestion, a way of measuring the safety within a traffic management system is defined with several potential methods for quantifying.

3.1 Rule-Based Algorithms

3.1.1 Introduction

Rule-based algorithms are a method of control that uses a set of rules to navigate an environment. The main output of a rule-based algorithm is a total goal velocity for a given agent, $\mathbf{v}_i^g$. This goal velocity is the sum of the goal velocity of all rules, $\mathbf{v}_i^{g,r}$, where $r \in \alpha$ is the set of all rules that are used. A gain is used to weight each rule in the final summation. This gain is unique to each rule, K^α. Rules are constructed from three components, referred to as rule magnitudes M, rule directions $\mathbf{D}$ and sums. The combination of these components is at the designers' discretion to achieve some desired behaviour. Figure 3.1 shows how all these components fit together in a rule-based algorithm.

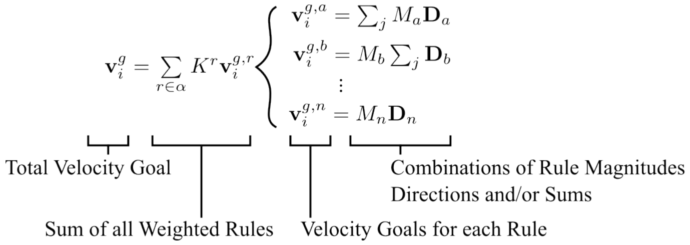

$\mathbf{v}_i^g = \sum_{r\in\alpha} K^r \mathbf{v}_i^{g,r}$
$\mathbf{v}_i^{g,a} = \sum_j M_a \mathbf{D}_a$
$\mathbf{v}_i^{g,b} = M_b \sum_j \mathbf{D}_b$
$\vdots$
$\mathbf{v}_i^{g,n} = M_n \mathbf{D}_n$
Total Velocity Goal
Combinations of Rule Magnitudes
Directions and/or Sums
Sum of all Weighted Rules
Velocity Goals for each Rule

Figure 3.1. Outline of all general components within a rule-based algorithm.

In figure 3.2, an example of a rule-based algorithm is implemented that causes agents of the same colour to attract and those of different colours to repel.

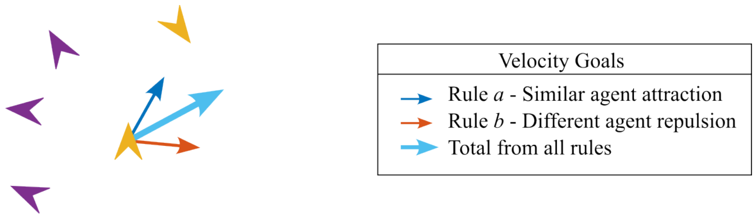

Velocity Goals
Rule a - Similar agent attraction
Rule b - Different agent repulsion
Total from all rules

Figure 3.2. Example of a rule-based algorithm that causes agents of similar colours to attract and different coloured agents to repel. Two rules are designed, rule a causes agents of the same colour to attract, rule b causes agents of a different colour to repel. For the central agent, the result of both of these rules are shown by the velocity goal vectors. The total velocity goal is also shown.

Once a total velocity goal is generated, it is used to compute a force that is applied to the agent. The applied force is computed by subtracting the current velocity from the total goal velocity, Figure 3.3. If the velocities are the same, no force is applied; if they are different, a

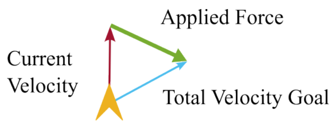


Figure 3.3. Vector diagram of how a force is computed and applied to an agent based on current velocity and total goal velocity.

force is applied.

It should be noted that the concept of rule-based algorithms is not novel. Many flocking implementations are built using these principles [79], [81], [86]. However, in many of these works, the process of rule design is application-specific. The aim of this section is to generally define a rule-based algorithm that can be used in any similar future work.

In this section, each of these components is described. However, first agent kinematics is outlined, including collision kinematics. Next, the concept of a rule and best practices for rule design are outlined. This includes some examples of rule directions and rule magnitudes followed by some completed example rules. Finally, the actuation method of the total goal velocity on the agents is formalised. The theory outlined here is used in the following sections to build and test components that are part of a larger project.

3.1.2 Agent Kinematics

Agents are modelled as point masses that move in a 3D space xyz with speed u. It is assumed that agents are implicitly able to support their weight through the action of an unspecified control system such that the effect of gravity on the dynamics can be ignored. The position and velocity of the agent are $\mathbf{x} = [x\ y\ z]$ and $\mathbf{v} = [\dot{x}\ \dot{y}\ \dot{z}]$, respectively, and $|\mathbf{v}| = u$.

The motion of the agent is controlled by applying an acceleration $\mathbf{a} = [\ddot{x}\ \ddot{y}\ \ddot{z}]$. The agility of the agent is defined by the maximum acceleration, a_{max}, available to it. For simplicity, we assume that an agent can accelerate equally in all directions. This is clearly not the case in practice, e.g. the magnitude of acceleration available downwards parallel with gravity will be larger than the equivalent upward acceleration. However, this approximation is fit for the purpose of the work in this thesis in that it allows objective comparison of the performance of different rules. Vertical and horizontal accelerations can be uncoupled in future work with modest changes to the code if needed.

Agent dynamics is characterised by a reference length, l_c defined as.

$$l_c = \frac{u_{\text{ref}}^2}{a_{\text{max}}} \tag{3.1}$$

where u_{ref} is the nominal cruise (reference) speed of the agent and a_{max} is the maximum acceleration as defined above. The reference length is equal to the turn radius of an agent travelling at cruise speed with maximum acceleration applied normal to the flight path direction, or equivalently twice the distance required to bring the agent to rest from cruise when decelerating at maximum acceleration.

3.1.3 Collision Kinematics

Agents are modelled as point masses with two critical radii, a conflict distance d_c and a collision distance[1] D_c. The conflict distance is the distance at which, if another agent enters, a deconfliction method should be enacted to avoid a collision. The collision distance is the distance at which, if another agent enters, a collision has occurred.

We define the Dimensionless Conflict Distance (DCD) as d_c/D_c, where a DCD of 2 has a deconfliction distance twice the size of the collision distance. The larger this ratio, the more conservative the deconfliction method. However, the larger the ratio, the fewer agents that can remain within a given volume without overlap, and hence airspace efficiency is reduced.

3.1.4 General Rule Design

The method used here to design various rules was as follows.

1. **Define the primary requirements of the rule** - What is the desired behaviour including all edge cases?

[1] A memory item to remember which symbol is which is the lower case is a conflict and is not too serious, whereas a collision is SERIOUS and is hence capitalised.

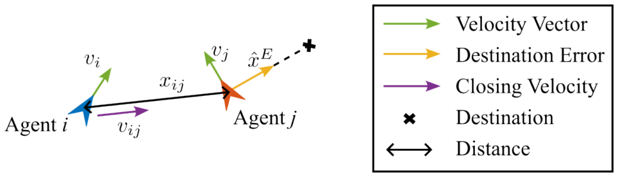


Figure 3.4. A sample of agent attributes that can be used as inputs in a rule-based algorithm.

2. **Formulate a rule** - Mathematically describe the magnitude and direction that a rule should output to achieve the desired behaviours.

3. **Individual rule test** - Implement rule formulation in several edge cases to ensure the agent performs as expected.

4. **Full rule test** - Implement the rule in a full-scale test case.

3.1.5 Rule Magnitudes and Directions

Rule magnitudes, rule directions and sums are the basic components that make up any rule. Generally speaking, a rule magnitude is a scalar quantity, and a rule direction is a unit vector. Both are made up of inputs from agent attributes or from the environment. Several examples of inputs that can be used are found in Figure 3.4. In this work, rules are generally constructed as the product of several rule magnitudes and a rule direction. The rule magnitudes are generally normalised, but not in all cases. The result of each rule is a specific goal velocity.

Several example rule magnitudes and directions are outlined below.

Rule Magnitude Based on Distance between Agents

In this rule magnitude, a weighting is applied based on the proximity between two agents. For agents that are closer, a larger weighting is applied, and vice versa.

Let $\mathbf{x}_{ij}$ be the position vector of an agent i in the population relative to agent j,

$$\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j \tag{3.2}$$

with length

$$x_{ij} = |\mathbf{x}_{ij}| \tag{3.3}$$

The rule magnitude between agents i and j based on distance (superscript x) is based on a hyperbolic function with a maximum:

$$M^x_{ij} = \min\left(1, \frac{D_c}{x_{ij}}\right) \tag{3.4}$$

Here, M is the rule magnitude and D_c is the collision distance, which is used to normalise the function.

Rule Magnitude based on Angle between Agent Velocities

In this rule magnitude, a weighting is applied based on the similarity of the velocity vectors between two agents. For agents travelling in similar directions, a large weighting is applied, and vice versa.

This weighting by alignment of velocity vectors (superscript θ) is implemented by taking the dot product of the velocity vectors of agent i and j:

$$M^\theta_{ij} = \frac{1}{2}\left(\hat{\mathbf{v}}_i \cdot \hat{\mathbf{v}}_j + 1\right) \tag{3.5}$$

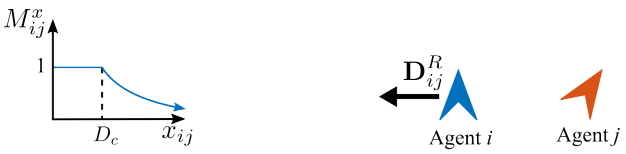
When the velocities of agent i and j are parallel, the weight is equal to 1 whereas if they are anti-parallel it is equal to 0, as required.

Rule Direction Away from Neighbouring Agent

This rule direction generates a repulsive effect between neighbouring agents (superscript R for repulsion). It should be noted that all rule directions must be unit vectors.

$$\mathbf{D}^R_{ij} = -\hat{\mathbf{x}}_{ij} \tag{3.6}$$

Here D is short for direction.



(a) Rule magnitude based on distance between agents

(b) Rule direction away from neighbouring agent

Figure 3.5. Components used in the example repulsive rule. (a) Rule magnitude against distance between agents. (b) Rule direction repelling from neighbouring agent.

Rule Direction Orthogonal to Velocity Vector and Towards Neighbouring Agent

This rule direction applies a goal velocity to agent i that is orthogonal to the current velocity vector of agent i and towards a neighbouring agent j. As this rule is always applied orthogonal to the direction of motions, it can be considered a side force (superscript SF).

$$\mathbf{D}^{SF}_{ij} = \frac{(\hat{\mathbf{v}}_i \times \hat{\mathbf{x}}_{ij}) \times \hat{\mathbf{v}}_i}{|(\hat{\mathbf{v}}_i \times \hat{\mathbf{x}}_{ij}) \times \hat{\mathbf{v}}_i|} \tag{3.7}$$

This equation can be simplified using triple product rules; however, it is more understandable in this format, as it is possible to use the right-hand rule to infer the output direction.

3.1.6 Rule Examples

It is possible to combine these rule magnitudes and directions with summations to construct rules with various behaviours.

Repulsive Rule Example

The primary function of this example repulsive rule is to repel agents when they are too close. The effect should be large when the agents are close and small when they are far apart. This repulsive rule will be denoted by $\mathcal{R}$ and is constructed using the following magnitude and direction:

$$\mathbf{v}^{g,\mathcal{R}}_i = \sum_j M^x_{ij}\mathbf{D}^R_{ij} = \sum_j \left(\min\left[1, \frac{D_c}{x_{ij}}\right]\right)(-\hat{\mathbf{x}}_{ij}) \tag{3.8}$$

Here, $\mathbf{v}_i^g$ denotes the goal velocity for agent i. Figure 3.5 visually presents the components used in this rule.

Velocity-Based Attractive Rule Example

The primary function of this example attractive rule is to apply a ‘side force’ like velocity vector towards agents travelling in a similar direction. Agents closer to each other should be more affected than those further away. This attractive rule will be denoted by $\mathcal{A}$ and is constructed as follows:

$$\begin{aligned}\mathbf{v}_i^{g,\mathcal{A}} &= \sum_j M_{ij}^x W_{ij}^\theta \mathbf{D}_{ij}^{SF} \\ &= \sum_j \left(\min \left[1, \frac{D_c}{x_{ij}} \right] \right) \left(\frac{1}{2} \left(\hat{\mathbf{v}}_i \cdot \hat{\mathbf{v}}_j + 1 \right) \right) \left(\left(\hat{\mathbf{v}}_i \times \hat{\mathbf{x}}_{ij} \right) \times \hat{\mathbf{v}}_i \right)\end{aligned} \tag{3.9}$$

External Velocity Rule Example

This rule is an externally commanded velocity rule, $\mathcal{E}$, where the external velocity $\mathbf{v}^{\mathcal{F}}$ applies to all agents. The magnitude of the external velocity uses the reference speed u_{ref}.

$$\mathbf{v}_i^{g,\mathcal{E}} = \mathbf{v}^{\mathcal{E}} \tag{3.10}$$

Discussion

It should be recognised that many variants are possible in the detailed implementation of agent behaviour and that as long as the basic structure of the rules is sound, some sort of interesting behaviour will be obtained. However, the guiding principles of the approach were adopted as follows.

- Normalisation of quantities at different points along the information processing chain must be considered carefully. The goal is to include and propagate magnitude information where it adds value to the signal chain but otherwise normalises. Normalisation is also important as part of good numerical housekeeping on the velocity goals of different rules such that rule weights are of a similar relative magnitude and behave in an intuitive manner.

- The use of a dimensionless approach is relatively straightforward to implement and is insightful because it provides a general, scalable solution to the problem.
- It is good practice to ensure that the rules are smooth in nature and that there are no step changes that can occur within the direction or magnitude of the rules. This is because step changes in goal velocities tend to result in unwanted jittery behaviours. For example, it would be bad practice to enforce a rule that would apply a constant goal velocity if an agent was travelling east and no goal velocity if the agent was travelling west, meaning a step change in rule magnitude between east and west.
- It is noted that there are different rule types depending on the combinations of rule magnitudes, directions and sums. For example, one type of rule applies a sum to an attribute of all neighbouring agents, in effect generating a population state, which is then acted on by a rule magnitude and/or direction. Another rule type computes rule magnitudes and directions for each neighbouring agent first and then performs a sum.

3.1.7 Rule Summation

To find the velocity goal of an agent, the velocity goals of all rules are multiplied by a gain factor and summed.

Let α be the list of all rules. Each rule $r \in \alpha$ has a gain K^r that is a scalar quantity that controls the weight of a given rule for all agents. The final summation of all rules is hence

$$v_i^g = \sum_{r \in \alpha} K^r \mathbf{v}_i^r \tag{3.11}$$

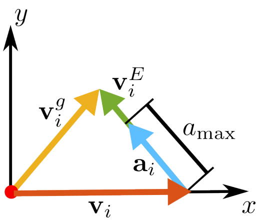


Figure 3.6. Acceleration applied to agent i in response to goal velocity $\mathbf{v}_i^g$

3.1.8 Rule Actuation

The method chosen here to actuate an agent uses a proportional controller to apply an acceleration proportional to the velocity error, $\mathbf{v}_i^E = \mathbf{v}_i^g - \mathbf{v}_i$ with a gain coefficient of acceleration K^a.The maximum acceleration limit is set based on a_{max}.

$$\mathbf{a}_i = \begin{cases} K^a \, \mathbf{v}_i^E, & \text{if } |K^a \, \mathbf{v}_i^E| \leq a_{\text{max}} \\ a_{\text{max}} \, \hat{\mathbf{v}}_i^E, & \text{otherwise} \end{cases} \tag{3.12}$$

This method was chosen as it is simple and makes use of the fact that agents can slow down and accelerate from a preferred speed. Another potential method is the side force method. This method converts the goal velocity into a side force that causes the agent to maintain a constant speed and turn with a given turn radius. This is more applicable to fixed-wing implementations.

A final comment on this process is on the rule computation interval. This is the interval at which a goal velocity is computed for an agent and is denoted as τ_r. A high interval results in infrequent recomputation and a higher risk of collision. A low interval requires significant computational power. The lower limit for τ_r is the timestep of the experiment. It is at the discretion of the designer to set τ_r appropriately. A reasonable starting point is the time it takes for an agent to travel a collision distance: $\tau_r = D_c/u_{\text{ref}}$.

3.2 Predefined Aerial Routing Networks

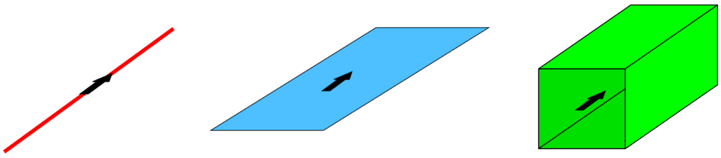
Predefined aerial routing networks are a type of structure that can be used to coordinate traffic within a UTM and are composed of paths and intersections. Paths are analogous to roads and railways in the sense that agents flow in the longitudinal direction and some control method or physical entity keeps the agents from drifting laterally. Intersections are where multiple paths intersect. The main difference between predefined aerial routing networks and road and rail is that predefined aerial routing networks have no physical structure and are simply rules that describe a 'virtual' structure for agents to follow.

In this section, predefined aerial routing networks are described from first principles, followed by a proposed method to implement them within a UTM. Following this, methods in which paths interact, allowing agents to pass from one path to another, are also proposed. This body of work builds on some principles outlined in previous literature, all of which are referenced; however, the remainder of this work is, to the authors' knowledge, unique and novel.

3.2.1 Basic Geometries of Path

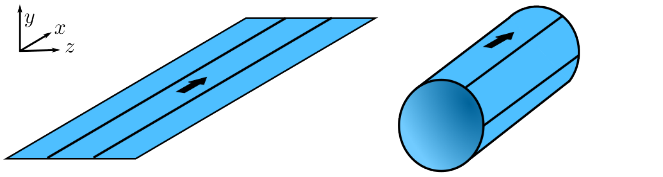
We define paths here as the following: an aerial navigation structure composed of a list of waypoints that join to form a centerline and a cross-sectional shape that describes its overall geometry. A common example of this is the concept of a corridor [1], [7], [52], [59].

Paths can be separated into three distinct types based on the number of dimensions in which they operate. Geometrically, a 1D path is a line, a 2D path is a ribbon, and a 3D path is a prism, however, to conform to previous nomenclature, a 1D path is called a *lane*, a 2D path is called a *strip* and a 3D path is called a *corridor*. All of these are defined schematically in Figure 3.7. The



(a) 1D - Lane (b) 2D - Strip (c) 3D - Corridor

Figure 3.7. Visual definitions of three types of paths with different numbers of dimensions.



(a) Unwrapped Surface (b) Wrapped Surface - Tube

Figure 3.8. Surface permutation (a) Simple surface with a fundamental direction and flow lines for context. (b) Wrapped surface with similar fundamental direction and flow lines. Agents behave similarly on both structures, travelling only on the surface and not 'within' the tube. The predominant topological difference between the two structures is the edge cases. When an agent reaches the edge of an unwrapped surface, the agent is constrained to stay on the surface where as on a tube, there are no edges and agents are able to travel around the centre of the tube freely.

choice of distinguishing between dimensions is due to the clear change in dynamics the agents will exhibit between these different structures. In a lane, an agent is able to simply follow a line. This means that agents are not able to overtake or travel at different speeds. In a strip, agents are able to travel along the principle direction of the path and in *one* of the orthogonal axis to the principle direction, that being along the surface of the ribbon. In this type of structure, agents are able to overtake each other and travel at different speeds. Strips do not need to be straight and flat, the surface can be transformed in different ways (for example the surface of a cylinder). This makes the traffic dynamics different to corridors. Finally, in a 3D path, agents are able to move in the principle direction and in *two* orthogonal directions. Similar to strips, agents are able to overtake and travel at different speeds in corridors. These are considered the fundamental types of paths.

Another permutation of the fundamental building blocks is the shape of the 2D path cross section. The strip can be any shape, including wrapped around itself to form the curved surface of a cylinder. This is a special type of strip that will be referred to here as a *tube*, Figure 3.8. Similar to a strip, agents only travel on the surface of the tube. A tube has a fundamental topological difference from an 'unwrapped' strip. When an agent reaches the edge of an unwrapped strip, it is constrained to stay within the geometry of the surface. However, in a tube there exist no edges; hence, agents are able to move freely around the surface.

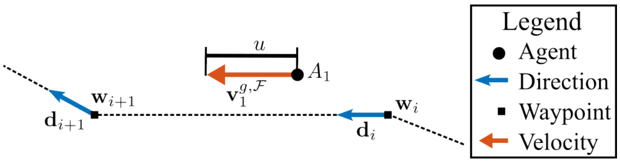


Figure 3.9. Definitions of flow rule nomenclature. This example shows part of a path that direct agents in the correct direction.

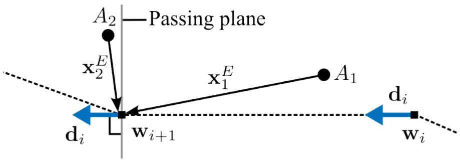


Figure 3.10. Visual definition of when an agent is considered to have 'passed' a waypoint. The direction vector, $\mathbf{d}_i$ has been superimposed on waypoint $\mathbf{w}_{i+1}$ to aid the visual comparison between the direction vector and the error vectors of agent i, $\mathbf{x}_i^E$. The grey line is the passing plane that is perpendicular to $\mathbf{d}_i$ and is coincident with $\mathbf{w}_{i+1}$.

3.2.2 Constructing Paths for Predefined Aerial Routing Networks

All paths are made up of two fundamental rules; the first rule is the rule that dictates the flow of the path; this will be referred to as the *flow rule*, and the second applies a force towards the geometry of the path; this will be referred to as the *geometry rule*.

Flow Rule

A path is defined by a list of waypoints Y, connected by a centerline. A waypoint, $\mathbf{w}_i$, is a position defined as $\mathbf{w}_i = [x^w, y^w, z^w]$. For a list of n waypoints, D is defined as the list of $n-1$ direction vectors. Each direction is defined as $\mathbf{d}_i = \mathbf{w}_{i+1} - \mathbf{w}_i$. For a basic path, we define the goal velocity of the flow rule as $\mathbf{v}^{g,\mathcal{F}}$ and is generally calculated using equation 3.13.

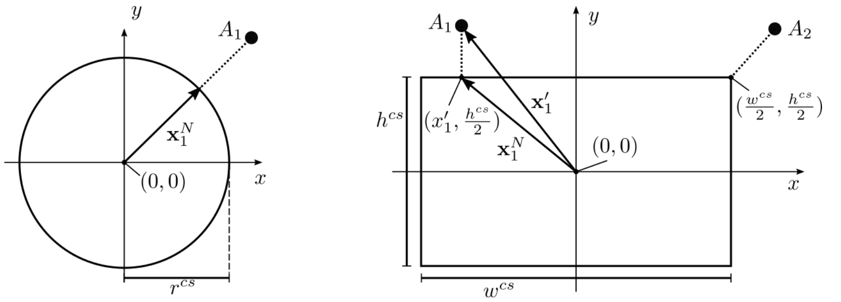
$$\mathbf{v}_i^{g,\mathcal{F}} = u_{\text{ref}}\,\hat{\mathbf{d}}_i \tag{3.13}$$

Here, u_{ref} is the reference speed of the agents. This is defined schematically in Figure 3.9.

It is important to note that $\mathbf{d}_i$ is *not* the direction from *agent* to *next waypoint*, as is commonly done, but from the *previous waypoint* to the *next waypoint*. In the general case, the flow rule is applied longitudinally along the path.

The method chosen here to define when an agent has 'passed' a waypoint uses the following process. Let there be a plane, called a *passing plane*, that is perpendicular to $\mathbf{d}_i$ and coincident with $\mathbf{w}_{i+1}$, shown as a grey line in Figure 3.10. An agent is considered to have passed the waypoint when, at a given timestep, it has passed through this plane. To find whether this has occurred, the error vector between the position of a given agent i and the next waypoint is found, $\mathbf{x}_i^E$. The dot product between this vector and $\mathbf{d}_i$ can be found, and the result will be negative when the agent has passed the plane. The reason why the dot product uses $\mathbf{d}_i$ and not $\mathbf{d}_{i+1}$ is that if there is a large difference between the directions of these vectors, agents will satisfy this condition very far from the waypoint. A simple thought experiment is that if $\mathbf{d}_{i+1}$ is anti-parallel to $\mathbf{d}_i$, all agents will satisfy this condition immediately. For this reason, $\mathbf{d}_i$ is used to define the passing plane. Once a waypoint is passed, the waypoint and path observed by an agent are updated to the next pair.

The rationale behind using this method, as apposed to the traditional ‘distance to a waypoint’ is the following. Firstly, in the traditional method, a radius is set, and when an agent enters the radius, the waypoint is updated. This means that an appropriate radius must be selected, in this case depending on the size of the path. This produces the possibility of agent A updating its waypoint before agent B, hence ‘pushing’ agent B out of the way, and hence causing agent B to miss the radius and continue on following the flow rule. The new method removes this problem and is also ‘cleaner’ as all agents change direction along the same plane.



(a) Circular-like Cross-section

(b) Rectangular-like Cross-section

Figure 3.11. Geometry rules schematic diagram for both types of path cross-section. Circular-like cross-sections are symmetric in the xy plane. Rectangular-like cross-sections are not symmetric and must be treated differently.

Geometry Rule

The geometry rule is a rule that ensures that agents remain within the cross-section of a path. Path cross-sections can be categorised into two types; those that are circular in nature and those that are rectangular in nature. Circular-like cross-sections are radially symmetric, and hence the force is always towards the centre line. Rectangular-like cross-sections have two components in the x and y directions. It can be useful to separate the geometry rule into the x and y components; however, these must be combined into one goal velocity to conform to all other geometry rules.

The goal velocity for the geometry rule is denoted as $\mathbf{v}_i^{g,\mathcal{G}}$. This velocity is applied from any direction orthogonal to the direction vector. To do this, it is useful to define a plane with origin along the centerline of the path that is orthogonal to the centerline. From this new frame of reference, the position of the ith agent is denoted as $\mathbf{x}_i'$. If the agent is outside the boundary of the cross-sections, the vector from the agent to the nearest point on the cross-section is required. To do this, the nearest point to the agent must be found. The method of doing this is different for a circular and rectangular-like cross-section, as shown in Figure 3.11.

For a circular-like cross-section, this is straightforward, as shown in Figure 3.11 (a). The equation to find the vector to the nearest point is shown in Equation 3.14.

$$\mathbf{x}_i^N = r^{cs}\hat{\mathbf{x}}_i' \tag{3.14}$$

where $\mathbf{x}_i^N$ is the vector to the nearest point on the cross section of agent i and r^{cs} is the radius of the path cross-section. For a lane, this is zero.

For a simple rectangular cross-section, $\mathbf{x}_i^N$ can be found using Equation 3.15. The sign function sgn is used to ensure that $\mathbf{x}_i^N$ is in the correct direction.

$$\mathbf{x}_i^N = \begin{cases} \left(\text{sgn}(x_i')\frac{w^{cs}}{2},\ \text{sgn}(y_i')\frac{h^{cs}}{2}\right), & \text{if } |x_i'| > \frac{w^{cs}}{2} \text{ and } |y_i'| > \frac{h^{cs}}{2} \\ \left(\text{sgn}(x_i')\frac{w^{cs}}{2},\ y_i'\right), & \text{if } |x_i'| > \frac{w^{cs}}{2} \text{ and } |y_i'| \le \frac{h^{cs}}{2} \\ \left(x_i',\ \text{sgn}(y_i')\frac{h^{cs}}{2}\right), & \text{if } |x_i'| \le \frac{w^{cs}}{2} \text{ and } |y_i'| > \frac{h^{cs}}{2} \end{cases} \tag{3.15}$$

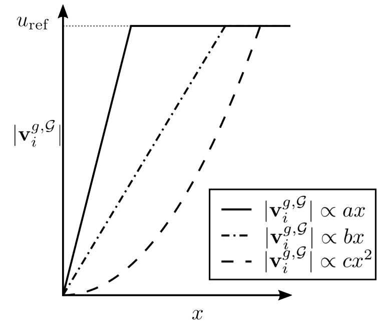


Figure 3.12. Example Magnitude shapes that can be used in the geometry rule. For this work, a linear function is used with the gradient $m = 1/D_c$.

Once $\mathbf{x}_i^N$ is found, the direction of the geometry rule is simply the following.

$$\hat{\mathbf{v}}_i^{g,\mathcal{G}} = \frac{\mathbf{x}_i^N - \mathbf{x}_i'}{|\mathbf{x}_i^N - \mathbf{x}_i'|} \tag{3.16}$$

The final part is to design the 'shape' of the magnitude of the geometry rule. As described previously, it is good practice to ensure that the edges of paths do not act like 'solid' walls by having a step increase in the goal velocity. Step changes in goal velocities tend to result in unwanted jittery behaviours. The shape of the magnitude is the choice of the designer and is not an exact science. The requirements are that agents should not be able to diverge from the path by more than the size of an agent but should not be too similar to a step function. For the present work, this distance was set to the collision distance of an agent. There is no requirement on the shape of the rule magnitude; for example, the shape can be linear or non-linear. In this body of work, a linear function has been shown to be fit for purpose and is more simple than a non-linear function. Examples of linear and non-linear functions are shown in Figure 3.12.

Once the goal velocity for the geometry rule is computed, it is converted from path axes to earth axes. This process is a simple rotation matrix and is described in full in Appendix 8.2.6.

A schematic overview of the direction and magnitude of the geometry rule is presented for the three forms of path in Figure 3.13.

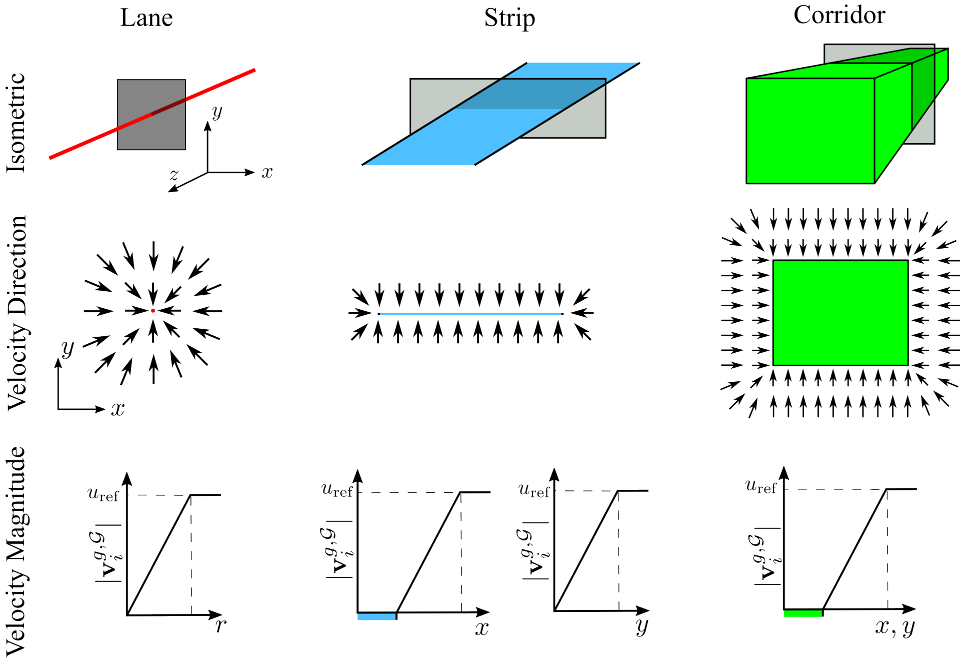


Figure 3.13. Definition of direction and magnitude of geometry rule for standard 3 types of paths. The top row is a visual aid for understanding the cross-section seen in the velocity direction row. It should be noted that for the rectangular-like paths, the geometry rule is not towards the centerline but the nearest point on the path. The velocity magnitude row shows the magnitude of the geometry rule as an agent leaves the boundary. Pay close attention to the 'x'-axis on each as they are all different and refer to each unique type of path.

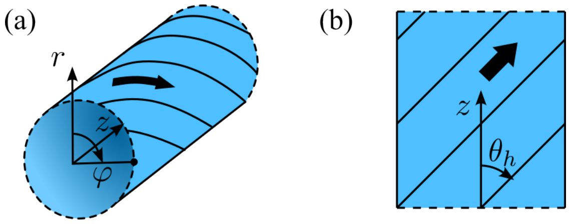


Figure 3.14. Defining the helical path of agents on a tube. (a) Isometric view showing agent path around the center of the tube. (b) Top view showing the helix angle θ_h.

Helical Flow Rule

A tube may have a component of the flow rule that is tangential to the surface. This causes the paths of the agents to form helices. The angle at which the helical path crosses the longitudinal axis is termed the helix angle, θ_h. This is achieved by defining the parallel component, $v_{\parallel}^{g,\mathcal{F}}$, and the tangential component, $v_{\perp}^{g,\mathcal{F}}$.

$$
\begin{aligned}
\mathbf{v}_{\parallel}^{g,\mathcal{F}} &= u_{\text{ref}}\cos(\theta_h)\hat{\mathbf{d}}_i \\
\mathbf{v}_{\perp}^{g,\mathcal{F}} &= u_{\text{ref}}\sin(\theta_h)(\hat{\mathbf{x}}_i^N \times \hat{\mathbf{d}}_i) \\
\therefore \mathbf{v}_i^{g,\mathcal{F}} &= \mathbf{v}_{\parallel}^{g,\mathcal{F}} + \mathbf{v}_{\perp}^{g,\mathcal{F}}
\end{aligned}
\tag{3.17}
$$

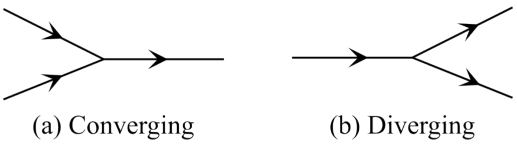
3.2.3 Constructing Intersections for Predefined Aerial Routing Networks

In this work, intersections are defined as two or more paths intersecting each other. There are many examples of intersections in other traffic systems. In rail, most intersections are simple converging and diverging junctions (colloquially called Y junctions) and some crossing intersections (colloquially known as X crossings). In road traffic, there are many more types of intersections between roads, from simple T junctions to roundabouts.

In the following sections, intersections in aerial routing networks are examined. The key contribution of this section is the identification of several intersection axioms.

Defining Junctions

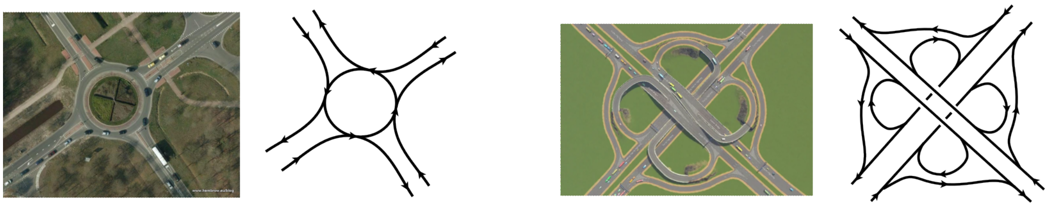
As previously defined, paths are constructed from a list of waypoints. An intersection is defined as a point where two or more paths converge, diverge, or cross. Within the body of work presented here, it is clear that a cross-intersection would not yield favourable results, and hence it is not considered. Therefore, the term *junction* is referred to here as a converging or diverging junction, presented in Figure 3.15.



(a) Converging (b) Diverging

Figure 3.15. The fundamental junctions considered in this body of work. Converging junctions have two paths join to form one path. Diverging junctions have a single path split into two paths. Although the junctions are not dissimilar, considerable differences in the behaviour of the agents are observed.

When considering other traffic networks, it can be seen that generally, most intersections that involve more than 3 paths are a series of converging and diverging junctions. For example, roundabouts and clover-leaf interchanges can be simplified to several junctions, as seen in Figure 3.16. In this work, the term *interchange* refers to a series of junctions used to travel between several different paths.



(a) Roundabout

(b) Cloverleaf Interchange

Figure 3.16. Examples of large intersections simplified into a combinations of converging and diverging junctions. On the left of each is a birds eye view of a road interaction and on the right is simplification of the paths. It is easy to see both examples are different combinations of junctions.

Parametrisation of Junctions

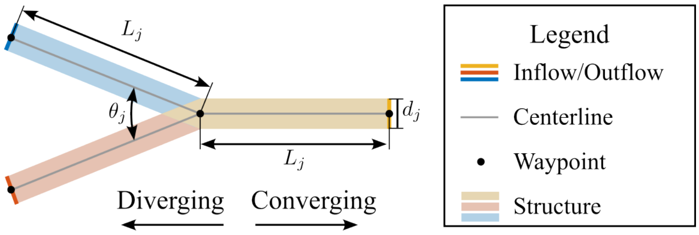
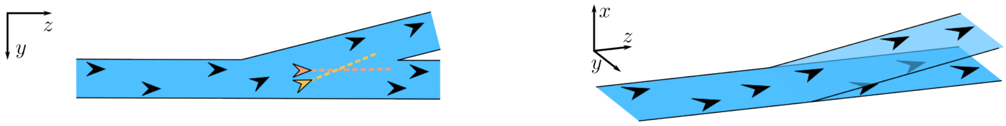


Figure 3.17. Parametrisation of a junction. The flow is left to right for a converging junction, and right to left for a diverging junction. The use of orange and blue are used to represent regions that are orange and blue agents specific. The use of yellow is for shared regions.

The main components of a junction are presented in Figure 3.17. The definitions of each of these parameters are in Table 3.1.

Table 3.1. Junction parameter definitions

Symbol	Unit	Definition
L_j	m	Distance between inflow/outflow to the intersecting point
θ_j	°	Junction angle between inflowing/outflowing paths
d_j	m	Characteristic size of structure



(a) In-plane Diverging

(b) Out of Plane Diverging

Figure 3.18. Two special cases of ribbons in a diverging junction, (a) In-plane divergence, $\varphi = 0$, where it is noted that agents will cross each other paths during a divergence, as seen by the yellow and orange agents and predicted paths (b) Out-of-plane divergence, $\varphi = 90°$, where agents do not cross paths during a divergence.

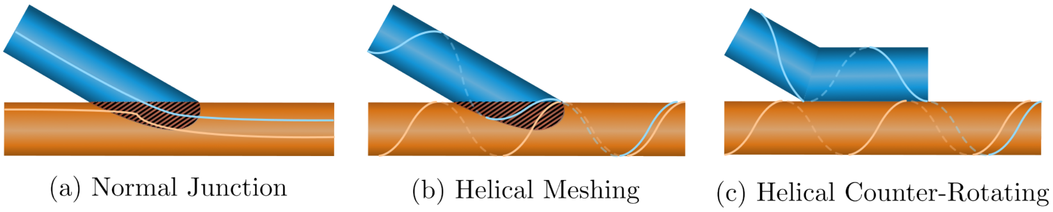

(a) Normal Junction (b) Helical Meshing (c) Helical Counter-Rotating

Figure 3.19. Tube junctions with flow and geometry rules. Red striped region shows areas with high closing velocities between agents. (a) Normal junction with no additional rules. (b) Helical meshing junction with $\theta_h \neq 0$. (c) Helical counter-rotating junction with $\theta_h \neq 0$. Fundamental structure is changed, hence a new crossing rule is required.

A parameter for strips, φ, defines the angle of the strip around the centerline. Figure 3.18 shows two edge cases, where $\varphi = 0$ for strips diverging in-plane and $\varphi = 90^\circ$ for ribbons diverging out-of-plane.

Parametrisation of Helical Tube Junctions

When two tubes converge, with the standard flow rule, as shown in Figure 3.19 (a), agent behaviour is similar to a corridor convergence junction. Areas of disruption are shown in the red-striped region. If $\theta_h \neq 0$, there is still a similar area of disruption, as shown in Figure 3.19 (b). An interesting solution to this problem is to ensure that the two tubes are counter-rotating when arriving at the junction. This can be done by reversing the helix angle when agents switch from one tube to the other. The structure of this junction is slightly different from the previous examples. In a counter-rotating junction, the longitudinal axis of both paths become parallel at a distance where the surfaces are coincident. A new condition is introduced, in which, when the agents pass the point where the two tubes are coincident, the geometry rule changes to the new paths, and the agents will begin orbiting the new centerline. One of the benefits of this junction is that since the tubes are symmetric around their longitudinal axis, there can be multiple tubes converging at the same time from different directions. For the remainder of this thesis, the counter-rotating helical tube junction is the only junction considered for helical tubes.

The parameterisation of these tubes is shown in Figure 3.20. The only new parameter for a helical junction is the switching distance, L_s. This is the distance in which the agents switch from one tube to the other. There must be a full orbit of the terminating tube to allow all agents to switch, therefore, there is a minimum allowable value $L_{s,\text{min}}$. The value of $L_{s,\text{min}}$ depends on the size of the tube and the helix angle and can be found analytically by ‘unwrapping’ tube and measuring the length of one orbit.

$$L_{s,\text{min}} = \frac{\pi d_j}{\tan \theta_h} \tag{3.18}$$

The motion found in the counter-rotating helical tube junctions is somewhat analogous to the motion of helical gears, where the paths of the agents follow the teeth of the gear as they move longitudinally along the shaft, as illustrated in Figure 3.21. There are several other gear analogies that can be used to describe novel tube interactions, discussed in Chapter 7.

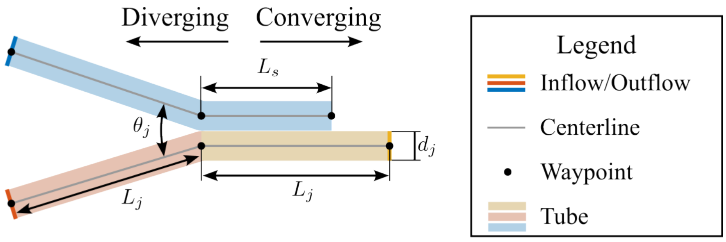


Figure 3.20. Parameterisation of helical tube counter-rotating junctions.

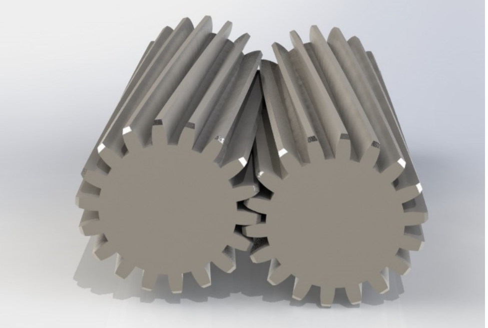


Figure 3.21. Helical gears meshing, similar to counter-rotating tubes where agents follow the path of the teeth.

3.3 Emergent Aerial Routing Networks

3.3.1 Introduction

Emergent aerial routing networks are the formation of paths within environments with no predefined path topologies. This phenomenon is found in many biological systems, from humans walking across busy intersections to herds of sheep in a field. In Figure 3.22, emergent lanes are seen forming in human simulations in a contra flow test case.

In this section, the free-flight model will be presented first. Following this, the design and implementation process for the migration rule and the procession rule is outlined.

3.3.2 Free Flight Model

Free flight is a method of air traffic control in which there is no centralised control. The defining feature of free flight is that agents can generally fly the shortest route from departure to arrival, making the route more efficient. There are many interpretations of this. In commercial manned aviation, a similar method called *Free Route* is being examined and implemented in some

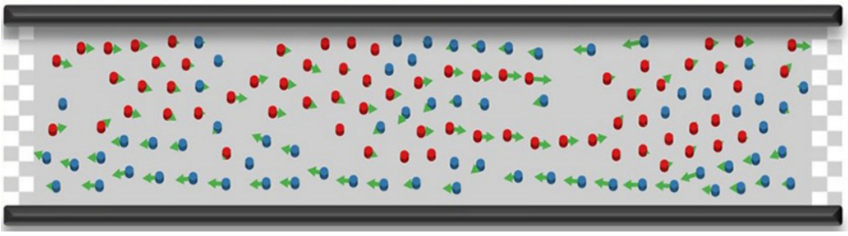


Figure 3.22. Emergent lane formation found in simulation of humans in a contra flow test case. (Source: [101])

small regions [102], [103]. In this work, we refer to free flight in a light similar to that in most of the aerial robotic literature. In this implementation, agents use the shortest route and communicate with/use cameras to see other agents to ensure deconfliction [1], [7]. Free flight is practically implemented here with a migration and separation rule, causing agents to travel to their destinations while avoiding other agents.

3.3.3 Migration Rule

In all emergent topologies, a destination-seeking rule is used to apply a velocity toward the destination of an agent. The rule used here is analogous to the migration rule used in [77] and will be referred to as such. The migration rule is simply the following:

$$\mathbf{v}_i^{g,\mathcal{M}} = u_{\text{ref}}\hat{\mathbf{x}}_i^E \tag{3.19}$$

Here, $\mathbf{x}_i^E$ is the error vector from agent i to the destination of the agent. Destinations can be a point, such as a waypoint, or a plane. The error vector in the case of a plane is simply to the nearest point on the plane.

3.3.4 Procession Rule

The procession rule attracts agents travelling in similar directions to form paths. There are many ways of constructing a rule to generate this effect. In this study, we consider the three components used to construct this rule. These are (1) the rule magnitude based on distance between agents, (2) the rule magnitude based on the direction of travel between agents and (3) the a rule direction towards agents travelling in similar directions. Within each of these magnitudes and directions, several methods are proposed. It is noted that this is not an exhaustive list, simply some useful methods.

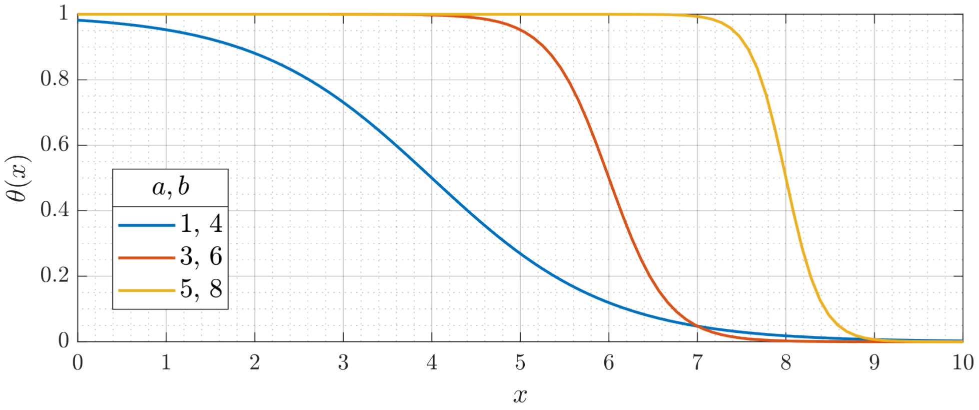


Figure 3.23. Sigmoid operator used for rule magnitude based on distance in the procession rule.

Rule Magnitude Based on Distance

This rule magnitude changes based on the proximity between agents. The requirement of this magnitude is that it has an output range between 0 and 1, and it causes agents within a range to coalesce to form a path. It is also best practice to use a continuous function. A sigmoid function was chosen as it meets the requirements and has two useful variables that change the shape of the function in a useful way. The equation for the sigmoid function is as follows.

$$M_{ij}^{\sigma} = \frac{1}{1 + e^{a(x_{ij}-b)}} \tag{3.20}$$

Where a is the gradient at the change of inflection and b is the position on the x-axis where the change of inflection is and is simultaneously the point at which the output of the function is 0.5. Graphical examples are presented in Figure 3.23.

Rule Magnitude Based on Velocity

This rule magnitude changes according to the direction in which the agents travel with respect to each other. The requirements of this rule magnitude are that agents travelling in similar directions should be attracted to each other and agents travelling in opposite directions should be repelled. Similarly to the previous rule, the rule magnitude should have a maximum of 1 but can have a minimum of -1. There are two methods to achieve this, using *velocity direction* and *destination error*. The velocity direction method compares the unit vectors of the agent's velocities. The destination error method compares the destination error unit vector between agents, identical to the error used in the magnitude rule. The equations for both methods are as follows:

$$M_{ij}^{\theta} = \hat{\mathbf{v}}_i \cdot \hat{\mathbf{v}}_j \quad \text{For Velocity Direction} \tag{3.21}$$

$$M_{ij}^{\theta} = \widehat{\mathbf{x}_i^E} \cdot \widehat{\mathbf{x}_j^E} \quad \text{For Destination Directions} \tag{3.22}$$

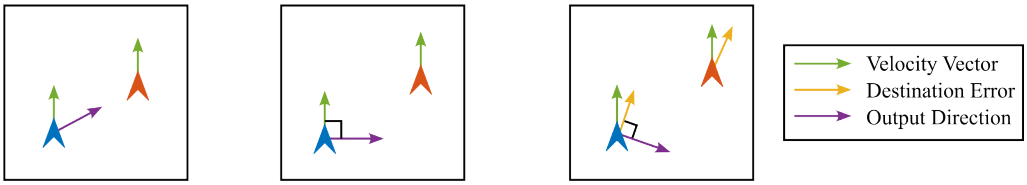
Here, $\mathbf{x}_i^E$ is the error vector from agent i to the destination of the agent. The result of either of these weightings is 1 for parallel vectors and -1 for anti-parallel vectors.

Rule Direction Based on Relative Position

The rule direction has three possible methods. The first is directed toward or away from other agents. The second is orthogonal to the velocity vector and towards the neighbouring agent. The last is orthogonal to the destination error vector and towards the neighbouring agent. See Figure 3.24 for diagrams. The following equations with cross-products have been written in this way so that readers can use the right-hand rule to help understand; however, the equations can be made more computationally efficient by using laws of triple products.

$$\hat{\mathbf{v}}_i^{g,\mathcal{P}} = \hat{\mathbf{x}}_{ij} \quad \text{For Attractive} \tag{3.23}$$

$$\hat{\mathbf{v}}_i^{g,\mathcal{P}} = (\hat{\mathbf{x}}_{ij} \times \hat{\mathbf{v}}_i) \times \hat{\mathbf{v}}_i \quad \text{For Orthogonal Velocity} \tag{3.24}$$

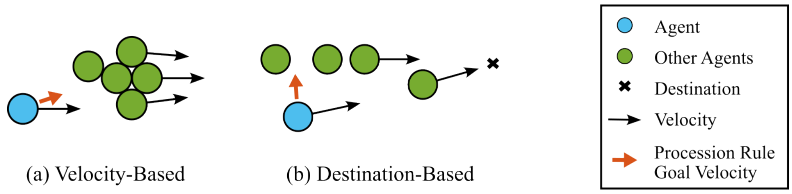


(a) Directly toward neighbour

(b) Towards neighbour + Orthogonal to velocity

(c) Towards neighbour + Orthogonal to destination error

Figure 3.24. Diagrams of different methods used to calculate the output direction for the procession rule. Output direction is only computed for left agent in all cases.



(a) Velocity-Based

(b) Destination-Based

Figure 3.25. Sketches of simulation results from simple experiments of two implementations chosen to further examine in this thesis. Both of these sketches will be used in future diagrams to aid the reader recall the two implementations.

$$\hat{\mathbf{v}}_i^{g,\mathcal{P}} = (\hat{\mathbf{x}}_{ij} \times \hat{\mathbf{x}}_i^e) \times \hat{\mathbf{x}}_i^e \quad \text{For Orthogonal Destination} \tag{3.25}$$

Procession Rule Implementations

Using these components, there are many possible implementations. Within this thesis, two implementations were selected for further examination. The first is the simplest to compute and the second is the method that performed the best in the simple experiments. The implementation that is the simplest to compute uses the **sigmoid** distance magnitude, the **velocity** direction magnitude, and the **directly towards agent** direction method. This implementation will be coined the *velocity-based* implementation. The equation for the rule is as follows:

$$\mathbf{v}_i^{g,\mathcal{P}} = \sum_j \frac{(\hat{\mathbf{v}}_i \cdot \hat{\mathbf{v}}_j)\hat{\mathbf{x}}_{ij}}{1 + e^{a(x_{ij}-b)}} \tag{3.26}$$

The implementation that performed the best in simple experimentation also used the **sigmoid** distance magnitude, the **destination** error magnitude and the **towards neighbour and orthogonal to destination error** direction. This implementation is referred to as the *destination-based* implementation. The equation for the rule is as follows:

$$\mathbf{v}_i^{g,\mathcal{P}} = \sum_j \frac{(\widehat{\mathbf{x}_i^e} \cdot \widehat{\mathbf{x}_j^e})(\hat{\mathbf{x}}_{ij} \times \hat{\mathbf{x}}_i^e) \times \hat{\mathbf{x}}_i^e}{1 + e^{a(x_{ij}-b)}} \tag{3.27}$$

Within simple experiments, some observations can be made about these two implementations. Figure 3.25 shows a sketch of the results of each implementation. Figure 3.25 (a) Shows the clumping effects found in velocity-based methods as agents are attracted towards agents travelling with similar velocity vectors. Although this clumping effect is not desirable and may act to reduce the performance of the system, it is investigated further in this thesis because of its simplicity. Figure 3.25 (b) Shows paths formed in the destination-based implementation due to the side forces acting toward a path centerline. These sketches are used throughout this thesis to aid the reader in recalling the implementations.

3.4 Collision Avoidance Methods in Distributed Control UTM

3.4.1 Introduction

In distributed control UTMs, collision avoidance methods are a necessity. In this work, two collision avoidance methods that use rule-based algorithms are presented. The first, simple separation (also known simply as the separation rule) is a close-range repulsion rule that is used to avoid collisions with other agents. It is computationally simple and has been used in a wide range of applications since its inception in [104], namely [77]. The second is a novel rule referred to as orthogonal separation. This rule causes agents to avoid collision by moving orthogonal to the plane of motion. For example, if the agents are moving in the horizontal plane, they will avoid collision by accelerating on the vertical axis. This is similar to the operating procedure of traffic collision avoidance system (TCAS) in manned aviation[2].

3.4.2 Simple Separation Rule

The simple separation rule (from now on referred to simply as the separation rule) computes a repulsive velocity between agents, which is found by summing multiple contributions of neighbouring agents into a total goal velocity. To explain the separation rule, consider an environment with a main agent for which the separation rule will be computed, with three neighbouring agents, numbered 1 to 3. Agent 1 is outside the conflict distance of the main agent, and hence there is no contribution to the separation velocity. Agent 2 is inside the conflict distance, and hence is being actively deconflicted by applying a repulsive contribution to the main agent proportional to the distance between the agents. Agent 3 is also within the conflict

[2] Traffic Collision Avoidance System (TCAS) - https://en.wikipedia.org/wiki/Traffic_collision_avoidance_system

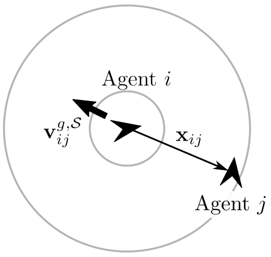


Figure 3.26. Resultant velocity goal of the separation rule due to agent j on agent i in the anti-parallel direction of $\mathbf{x}_{ij}$.

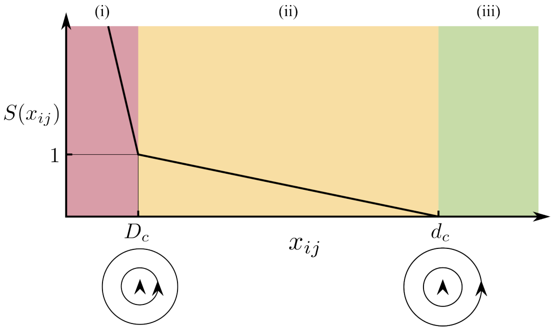


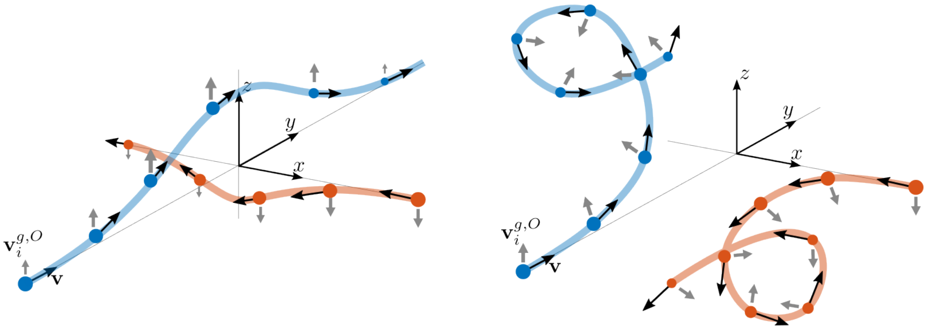
Figure 3.27. Magnitude of separation rule based on proximity to neighbouring agent. Three regimes of the separation magnitude, (i) critical collision event, (ii) deconfliction, (iii) no separation.

distance and hence also contributes another repulsion force. The sum of the repulsion force of agents 2 and 3 is calculated and applied to the main agent.

The direction of the separation rule is always away from the neighbouring agent. This is represented visually in Figure 3.26. In this implementation, the magnitude of the rule varies with proximity between agents, as seen in Figure 3.27. An interactive graph is found here [3].

This method of deconfliction has no prioritisation. This means that any two agents in conflict will experience equal and opposite repulsive goal velocities. Similarly, this means that there is no ‘decision making’ protocol about which agent will perform an evasive manoeuvre. Due to this, communication between the agents is vitally important to enable the rule-based algorithm to generate a repulsive velocity effectively. This process is very similar to the T-CAS system described in Section 2.4.3. T-CAS is a system that identifies a potential collision and based on a simple algorithm, one aircraft is commanded to climb and the other to descend. In this work, a similar method is prescribed, however, a portion of an agents commanded to move in the direction directly away from the conflicting agent instead of directly up or down.

[3] Interactive Graph of Separation Magnitude - https://www.desmos.com/calculator/vpk2fnkyaz



(a) Ideal

(b) Orthogonal Velocity Output Direciton

Figure 3.28. Orthogonal Separation rule trajectories. (a) The ideal orthogonal separation trajectory. (b) The trajectory if orthogonal separation rule direction is velocity based.

The complete method to calculate an agent's goal velocity i is found using Equation 3.28.

$$\mathbf{v}_i^{g,\mathcal{SS}} = \sum_j u_{\text{ref}}\, S(x_{ij})\, \hat{\mathbf{x}}_{ji} \tag{3.28}$$

Where u_{ref} is the reference velocity of the agent, x_{ij} is the magnitude of the vector from agent i to agent j and $S(..)$ is the normalised magnitude of the velocity command. The shape of the function, $S(x)$, increases linearly to 1 as the distance between agents decreases. If the distance between agents falls below the collision distance, meaning, they have collided, the gradient of the $S(x)$ function increases to a value k. This can be considered as the stiffness of the material within the collision distance. This function is defined formally in Equation 3.29 and visualised in Figure 3.27.

$$S(x_{ij}) = \begin{cases} \frac{k(D_c - x_{ij})}{D_c} + 1, & \text{if } x_{ij} \in [0, D_c) \\ \frac{d_c - x_{ij}}{d_c - D_c}, & \text{if } x_{ij} \in [D_c,\, d_c) \\ 0, & \text{otherwise} \end{cases} \tag{3.29}$$

It is possible to normalise the collision and conflict distance using the characteristic length of agents. A good starting point for a low collision risk is $D_c = l_c$ and $d_c = 10l_c$. The same can be applied to k, and a similar conservative starting point is $k = 100l_c$.

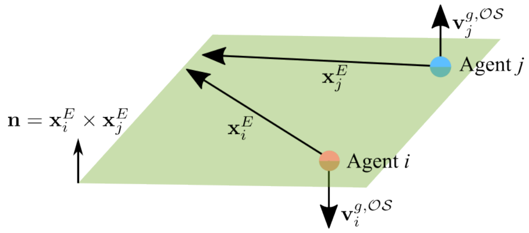

$\mathbf{v}_j^{g,\mathcal{OS}}$
Agent j
$\mathbf{x}_j^E$
$\mathbf{n} = \mathbf{x}_i^E \times \mathbf{x}_j^E$
$\mathbf{x}_i^E$
Agent i
$\mathbf{v}_i^{g,\mathcal{OS}}$

Figure 3.29. Diagram of orthogonal separation rule direction using destination error. Agents traverse on the plane of motion (green) and each deviate in the direction of the goal velocity to avoid collision.

3.4.3 Orthogonal Separation Rule

Orthogonal separation is a rule that makes use of the third dimension available to aerial vehicles to avoid collisions. This is in contrast to the separation rule which applies a repulsive goal velocity in the same plane of motion as the agents are travelling. The aim is that agents should deviate from their path as little as possible while avoiding collisions. Figure 3.28 (a) shows an ideal deconfliction case.

To achieve this ideal deconfliction, the following requirements are established. The rule magnitude must increase with decreasing proximity to other agents. The method used in the separation rule is fit for purpose here. The rule magnitude must also increases as closing velocity increases. To achieve this, a rule magnitude is chosen that increases linearly with closing velocity, $M_{ij}^v = \frac{v_{ij}}{2u_{\text{ref}}}$.

The rule direction must cause agents to avoid collision with a minimum deviation from their path. It is also required that the rule be simple to implement with low computation costs. There are two possible methods that can be used. The most intuitive is to use the cross product of the velocity vectors; however, this causes a runaway effect where agents spiral away from each other; see Figure 3.28 (b). Due to this, the destination error is used as this does not change considerably as the agents execute the deconfliction manoeuvre. It can be useful to visualise this rule as two agents traversing a plane of motion. The plane of motion is defined as the plane that is orthogonal to both agent destination errors and coincides with the midpoint of $\mathbf{x}_{ij}$. This rule direction applies a goal velocity away from the plane of motion, in opposite directions for each agent, Figure 3.29.

The formal definition of orthogonal separation combines all these requirements. The rule direction is found from the cross product between the destination error vectors, the rule magnitude uses the same function as the separation rule, and the closing velocity magnitude, M_{ij}^{v}.

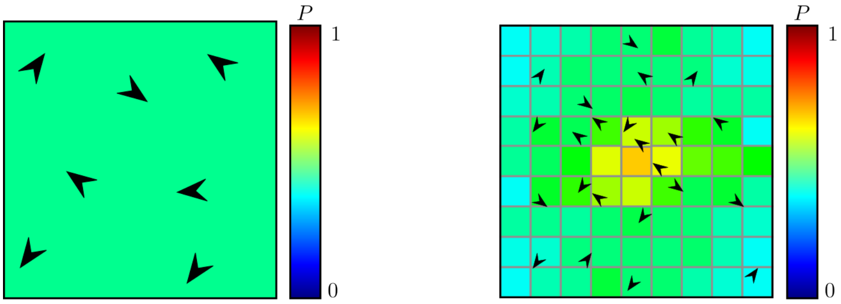
$$\mathbf{v}^{g,\mathcal{OS}} = u_{\text{ref}} \sum_{j} S(x_{ij}) \cdot \frac{v_{ij}}{2u_{\text{ref}}} \cdot (\hat{\mathbf{x}}_i^E \times \hat{\mathbf{x}}_j^E) \tag{3.30}$$

3.5 Measuring Congestion in MAS

3.5.1 Introduction

The measure of congestion is defined here as an approximation to the probability of collision between agents in a certain region of space. In this section, several methods are explored to measure this. The first method uses a quantity defined in traffic theory as traffic pressure. The second method uses a gas kinematics approach in which the agents are assumed to be analogous to gas particles. Although traffic pressure is generally not used to predict the probability of collision, it is hypothesised that there is a relationship between traffic pressure and the probability of collision. These studies will be carried out on a small scale to identify the relationship with various types of flows, as demonstrated in Figure 3.30 (a).

Following this, a brief description of how to divide an environment into cells to compare congestion in different regions is presented, similar to that seen in Figure 3.30 (b).



(a) Single cell

(b) Multi cell

Figure 3.30. Measuring congestion in different environments. The probability of collision, P, is a proxy for the congestion metric. (a) Measuring congestion in a single cell. (b) Measuring congestion in a large environment divided into smaller cells.

3.5.2 Methods for Quantifying Congestion in a Single Cell

Using Traffic Pressure

Traffic pressure is mathematically defined in [27] as

$$P = \rho\Theta \tag{3.31}$$

Here, ρ is density and θ is velocity variance. Density is defined in several ways in traffic theory and gas kinematics. In this work, density is generally defined as

$$\rho = N/V \tag{3.32}$$

Here, V is the volume and V is the volume. In some future experiments, the density will be defined differently to be appropriate for the given context.

The velocity variance is mathematically defined in [27] as

$$\Theta = \langle v_i^2 \rangle_i - \langle v_i \rangle_i^2 \tag{3.33}$$

Where $\langle .. \rangle_i$ is the mean operator equivalent to $\langle x_i \rangle_i = \sum_i x_i/N$ and v_i is the velocity of agent i.

This equation was originally derived to find the velocity variance of cars that drive along a road where all agents have the same path. Θ should really be called velocity *magnitude* variance. Considering the direction of agents also presents a challenge, especially if agents are travelling in three dimensions. In this work, *many* methods have been explored to quantify the velocity variance in these environments. In this thesis, only the methods that have presented a substantially positive correlation with the collision rate will be described. These methods are velocity direction variance and generalised variance.

Velocity direction variance is the simplest and assumes that all agents are travelling at the same speed; it is simply the direction that differs. This simplifies the velocity variance term into a single degree of freedom, similar to the velocity variance used in traffic theory, using direction instead of speed. Velocity direction variance, Θ_d, is defined as

$$\Theta_d = \langle \theta_i^2 \rangle_i - \langle \theta_i \rangle_i^2 \tag{3.34}$$

Here, θ_i is the direction in which an agent travels with respect to an arbitrary direction, for example, from the x-axis.

Generalised variance considers at least two degrees of freedom. This is beneficial as it does not assume that the agents are travelling at the same speed. Here, a description of the method for finding the generalised variance for a 3D environment is outlined. The same process can be used for a 2D environment, with minor changes.

Generalised variance is a method to find variance in multivariate data and is formally defined as the determinant of the covariance matrix [105]. In this case, the covariance matrix was constructed from the components of the velocity vectors of each agent. Generalised variance should be used with caution when using different measures. For example, if the covariance matrix is constructed using a magnitude and a direction, with units ms^{-1} and $^{\circ}$, respectively, this would give inconsistent results. However, in this work, the covariance matrix is constructed based on the xyz components of the velocity vector and therefore is a valid use case.

Recall that the velocity of an agent is defined as

$$\mathbf{v} = [\dot{x},\ \dot{y},\ \dot{z}]^T \tag{3.35}$$

There exists a differing nomenclature for denoting the covariance matrix. Here, the nomenclature is consistent with [106]. The covariance matrix of the velocities of agents $\mathbf{K_{vv}}$ in a 3D environment is a 3x3 square matrix and is found using the following.

$$\mathbf{K_{vv}} = \text{cov}(\mathbf{v}, \mathbf{v}) = E\left[(\mathbf{v} - E[\mathbf{v}])(\mathbf{v} - E[\mathbf{v}])^T\right] \tag{3.36}$$

Here, E denotes the expected value, in this case the mean.

To find the generalised variance Θ_g, the determinant of this matrix is found.

$$\Theta_g = \det\left(\mathbf{K_{vv}}\right) \tag{3.37}$$

This quantity can be substituted into the standard traffic pressure equation.

Using Gas Kinematics

The expected collision rate is a method to predict the probability of collision in gas kinematics. Here, a derivation from the mean free path is included for context.

The fundamental mean free-path equation with the $\sqrt{2}$ correction factor is [107], [108]

$$\lambda = \frac{1}{\sqrt{2}\pi D_c^2 \rho} \tag{3.38}$$

Where λ is the mean free path of a particle, D_c is its collision distance of an agent and ρ is the number density. To find the mean free time, we use the following.

$$\begin{aligned} \bar{v} &= \frac{\lambda}{\tau} \\ \therefore \tau &= \frac{\lambda}{\bar{v}} \\ &= \frac{1}{\bar{v}\sqrt{2}\pi D_c^2 \rho} \\ \therefore Z &= \bar{v}\sqrt{2}\pi D_c^2 \rho \end{aligned} \tag{3.39}$$

Where τ is the mean free time, $\bar{v}$ is the mean speed of the particles, and Z is the frequency of collision of a given particle. To find the overall frequency of collisions, we multiply by the number of particles and divide by half. We are only considering collisions between 2 particles at a time, as the probability of 3 or more particles colliding simultaneously is vanishingly small in comparison to the probability of 2.

$$z = \frac{1}{2}NZ = \frac{1}{2}N\bar{v}\sqrt{2}\pi D_c^2 \rho \tag{3.40}$$

We can see here that the collision frequency is a function of the average speed of the agents, the number of agents, the number density of agents, and the size of the agent. Recall that this equation assumes a uniformly random motion. If all agents are travelling in the same direction at the same speed, this would result in no collisions, and this equation does not hold. It is proposed that if a drift velocity is present in the sample, it can be subtracted. This is analogous to making the drift velocity the new frame of reference.

Recall that $\bar{v}$ is the mean *speed*. It is proposed that the drift (mean) velocity is found and subtracted from all velocities, and the magnitude of this will become the new mean speed.

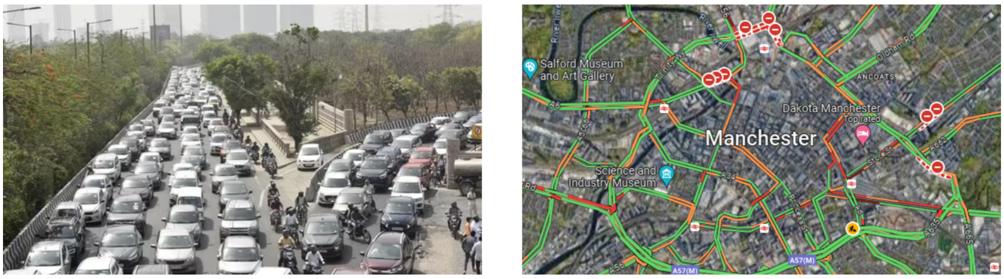
$$\bar{v} = \langle |\mathbf{v}_i - \mathbf{v}_{drift}| \rangle_i \tag{3.41}$$

This is substituted into 3.40 to give the following:

$$z = \frac{1}{2}NZ = \frac{1}{2}N\langle |\mathbf{v}_i - \mathbf{v}_{drift}| \rangle_i \sqrt{2}\pi D_c^2 \rho \tag{3.42}$$

3.5.3 Methods for Quantifying Congestion in Multiple Cells

The aim of this work is to be able to identify congestion in large environments for the purpose of redesigning navigation structures to increase efficiency and safety. In a road traffic network, congestion is usually found when there exists an increase in density, that results in a reduction in speed and therefore trip time, Figure 3.31 a). Congestion in a road traffic network is defined by agent speed compared to the normal speed on a section of road. This can then be visually displayed using a heat map, as shown in Figure 3.31 b). In this section, the concept of congestion within a UTM is discussed and how to measure and use it to optimise the design of a aerial navigation structure. The definition for congestion in road traffic can be repurposed for a UTM, however, this definition is only useful for increasing efficiency. In UTMs, it is also important to increase the factor of safety as much as possible to reduce the risk of a potential collision. For this reason, congestion should be a metric that considers efficiency and risk of collision.



(a) Road Traffic Congestion

(b) Traffic Congestion Heatmap

Figure 3.31. Road congestion with heat map.

Next, to construct a heat map of congestion, the environment must be divided into cells. There exists a trade-off between the size of a cell and the statistical significance of the congestion metric in a cell. The two desirable quantities are high resolution and high statistical significance. For a high resolution, small cells are required; however, with small cells comes a fewer number of agents within it, hence there is a smaller statistical significance. This trade-off can be adjusted for individual cases, however, in this work, the cell size is generally on the order of l_c.

To increase the statistical significance of the congestion metric within a cell, a time history of agents can be used. This allows for a long term view of the airspace congestion.

It should be noted, this process is shares the unpredictable nature of road traffic. There is no way of predicting congestion without measuring congestion experimentally or having an accurate model of the traffic in an area to simulate congestion.

The resultant heat map will allow an airspace designer to find and apply navigation structures to reduce the congestion. It will also allow a designer to compare different navigation structures to find if there is a structure that performs better.

Chapter 4

Research Methods

In this chapter, all components of the research methodology performed in this work is outlined. The majority of this work is performed in simulation, hence identifying an appropriate simulation environment is outlined. Next, the method for setting up the simulation environment is described, including agents and test cases, but most importantly an appropriate timestep. Within experiments, transit efficiency, dimensionless conflict rate and dimensionless collision rate are the main metrics used to measure performance. A formal definition for each of these is identified. Next, an overview of the data collection processes is presented, including how runs were executed, how convergence was monitored and how data way analysed. Finally, experimental procedures for each experiment is outlined.

4.1 Research Questions

The questions to be answered in this investigation are threefold.

1. What features of an aerial routing network affect the dimensionless conflict rate and transit efficiency?
2. Under what circumstances does the orthogonal separation rule perform better than the separation rule?
3. Which method of measuring congestion has the strongest relationship with the collision rate? Does this method effectively show regions of high congestion in multi-cell environments?

Each of these questions will be explored in a series of simulated and practical experiments. The outline of the experimental method is presented in the following sections.

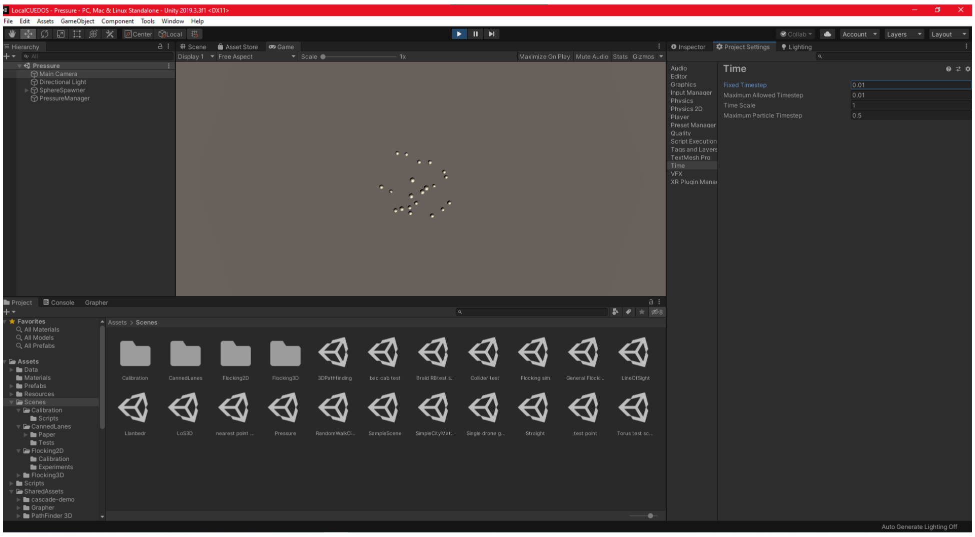
4.2 Simulation Environment

4.2.1 Introduction

In this section, the process of choosing and setting up a simulation environment is described. First, the requirements for a simulator are identified. Next, an appropriate timestep is outlined that gives physical results without using excessive computational time. Finally, the method for simulating agents and the environment are outlined.

4.2.2 Choosing a Simulator

Simulations could be performed in several different languages using any of several different physics engines. The requirements of a simulator for this work are for the physics to be physical, rule-based algorithms to be simple to implement, and results to be attainable. Unity was selected



Time

Fixed Timestep	0.01
Maximum Allowed Timestep	0.01
Time Scale	1
Maximum Particle Timestep	0.5

Figure 4.1. Screenshot of Unity real-time simulating environment.

as it meets these requirements, as well as having a good renderer for visualisations and many useful features for debugging and rapid prototyping, Figure 4.1.

Within the simulation, agents are objects that can have scripts attached. The language used for scripting is C#. Multiple custom and built-in scripts can be attached to an object, for example, attaching a rigid-body script to an agent makes it obey the principles of physics. A script of a rule-based algorithm can be attached and will compute all associated rules and apply a force to the agent.

4.2.3 Choosing a Timestep

A physics timestep must be selected that gives physical results. There are two requirements that must be satisfied for experiments to be valid.

- Collision detection - the timestep must be small enough to capture all head-on collisions and most glancing collisions.
- General physics – the timestep must be small enough to accurately replicate the theoretical quantities of Newton's laws of motion.

A timestep that satisfies each of these requirements was found, and the minimum was used for all future experiments.

Collision Detection

In Unity, a collider is a type of script that detects collisions. There are various methods within Unity for detecting collisions with various computational expenses [109]. The simplest and, therefore, least computationally expensive method, discrete collision detection, was tested and found to be sufficient. This method checks if any colliders overlap in each timestep. Due to the discrete nature of this checking, it is possible that objects may pass through each other without a collision being registered given a sufficiently large timestep.

A simple experiment was designed with the setup shown in Figure 4.2. In this experiment, two agents approach each other in anti-parallel directions, with velocity **v** and an offset r_{min}. The reference frame was made so that agent i (orange) is stationary. Agent j (blue) is travelling at the same speed v, in the anti-parallel direction. The closing velocity is thus 2**v**. The orientation of the reference frame is rotated so that the closing velocity is orientated in the standard $+y$ direction. At timestep t_1, agent j is infinitesimally close to agent i, however, no collision has occurred. At timestep t_2, agent j has moved distance Δx and is now infinitesimally close to agent i, however, having seemingly passed through agent i. Using discrete collision detection, no collision would have been detected.

The distance between the straight-line path between two timesteps and the centre of the stationary object is modelled as an effective change in the collision radius of the object. To find a suitable timestep for discrete collision detection, the relationship between the effective collision radius, r_{min} and the timestep.

From Figure 4.2, it can be seen that the minimum distance between the centre of the two circles is r_{min}. The distance blue moves with velocity 2**v**, in one time step is Δx. The diameter of the circles is D_c. The relationship between timestep, Δt and Δx is as follows.

$$\Delta x = 2v\Delta t \tag{4.1}$$

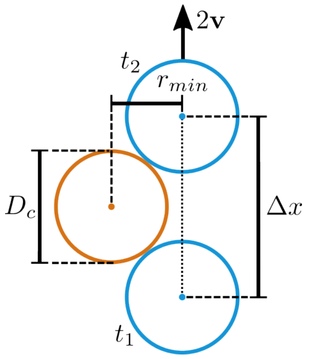


Figure 4.2. The effective collisions radius of a collision. In the frame of reference of the circle with the blue circle moving past with relative velocity $2\mathbf{v}$, in two timesteps, t_1 and t_2. We use $2\mathbf{v}$ as the worst case scenario is where both agents are moving in opposite directions with velocity ^{cs}. In timestep t_1, the blue circle is infinitesimally close to the red circle. In timestep t_2, the blue circle is also infinitesimally close to the stationary circle, having passed the red circle. If time was continuous, the centre of the moving circle would have been a distance r_{min} from the centre of the centre of the red circle and a collision would have been observed.

The value of r_{min} is found using Pythagoras' Theorem.

$$r_{min} = \sqrt{D_c^2 - \left(\frac{\Delta x}{2}\right)^2} \tag{4.2}$$

The value of Δx is normalised by D_c as a point of reference to find the distance travelled per timestep with respect to the diameter of an agent. This is repeated for r_{min} as well.

$$\Delta\tilde{x} = \Delta x / D_c,\ \tilde{r}_{min} = r_{min}/D_c \tag{4.3}$$

$$\therefore \tilde{r}_{min} D_c = \sqrt{D_c^2 - \left(\frac{\Delta\tilde{x} D_c}{2}\right)^2} = D_c\sqrt{1 - \frac{\Delta\tilde{x}^2}{4}} \tag{4.4}$$

$$\therefore \tilde{r}_{min} = \sqrt{1 - \frac{\Delta\tilde{x}^2}{4}} \tag{4.5}$$

We choose the allowable value for $\tilde{r}_{min} = 0.99$ that gives a value for $\Delta\tilde{x} \approx 0.28$. An example of a dimensional timestep for an agent of size $D_c = 1m$ and velocity $\mathbf{v} = 10ms^{-1}$ gives a timestep of $\Delta t \approx 0.014s$.

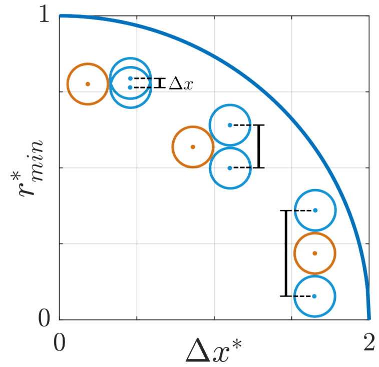
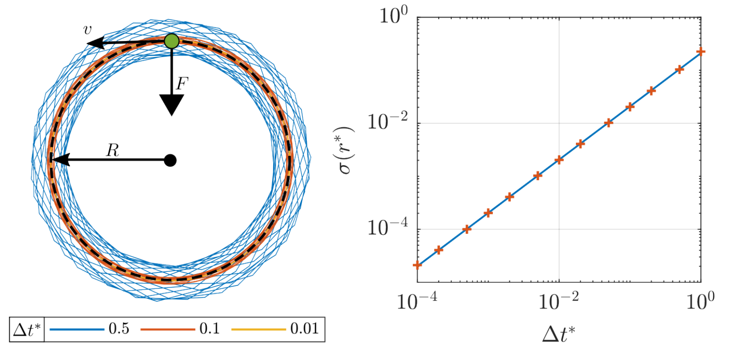


Figure 4.3. Relationship between normalised effective collision radius and normalised Δx. When time is effectively continuous, and $\Delta\tilde{x} = 0$, the normalised effective radius of the circle is 1, as timesteps are increased, Δx increases hence reducing the effective collision radius. When the timestep is so great that $\Delta\tilde{x} = 2$, a head on collision with a stationary circle can be missed.



(a) Experiment layout and example paths (b) Radius standard deviation against timestep

Figure 4.4. (a) Experimental layout of experiment, point mass at radius r from centre of environment. Velocity, $\mathbf{v}$, is tangent to the orbit and Force, F is applied towards the centre of orbit. Three example orbital paths show the overshooting with increasing timesteps. (b) Standard deviation of dimensionless radius of agent against dimensionless timestep on log-log axes.

Physics Engine

To verify the physics engine, a simple orbital experiment was performed with a point mass travelling at a constant speed with a force of constant magnitude applied towards the centre of the environment. The expected orbital radius R is found for the tangential velocity $\mathbf{v}$ and the force F applied to the agent. The actual distance from the centre is denoted by r. Non-dimensionalisation was performed on r and timestep, Δt, using the following

$$\tilde{r} = r/R,\ \Delta\tilde{t} = \frac{v\Delta t}{R} \tag{4.6}$$

The results in Figure 4.4 (a) show increased overshooting with increasing timestep. The standard deviation of the radius, r, for a range of timesteps shows a linear trend on log-log axes.

The fit coefficients are as follows:

$$\sigma(r) = \Delta t^{*} 1.0040 e^{-1.5734}, \; R^2 = 0.9966 \tag{4.7}$$

A standard deviation of the drift/overshoot that would be considered acceptable is 1% of the turning radius. This would give a dimensionless timestep of $\Delta t^* \approx 0.0488$. Given the dimensions of an agent with a maximum acceleration $a_{\text{max}} = 5$, a velocity of $5ms^{-1}$, and therefore a turning radius of $R = 5\ m$, the acceptable time step is $\Delta t \approx 0.0244\ s$.

From both investigations, the required timestep is 0.014 s, defined by the requirement for effective collision detection. This value is used in all future experiments.

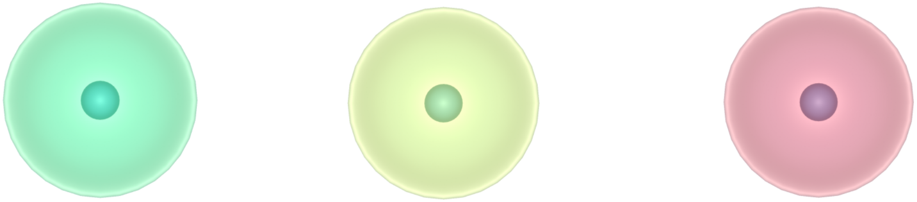
Numerical Stability

It is noted that numerical instability can occur in simulations when gains are increased such that agents saturate their maximum acceleration threshold in most timesteps. This will lead to oscillations in agent velocity. To ensure that simulations are remain positively stable, gains are maintained at a low enough level so as to ensure that the acceleration imposed on an agent is rarely saturated. Furthermore, timesteps are maintained at a level where, the change in velocity per timestep is considerably smaller than the reference velocity.

4.2.4 Simulating Agents

For the remainder of this work, agents are simulated as spheres with a diameter equal to the collision distance. The agent is inside a translucent sphere that has a colour that depends on the conflict state of the agent, Figure 4.5. The diameter of the translucent sphere is equal to the conflict distance.

Colour coding for agents and inflow/outflows is used. Orange agents use orange inflows and outflows. Similarly, blue agents use blue inflows and outflows. Yellow inflows and outflows are shared by both agent types.



(a) Deconflicted (b) Actively deconflicting (c) Critical Collision Event

Figure 4.5. Colour coding used to show agent conflict state in simulation.

4.2.5 Simulating Test Cases

The general components discussed are inflows and outflows. Components are described here with several methods in which they can be used.

Inflows are regions in which agents flow into the test case. The components of an inflow are the geometry of the inflow region, the inflow velocity, the agent type, and the inflow rate.

The geometry of an inflow region depends on the test case. For predefined topologies, the inflow geometry is defined by the geometry of the topology. For an emergent topology, inflow regions are at the discretion of the designer. The choices for the inflow geometries in this work are outlined in Section 4.6.3. All inflows must implement the following two considerations. First, the position where the agents' inflow should be uniformly random over the inflow region. Second, a method should be invoked to ensure that ensures an agent does not spawn such that it is in conflict with another agent. Due to the fact that it may not be possible to spawn an agent every time the spawn method is invoked, an *achieved spawn rate* variable should be used to state the actual inflow rate of the environment. In the results of this work, all inflow rates are the achieved inflow rate.

Inflows have an inflow rate that defines the regularity agents that flow into the environment. This is generally an independent variable that is used to change the density of the test case. The inflow rate is a global quantity, which means that the inflow rate is constant for the whole environment, regardless of the number of inflows. If there are multiple inflows, the sum of their individual inflow rates will be the global inflow rate. Randomisation of the inflow rate is achieved using the following formula.

$$t_{i+1} = t_i + \frac{2U}{\tau_i} \tag{4.8}$$

Where t_i is the time of the last agent to inflow, t_{i+1} is the time of the next agent to inflow, τ_i is the inflow rate and U is a uniformly pseudo-random variable generated such that $U \in [0\ 1]$.

When an agent is inflowing into the test case, the agent velocity is set such that ‘continuity’ is conserved. This is generally achieved by using the agent cruise speed and in the direction towards the destination of the agent.

Outflows are regions in which agents exit the test case. In this work, outflows are generally planes from which, when crossed, agents are removed.

Within the test case, any control method can be used to navigate agents from inflows to outflows.

4.3 Metrics

4.3.1 Introduction

At the time of writing, to the authors’ knowledge, there is no comprehensive list of metrics designed to measure the performance of a UTM. There are some rudimentary risk metrics in [4] and a wide selection of metrics that are not specific to the performance outlined in [18]. The purpose of this section is to outline the best practices used here for designing performance metrics and to define several metrics to be used to measure the performance of UTM planning and control methods. As a disclaimer, the work here is the initial concepts of how to design metrics with some metrics that were found to be useful in this work. As this field progresses, more metrics will become apparent and can be designed using this method and added to a more comprehensive list. The metrics used in this work are accumulated and adapted from previous work.

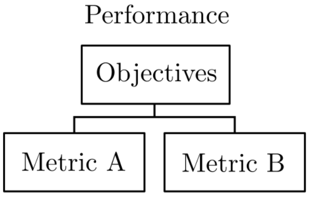


Figure 4.6. Measure of performance can be measured when an objective is set. The objective is a high-level metric which generally cannot me measured directly. Objectives can be broken into several measurable metrics.

As in standard practice, to find metrics that measure the performance of a system, the objectives of the system must be outlined. Once a set of objectives is outlined, reasonable metrics can be identified, as shown in Figure 4.6. Following best practices in metric design, metrics should be/have

- Low covariance - metrics should be independent as much as possible.
- Directional – metrics should have a beneficial direction.
- Complete set - metrics should cover all aspects of the objectives.

Some previously described UTM objectives are ‘to enable routine access to low-altitude airspace for small unmanned aerial systems in a safe, efficient, and fair manner’ [110]. Similarly, in [111] objectives include accuracy, safety, security, coordination, and compatibility. In this document, we focus on planning and control and hence will only use objectives in line with this. We choose the objectives of planning and control in UTM to be efficient and to meet appropriate safety margins. Several measurable metrics are described on the basis of the outlined objectives.

4.3.2 Transit Efficiency

Transit efficiency has been defined here as the average time taken to complete a flight compared to the fastest possible time the flight could have taken place without disruptions/deviations from the shortest flight path, as shown in Figure 4.7. This is the same approach as found in [112]. The reason for using time instead of distance is to allow comparability between agents with different mean speeds. This metric is formally defined as the deviation time Δt, and uses the theoretical

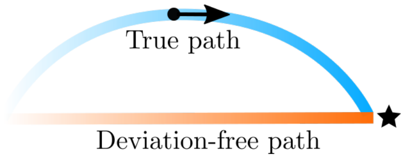


Figure 4.7. Comparison of true path to deviation-free path. Assuming the same speed, the true path will take longer, hence increasing Δt.

deviation-free time for agent i, t_i^d, and the true measured time for agent i, t_i^t.

$$\Delta t = \frac{1}{N} \sum_{i=1}^{N} \frac{t_i^d}{t_i^t} \tag{4.9}$$

Where N is the number of agents that are averaged over. If $\Delta t = 1$, then the UTM is operating at maximum transit efficiency. If $\Delta t = 0.5$, the average flight time is twice the deviation-free flight time.

4.3.3 Dimensionless Collision and Conflict Rate

The dimensionless collision rate is the collision rate divided by the inflow rate. The result of this is the average number of collisions per agent. Similarly, the same can be applied to make a dimensionless conflict rate. These are both useful as all agents are deconflicted when inflowing to a test case; hence collisions and conflicts are all due to the flow of agents in the test case.

4.4 Data Collection

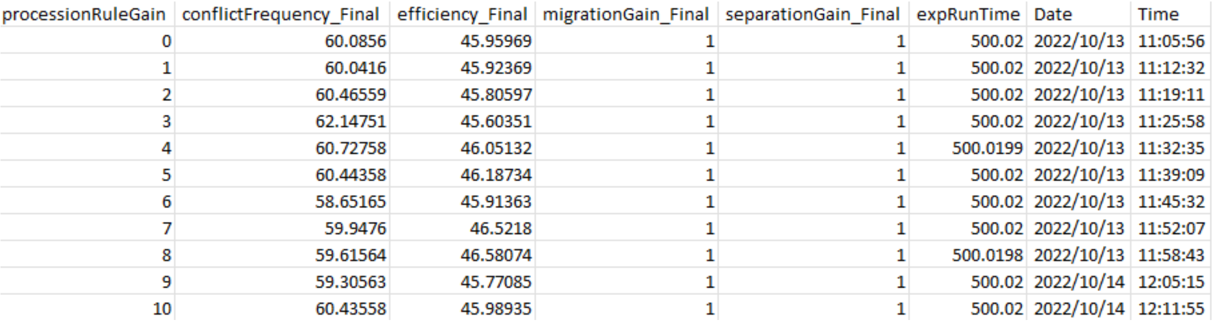
4.4.1 Simulation Experiment Manager

A C# script was developed to perform many runs consecutively. Independent variables are selected with values for minimum, maximum, and step size. Several dependent variables are selected to be recorded. Several control variables can also be selected to be recorded. The length of time for each run is found by performing a handful of preliminary experiments to find how long the dependent variables take to converge. This also acts as a debugging phase to ensure that all relevant variables are included and recorded as expected. All data are saved in CSV files.

4.4.2 Convergence of Simulation

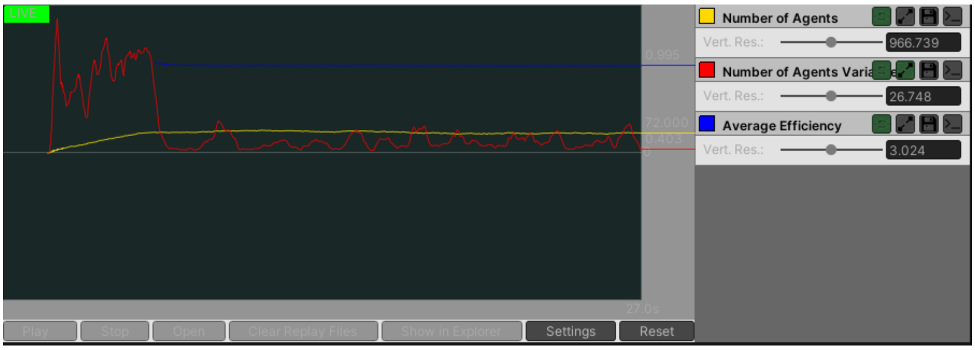
These experiments were set up to be continuous, meaning that they do not have a defined end. The simulation is considered complete once the dependent variables are converged to a point that satisfies the researcher. In practicality, this means that the simulation reaches a quasi-steady-state solution, where although the agents are still in motion, the dependent variables no longer change significantly. This approach is similar to a Monte-Carlo experiment, with a fundamental difference. In an example Monte-Carlo experiment, two dice are rolled. The result of the experiment is recorded and the experiment is repeated a large number of times. The more experiments are performed, the more clear what the most probable outcome of the experiment is. Finally, a probability distribution can be presented. If the experiment was changed so that instead of it being several discrete experiments, with each result recorded, but a continuous experiment, where each result is included in the computation of a mean, a similar result would ensue. This type of experiment is similar to how the experiments performed in this work are conducted. The longer the experiment is allowed to continue, the more converged the result will become.

The convergence of metrics was closely monitored in the Unity environment using a graphing tool, Figure 4.9. This tool is updated every timestep and allows an experimenter to observe the state of the dependent variables with respect to time.



processionRuleGain	conflictFrequency_Final	efficiency_Final	migrationGain_Final	separationGain_Final	expRunTime	Date	Time
0	60.0856	45.95969	1	1	500.02	2022/10/13	11:05:56
1	60.0416	45.92369	1	1	500.02	2022/10/13	11:12:32
2	60.46559	45.80597	1	1	500.02	2022/10/13	11:19:11
3	62.14751	45.60351	1	1	500.02	2022/10/13	11:25:58
4	60.72758	46.05132	1	1	500.0199	2022/10/13	11:32:35
5	60.44358	46.18734	1	1	500.02	2022/10/13	11:39:09
6	58.65165	45.91363	1	1	500.02	2022/10/13	11:45:32
7	59.9476	46.5218	1	1	500.02	2022/10/13	11:52:07
8	59.61564	46.58074	1	1	500.0198	2022/10/13	11:58:43
9	59.30563	45.77085	1	1	500.02	2022/10/14	12:05:15
10	60.43558	45.98935	1	1	500.02	2022/10/14	12:11:55

Figure 4.8. Example data in CSV from emergent aerial routing network experiment. Procession rule gain is the independent variable and the conflict frequency and transit efficiency are the dependent variables. Several controlled variables are also included. Experiment run time, date and time were also recorded.

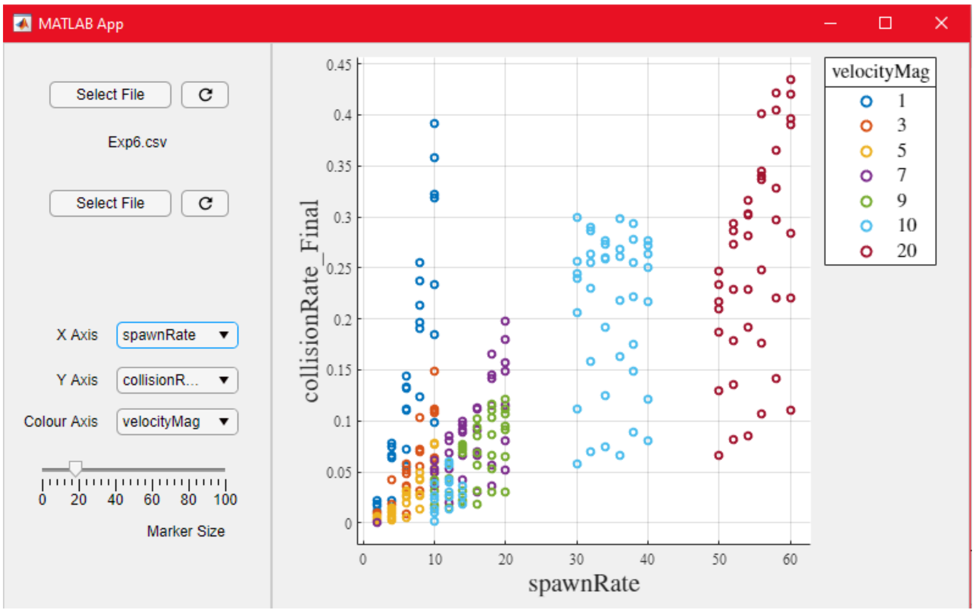


Number of Agents
Vert. Res.: 966.739
Number of Agents Varia
Vert. Res.: 26.748
Average Efficiency
Vert. Res.: 3.024

Figure 4.9. Screenshot of metrics being graphed for a particular run. The rolling graph populates from the right and slides to the left. Variables measured in this example are the number of agents (yellow), the variance in the number of agents (red) and the average transit efficiency (blue).

4.4.3 Data Analysis

Data analysis is performed in Matlab. Several Matlab applications were developed in this work. The most useful is shown in Figure 4.10 and was developed for rapid plotting of data. The plots were then further developed in a unique script for each figure.



Exp6.csv

X Axis spawnRate

Y Axis collisionR...

Colour Axis velocityMag

Figure 4.10. Matlab application developed in this work to rapidly plot data recorded by the simulation experiment manager. A particular file is selected along with the variables for the x, y and colour axis. In this example, data from initial congestion experiments is shown.

4.5 Uncertainty Analysis

Due to the perfect nature of simulation, there are no random errors introduced into the results that need to be quantified. Within the simulations, positions and velocities are known with almost infinite accuracy (without considering computational limits). This is not true for a practical experiment, where position and velocity are prone to errors in measurement and due to random errors such as wind. Communication between agents in simulation is instantaneous and fault free. In practical experiment, communication delays and packet loss are a source of error. These sources of errors are not simulated as it is out of the scope of this study, however, are considered in the practical investigation in Chapter 6.

4.6 Experimental Procedures

4.6.1 Introduction

Within this body of work, four studies were conducted; implementing and evaluating pre-defined aerial routing networks, emergent aerial routing networks, collision avoidance methods and the congestion metric. The procedure for each of these experiments is outlined in the following sections. For each of these experiments, previously described agents and environments are used unless noted in the experimental procedure.

4.6.2 Predefined Aerial Routing Networks

Several predefined aerial routing experiments were performed in this work. Validation experiments were first performed to ensure that all rules and methods functioned as expected. Following this, junction performance was evaluated.

Within all of these experiments, $u_{\text{ref}} = 5\ ms^{-1}$ and $a_{\text{max}} = 5\ ms^{-2}$ were chosen as agent characteristics. All dimensions and results are normalised by the turning radius of the agents, $l_c = u_{\text{ref}}^2/a_{\text{max}} = 5\ m$ and the inflow rate. Dimensionless results are expected to not change

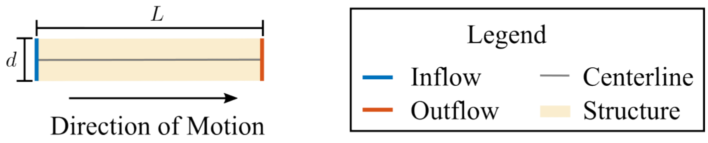


Figure 4.11. Straight test case for paths. Any path can be substituted as the structure within the experiment.

with these values, making the choice arbitrary.

In all of the following experiments, the separation rule is used as a simple collision avoidance method.

Straight Test Case

A set of two straight-line experiments was used to validate the geometry and flow rules. The two experiments were an *in-geometry* experiment and an *out-of-geometry* experiment. In both experiments, there was an inflow of agents that travelled directly to an outflow along a path. The difference between the experiments was that the in-geometry experiment has an inflow such that when flowing into the test case, agents are already within the geometry of the path. The out-of-geometry inflow has an inflow larger than the path geometry, hence requiring the use of the geometry rule.

In the experiments performed here, the only requirement is $L >> d$, particularly for the out-of-geometry experiments. This allows agents time to reach the geometry before the outflow. The variables chosen here are $L/l_c = 10$ and $d/l_c = 2$. τ_i was varied to show that the method continued to function correctly up to the point that the path became saturated. Out-of-flow geometry experiments had an inflow region size of $2d$.

Corner Test Case

A corner test case is used to validate that the waypoint-following method performs as expected. When an agent passes a waypoint, it will be directed towards the the next waypoint. When agents reach the corner, a slight overshoot is expected, the size of which will be relative to the rule gains, the reference velocity, the maximum load factor, and the angle of the corner. Corner angles, θ_C between 0 and 45 degrees will be considered as it is assumed that corners with greater angles can be broken into multiple corners.

Similarly, the variables chosen here are $L/l_c = 10$ and $d/l_c = 2$. τ_i was again varied to show that the method continued to function correctly up to the point where the path became saturated.

Converging and Diverging Junctions

The aim of the following experiments is to compare paths in converging and diverging junctions for different inflow rates. Performance was measured using dimensionless conflict rate. Experiments were split into three categories based on the number of dimensions used (1D, 2D and 3D). Tubes are comparable to both 2D and 3D paths; hence, they are compared in both categories.

There is only a single 1D path, lanes, hence these were used as a conflict rate baseline. The 2D junctions tested are in-plane strips, out-of-plane strips and helical tubes. The 3D junctions tested are corridors and helical tubes.

The length of the path is required to be significantly larger than the characteristic size of the path ($L_j >> d_j$). This was required so that there was no crossing of outflows at a diverging junction and allowed enough time for the agents to fully diverge. Figure 4.13 shows an example of a poorly designed junction. Furthermore, the characteristic size of the path must be greater than the turning radius of the agents. This is required specifically for tubes as agents must be able to stay within a reasonable distance from the tube geometry. Therefore, $d_j/l_c = 2$ and $L_j/l_c = 10$ were chosen to comply with the requirements.

The dependent variable of these experiments was the dimensionless density. Density is measured by dividing the inflow by the maximum possible inflow. The maximum inflow rate is

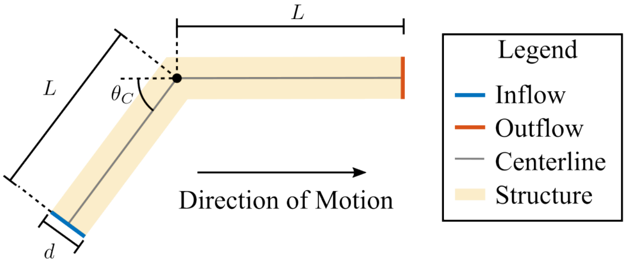


Figure 4.12. Corner test case for paths. Any path can be substituted as the structure within the experiment.

found based on the geometry of the path being used and the number of inflows. The maximum inflow is assumed to take the form of a simple cubic close-packed lattice, Figure 4.14. Although a hexagonal close-packed lattice is more likely, a simple cubic is simpler to convert to an inflow rate and is fit for purpose.

For a single lane, it can be seen that as the reference speed is increased, the maximum inflow rate is increased. Conversely, as the conflict distance increases, the maximum inflow reduced. This is due to the criteria that no agents can inflow into a conflict. From this, we can derive the maximum inflow rate of a single lane to be:

$$\tau_{i,\text{max}} = \frac{u_{\text{ref}}}{d_c} \quad \text{(Single Lane)} \tag{4.10}$$

If there are two inflow regions, there will be twice the number of agents able to inflow, therefore:

$$\tau_{i,\text{max}} = \frac{N^i u_{\text{ref}}}{d_c} \quad \text{(Multiple Lanes)} \tag{4.11}$$

Where N^i is the number of inflow regions. Recall that the inflow rate is a global quantity and hence considers all agents inflowing from all inflow regions. Moving to 2D paths, when considering how many agents can spawn on a strip, it can be seen that for a path with $d_j = 2d_c$, it is possible for 3 agents to spawn, Figure 4.15 (a). Although very unlikely, this is the absolute maximum that can inflow in a given space. Therefore, the maximum number of agents that can fit side by side on a strip is $(d_c + d_j)/d_c$. Combining this with the maximum inflow of a lane, the following is found:

$$\tau_{i,\text{max}} = \frac{N^i u_{\text{ref}}(d_c + d_j)}{d_c^2} \quad \text{(Strip)} \tag{4.12}$$

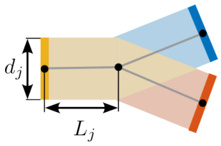
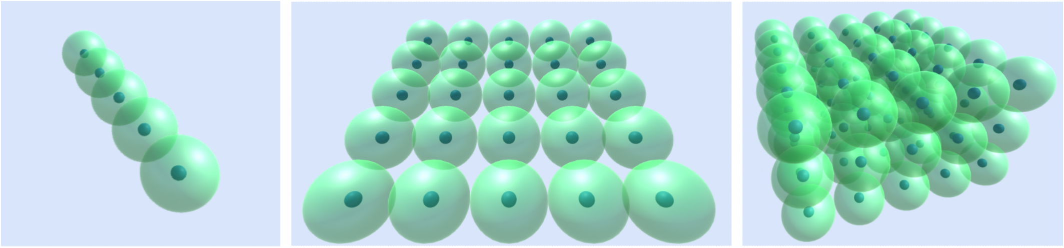
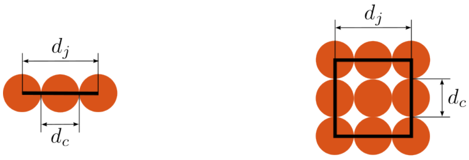


Figure 4.13. Example of a poorly designed experiment as agents do not have adequate time/space to diverge.



(a) Lane (b) Strip (c) Corridor

Figure 4.14. Maximum agent density for different density metrics. If agents inflow into the environment in a lattice like this, the dimensionless density will be 1.



(a) Strip (b) Corridor

Figure 4.15. Maximum number of agents that can inflow on a strip and corridor.

Finally, when considering a corridor, the maximum number of agents that can fit side-by-side is $(d_c + d_j)^2/d_c^2$. Hence the maximum inflow for a corridor is:

$$\tau_{i,\text{max}} = \frac{N^i u_{\text{ref}}(d_j + d_c)^2}{d_c^3} \quad \text{(Corridor)} \tag{4.13}$$

This equation can be generalised for all path in the following way.

$$\tau_{i,\text{max}} = \frac{N^i u_{\text{ref}}(d_j + d_c)^{N^d - 1}}{d_c^{N^d}} \tag{4.14}$$

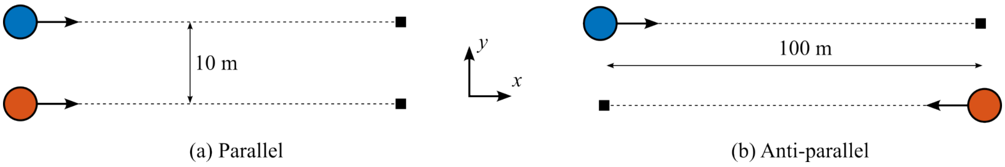
where the number of dimensions is denoted by N^d and is 1 for lanes, 2 for strips and 3 for corridors. The dimensionless density is therefore

$$\tilde{\rho} = \tau_i/\tau_{i,\text{max}} = \frac{\tau_i d_c^{N^d}}{N^i u_{\text{ref}}(d_j + d_c)^{N^d - 1}} \tag{4.15}$$

Dimensionless density is also used for tubes to ensure a fair comparison. When comparing a tube to a corridor, the maximum inflow rate of the corridor is used because the tubes still use the same volume of space. Similarly, when comparing a tube with a strip, the maximum inflow of a comparable strip is used.

Table 4.1. Junction experiments variables

Variable	Value	Unit	Variable	Value	Unit
Reference Speed, u_{ref}	5	ms^{-1}	Junction Angle, θ_j	20	°
Flow Rule Gain, $K^{\mathcal{F}}$	1	n.d.	Helix Angle, θ_h	45	°
Geometry Rule Gain, $K^{\mathcal{G}}$	1	n.d.	Path Size, d_j	10	m
Rule Interval, τ_r	0.1	s	Path Length, L_j	50	m
Max Acceleration, a_{max}	5	ms^{-2}	Collision Distance, D_c	1	m
Acceleration Gain, K^a	1	n.d.	Conflict Distance, d_c	5	m
Separation Gain, $K^{\mathcal{SS}}$	1	n.d.			



(a) Parallel

(b) Anti-parallel

Figure 4.16. Experimental set up for characterising the motion of agents using the procession rule. Circles are the agents starting point and velocity, destinations are represented as squares. It is expected that agents are attracted in (a) the parallel case and repelled in (b) the anti-parallel case.

The variables were set to those found in Table 4.1 unless otherwise noted.

4.6.3 Emergent Aerial Routing Networks

The following experiments are used to characterise and measure the performance of the procession rule. Unless stated otherwise, the variables used in the following experiments are as found in Table 4.2.

Characterising the Procession Rule in Two Agent Experiments

The aim of this experiment is to ensure that the procession rule behaves consistently with expectations. Two agents were placed 10 m apart on the y-axis. For parallel experiments, the agents were placed at the same location on the x-axis, for antiparallel experiments, the agents were placed 100 m apart on the x-axis. In both cases, the agents used the migration rule to travel to a

Table 4.2. Emergent Lane Formation experiments variables

Variable	Value	Unit	Variable	Value	Unit
Reference Speed, u_{ref}	5	ms^{-1}	Inflow Rate, τ_i	1	s^{-1}
Migration Rule Gain, $K^{\mathcal{M}}$	1	n.d.	Collision Distance, D_c	1	m
Separation Rule Gain, $K^{\mathcal{S}}$	1	n.d.	Conflict Distance, d_c	5	m
Rule Interval, τ_r	0.1	s	Acceleration Gain, K^a	1	n.d.
Max Acceleration, a_{max}	5	ms^{-2}	Distance Weighting: a, b	0.25, 15	n.d.

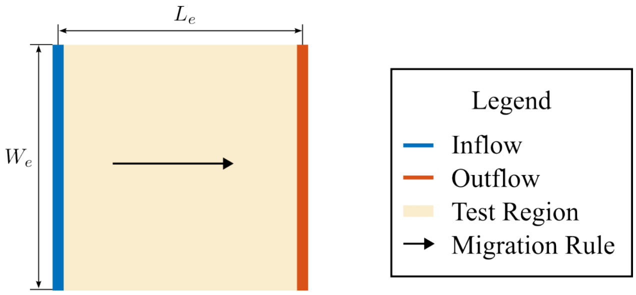


Figure 4.17. Characterisation of the structures formed in a uni flow test case due to the procession rule.

destination placed 100 m from the agent, Figure 4.16. A range of values for a, b and $K^{\mathcal{P}}$ were used to ensure the distance magnitude weighting behaved as expected.

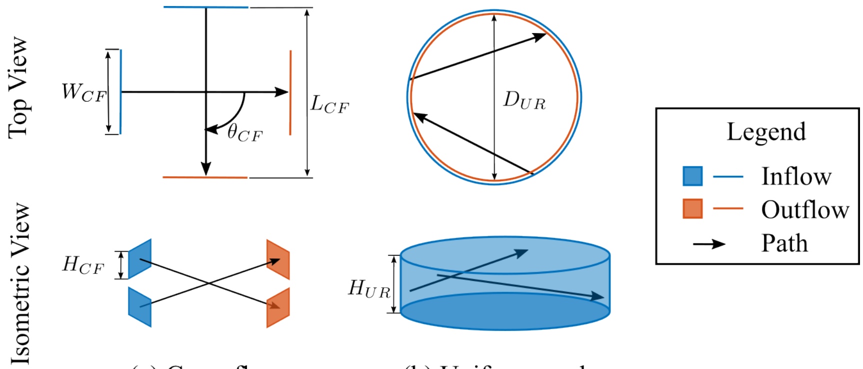
Characterising the Procession Rule in Uni Flow Test Cases

The aim of this experiment is to analyse the structures formed by the procession rule in a uni flow test case. The test case is set up as shown in Figure 4.17. The size of the test case is defined by the width W_e and the length L_e. The width must be large enough to ensure that the boundary has no effect on the structure found in the centre. To find the point at which boundary no longer affects the structure of the central flow, the width was increased until no noticeable change was found in the central structure. The length of the test case must be long enough to allow the structure to become fully established. This was incrementally increased until the structure was formed satisfactorily.

Flow Test Cases for Evaluating Procession Rule

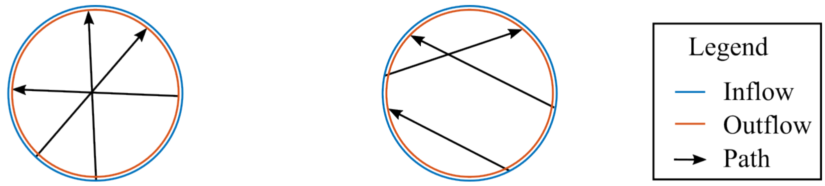
Flow test cases are designed to perform experiments that evaluate the procession rule. These flow test cases aim to mimic real-world UTM environments.

Two types of flow are proposed and are shown visually in Figure 4.18. All test cases use the principles described in Section 4.2.5. The *cross flow test case* is a flow environment in which two bulk flows cross, for example, between buildings or restricted airspace. A special case exists when $\theta_{CF} = \pi$ and agents travel in opposite directions, here referred to as *contra flow test case*. This test case is commonly used in modelling lane formation [29]. The *uniform random test case* is a uniform random inflow of agents from the sides of the cylindrical test region. This



(a) Cross flow (b) Uniform random

Figure 4.18. Flow test cases schematic diagram. These test cases aim to mimic standard flows found in a UTM. Fundamentally, agents flow into the test region from inflows and out through outflows. (a) Simple bulk flows cross with a given crossing flow angle θ_{CF}, and have three spacial dimensions: length, L_{CF}; width, W_{CF} and height, H_{CF}. This aims to mimic flows between restricted airspace or obstacles. (b) A random flow with agents travelling in all radial directions in a cylindrical volume with diameter, D_{UR} and height, H_{UR}. The flow is radial as agents do not inflow or outflow through the top and bottom of the cylinder.



(a) Opposite Destination (b) Random Destination

Figure 4.19. Outflow types for different uniform random test cases. (a) The opposite outflow has agents travelling to the opposite side of the cylindrical test region. (b) The random outflow has agents outflowing from a random point on the side of the cylindrical test region.

type of geometry was chosen because it most accurately mimics a UTM with agents generally travelling in the horizontal plane with minor changes in altitude. There are two special cases of the uniform random test case depending on the outflow of agents, shown in Figure 4.19. The *opposite* outflow test case has agents outflowing from the point at the same altitude, but the opposite side of the cylindrical test region. This is a test case used in [113]. The random outflow test case has agents outflowing from a random point chosen on the side of the cylindrical test region.

Outflows are constructed individually for each agent and use the same logic as passing a waypoint on a path. That is, for each agent, a destination point (analogous to a waypoint) and a direction vector (from inflow to outflow) are generated. Every timestep, the agent checks to see if it has passed the 'outflow plane', which is coplanar with the destination point and orthogonal to the direction vector. Once it has passed, it outflows from the test region and is destroyed.

Experiments are performed by randomly sampling inflow positions and times. The positions are sampled from a uniform distribution over the inflow boundary. The time for the next agent to flow into the test region is sampled from a uniform distribution with a minimum at 0 and a maximum at twice the inflow period (inverse of the inflow rate), Equation 4.8. This is to introduce randomness into the individual inflow period of each agent without affecting the overall inflow frequency. A uniform distribution was chosen as it has step limits and is simple to implement.

Evaluating the Performance of the Procession Rule

The aim of this experiment is to compare the performance of the two implementations of emergent lane formation. Performance is based on dimensionless collision rate and transit efficiency.

All possible independent variables are as follows. The agent variables: a_{max}, u_{ref}, τ_i, d_c, D_c. The rule variables: $K^{\mathcal{P}}$, a, b, $K^{\mathcal{SS}}$. The environment variables, for cross flow test cases, θ_{CF}, L_{CF}, W_{CF}, H_{CF}, and for uniform random test cases D_{UR}, H_{UR}. Lastly, these are to be tested on 2 implementations. As it would be impractical to perform experiments varying all variables, several particularly interesting experiments were selected to measure the performance of the procession rule. All variables were consistent with Table 4.2, unless stated.

Firstly, initial experiments were performed in a 2D contra flow test. This experiment aimed to find how changes to the weightings of the rules affect the performance. The environment was set up with $[L_{CF},\ W_{CF},\ H_{CF}]/l_c = [10,\ 10,\ 0]$. The independent variables were a, b and $K^{\mathcal{P}}$.

Next, a follow up experiment were performed in the same 2D contra flow test case. This experiment tested how inflow rate affected the performance of the procession rule. The environment was set up with $[L_{CF},\ W_{CF},\ H_{CF}]/l_c = [10,\ 10,\ 0]$. The independent variables were τ_i and $K^{\mathcal{P}}$.

Moving to 3D experiments, a two cross flow tests was performed at $\theta_{CF} = 90^\circ$ and $\theta_{CF} = 180^\circ$. This experiment tested how the $K^{\mathcal{P}}$ affected the performance of the environment. The environment was set up with $[L_{CF},\ W_{CF},\ H_{CF}]/l_c = [10,\ 10,\ 1]$. The independent variables were θ_{CF} and $K^{\mathcal{P}}$.

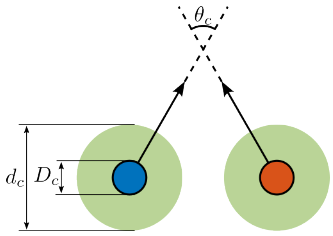


Figure 4.20. Experimental set up for simple two agent collision avoidance experiments. Closing velocity was varied using the crossing angle. For a range of closing velocities and dimensionless conflict distance (DCD), experiments found if agents experienced a collision or not.

Finally, two uniform random tests were performed. This experiment tested the performance of the procession rule when there is less bulk flow. The environment was set up with $D_{UR}/l_c = 20$ and $H_{UR}/l_c = 10$. The independent variables were the destination type (destination opposite or random destination) and $K^{\mathcal{P}}$.

4.6.4 Evaluating the Performance of Collision Avoidance Methods

Collision avoidance methods were first characterised in a simple two agent test case. The following experiments evaluate the performance of both collision avoidance methods in uniform random test cases.

Characterising Collision Avoidance Methods in Two Agent Experiments

The aim of these experiments is to find an envelope of safety for each collision avoidance method for a given closing velocity and dimensionless conflict distance (DCD). Recall, DCD $= d_c/D_c$.

The two agent test case is set up as shown in Figure 4.20. Closing velocity was varied by changing the crossing angle θ_c. An experiment was performed for each collision avoidance. Each experiment consisted of many runs with $\theta_c \in [0^\circ \ \ 180^\circ]$ and $d_c/D_c \in [1 \ \ 10]$. The initial runs used a coarse range of d_c/D_c and θ_c. The final runs used a fine range of variables with maximum steps of $\Delta\theta_c = 5$ and $\Delta d_c/D_c = 0.5$. Additional manual experiments were performed with a finer step for some regions in the range. For each run, if a collision was detected, the run was marked with a collision occurred flag, and vice versa. The final results are presented using the first run in which a collision *not* occurred for each DCD.

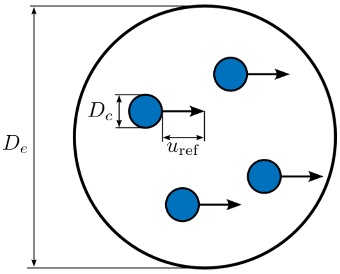


Figure 4.21. Experimental set up for comparing congestion metric implementations.

Evaluating the performance of Collision Avoidance Methods

The aim of this experiment is to compare the performance of collision avoidance methods in uniform random test cases.

Experiments were carried out in both uniform random test cases. The dependent variable in the experiments was the DCD. Performance was measured using dimensionless collision rate and transit efficiency. Each experiment was performed 3 times, from which the mean and standard deviation were found. The geometry of the test case was $D_{UR}/l_c = 20$ and $H_{UR}/l_c = 10$.

4.6.5 Measuring Congestion

Comparing Implementations

The aim of these experiments is to find the congestion implementation that has the strongest relationship with the collision rate.

The test case used for these experiments is a sphere with a diameter D_e. Agents inflow with a uniform random distribution over the surface of the sphere. Agents have a collision distance D_c and have no collision avoidance. To vary the collision rate within the test case, velocity direction, velocity magnitude u_{ref}, agent size (synonymous with collision distance) D_c, inflow rate τ_i and the diameter of the test case D_e can be changed. The agent velocity direction is varied using normal distributions. Agent velocity can be represented in spherical coordinates with a radial distance, r, polar angle, θ and azimuth angle φ. The velocity magnitude is set by $r = u_{\text{ref}}$. The direction is set using a normal distribution for θ and ϕ. The mean of the normal distributions is 0 and the standard deviation $\sigma(\theta)$ and $\sigma(\phi)$ is increased to raise the variance between agent velocity directions. The standard deviation are applied to both θ and ϕ hence will always be refered to simply as $\sigma(\theta)$. This method is fit for purpose as the aim is to be able to control the velocity variance between agents.

Each congestion implementation is recorded for a full-factorial experiment with $u_{\text{ref}} \in [1\ 20]$, $\sigma(\theta) \in [0\ 10]$ and $\tau_i \in [2\ 10]$. The relationship between the recorded collision rate and the various congestion implementation measures is compared. A relationship is considered strong if there is a correlation between the measured and actual collision rate.

Validation Experiments in Standard Flow Experiments

The aim of these experiments was to validate the congestion metric in the cross and uniform random test cases. For larger cross-flow angles θ_{CF}, the congestion metric is expected to be greater than smaller cross flow angles. Similarly, it is expected that the centre of a destination opposite the test case would have a large value of congestion, while a random destination would have a uniform value of congestion. Each of the test cases was discretised into 10 m cells and congestion was measured at each time step. This size was chosen as it is in the order of the turning radius of agents. The average value of congestion was recorded.

Example Use Case in Procedural Generated City

This experiment aims to present congestion as a useful metric to measure congestion in an entire city test case. A procedurally generated city was used to allow experimentation of city-like environments that are not geometrically consistent.

The geometry of the buildings is described by three macroscopic properties: floor area ratio (FAR), building coverage ratio (BCR) and block area. The block area is the area of the entire site where a building is built. FAR is the ratio of the total floor area of a building to the block area. BCR is the ratio of the ground floor area to the block area. FAR and BCR are frequently used in town planning; hence, it is useful to use them as the values are abundant for different

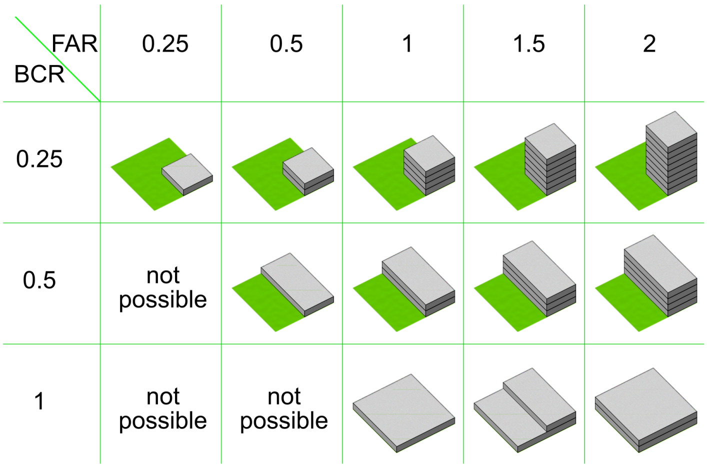


Figure 4.22. Visual definition of floor area ratio (FAR) and building coverage ratio (BCR). (Amended from [114])

Table 4.3. Common values for FAR and BCR

Region	BCR	FAR
Low-Rise Residential	0.3 – 0.6	0.5 – 2
High-Rise Residential	0.3 – 0.6	1 – 5
Commercial	0.7 – 0.9	2 – 13
Industrial	0.5 – 0.6	1 – 5

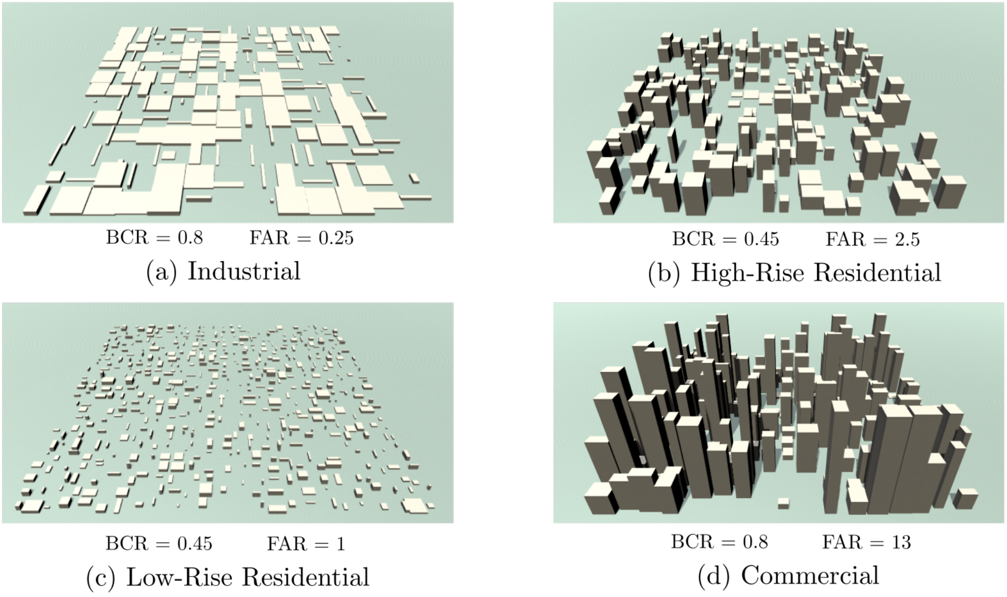
regions and types of cities. Standard values for different types of region are presented in Table 4.3. FAR and BCR are visually defined in Figure 4.22.

An example of several procedurally generated city tiles is shown in Figure 4.23.

Several of these tiles can be combined to generate larger cities, as shown in Figure 4.24.

Congestion was measured in a procedurally generated city by inflowing agents from distribution centres and assigning them a random destination. Once a destination is assigned, the agent requests a path that avoids obstacles. This method uses an A* pathfinding algorithm[1] and returns a list of waypoints for the agent to follow. This method has static obstacle avoidance, but will not avoid other agents. The simple separation rule was used to minimise collisions. Agent follow the path using the migration rule. Once the agent arrives at the destination, it follows the same path to return to the distribution centre.

[1] Credit to Pathfinding 3D tool in Unity Asset Store - https://assetstore.unity.com/packages/tools/ai/nav3d-100285



BCR = 0.8 FAR = 0.25

(a) Industrial

BCR = 0.45 FAR = 2.5

(b) High-Rise Residential

BCR = 0.45 FAR = 1

(c) Low-Rise Residential

BCR = 0.8 FAR = 13

(d) Commercial

Figure 4.23. Examples of procedurally generated city tiles.

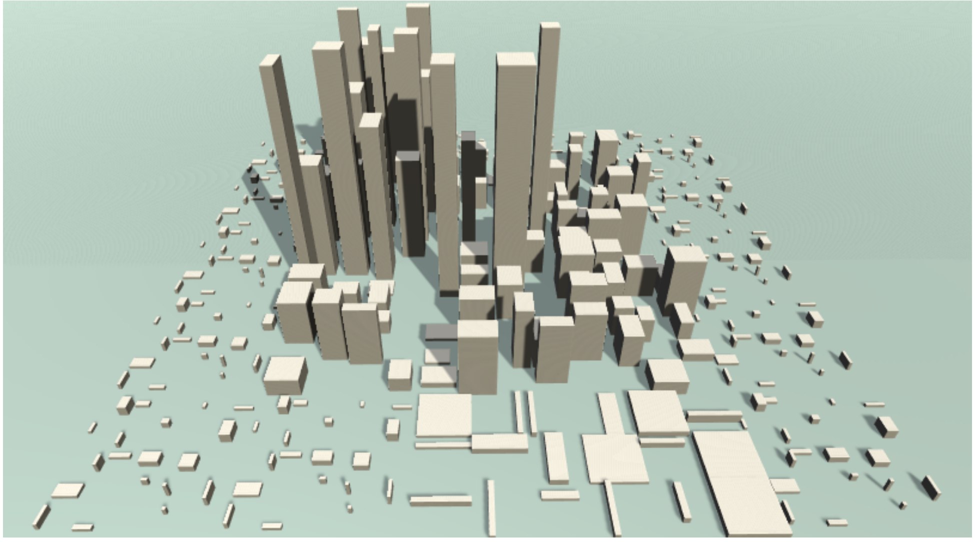


Figure 4.24. Example of several tiles combined together.

Chapter 5

Results

This chapter presents the results from this study in four sections. First, all components of the pre-defined aerial routing network are validated, followed by the evaluation of various junctions. Results reveal that there is a significant difference in performance of a junction based on the geometry of the path, as expected. Next, emergent aerial routing networks are implemented and characterised in several test cases. In these experiments, it was found that emergent aerial routing networks significantly reduce collision rate and increase transit efficiency, but only in certain types of flows. Next, a qualitative and quantitative comparison of the simple separation rule and the orthogonal separation rule is presented. From this work, it is clear that both methods have merit, however similar to emergent aerial routing networks, performance is flow dependent. Finally, the several methods for quantifying a congestion metric are compared, revealing that the gas kinematic method has the strongest relationship. This method is then used in several multi-cell test cases where it effectively identifies regions of high and low congestion.

5.1 Predefined Aerial Routing Networks

5.1.1 Introduction

In this section, results for predefined routing network experiments are examined. Initial validation tests were performed to show the geometry and flow rule perform as expected. Following this, validation of the waypoint update method was performed. Finally, the performance of converging and diverging junctions was evaluated.

5.1.2 Validation Experiments

Flow Rule

The agents behaved as expected, as seen in Figure 5.1, remaining within the geometry and travelling with the predefined flow. The only exception to this is for helical tubes. A video of the simulation results can also be found here[1].

In a helical tube, the introduction of a tangential velocity component to the flow rule causes an increase in the effective radius of the tube, everything else being equal. Although it is possible to minimise this effect in code such that the helicity of a tubular flow can be varied independently of tube radius, this was not implemented in order to reduce complexity. Investigation of the interaction between the flow rule and geometry rule showed that as long as the tube radius was larger than the agent minimum turn radius and the helix angle was less than 45°, the effective radius of the resulting helical tube can be guaranteed to be within 10% of the target radius.

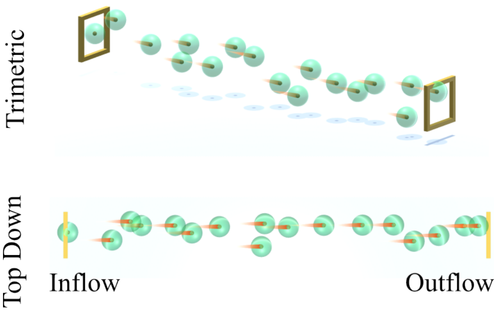


Figure 5.1. Simulation results of agents flowing from inflow to outflow of a straight corridor.

[1] Single Path Video - https://youtu.be/ybgijCbioIE

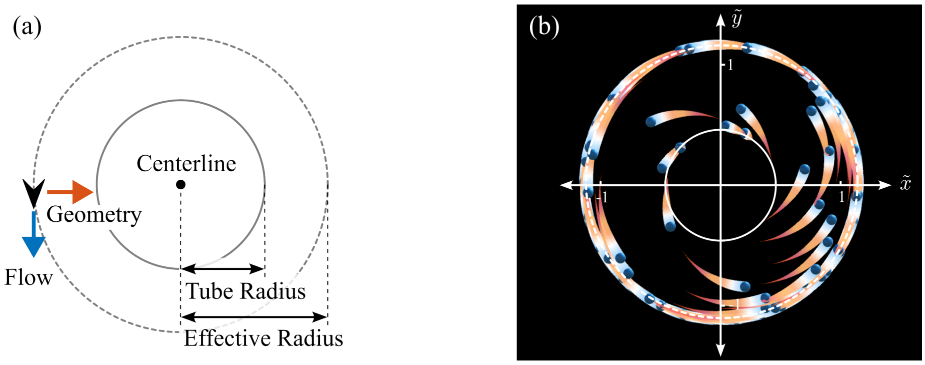
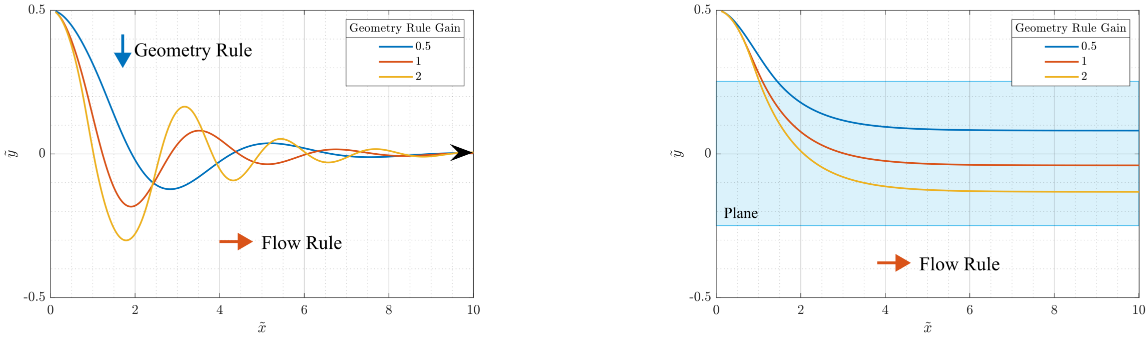


Figure 5.2. Resultant cross-section of helical tubes. (a) Overview sketch of accelerations on an agent resulting in an effective radius. (b) Numerical simulation snapshot with agents flowing into test environment on the tube radius and being accelerated outwards. Equilibrium is reached at the effective radius. Axes have been non-dimensionalised with the turning radius of agents.



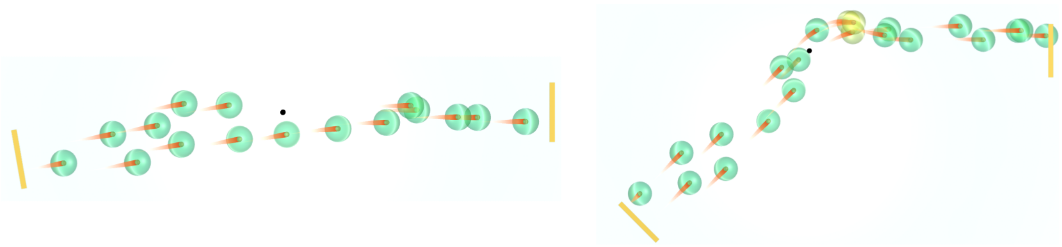
(a) Geometry rule on a lane

(b) Geometry rule on a strip

Figure 5.3. Validation of the geometry rule by visualising agent paths. Agents start at $0.5\ l_c$ from the centerline of the path, which lies on the x-axis. (a) For a lane, it can be seen that for increasing geometry rule gains, the overshoot increases, but the initial convergence point is earlier. (b) For a strip that has a width on the order of the turning radius of an agent, once the agent has entered the strip, the lateral component of velocity decays.

Geometry Rule

The behaviour of the agents was as expected when flowing into an environment not already within the confines of the path geometry. Figure 5.3 shows the path of an agent that is accelerated towards a lane by the geometry rule, while still moving forward in the x direction due to the flow rule. The agent exhibits some overshoot, as expected. It is not possible to generalise on the characteristics of the overshoot due to the nonlinear nature of the agents control algorithm; however, it is clear that the path of the agent is similar to a classic P-controller output. The gain of the geometry rule is synonymous with the gain of a P-controller, where a higher gain causes faster initial convergence, but has large overshoot. The difference between the two control algorithms is that the geometry rule becomes saturated at a certain distance from the path.



(a) $\theta_C = 10°$

(b) $\theta_C = 45°$

Figure 5.4. Simulation results of agents turning corners of differing angles. The overshoot exhibited by agents is exacerbated by larger corner angles, θ_C. Agents in yellow show a conflict has occurred.

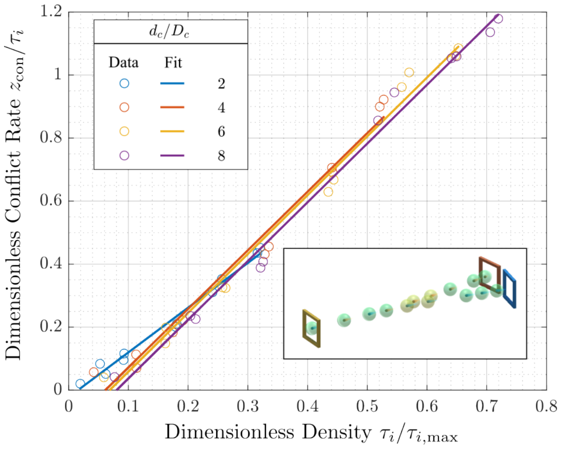


Figure 5.5. Evaluating the performance of converging junction of lanes for various dimensionless densities. The junction angle was $\theta_j = 20°$ for all runs. Resulting linear trends are presented for each value of DCD. It is clear that there is a linear correlation between dimensionless density and dimensionless conflict rate. This presents a useful baseline for all other junctions.

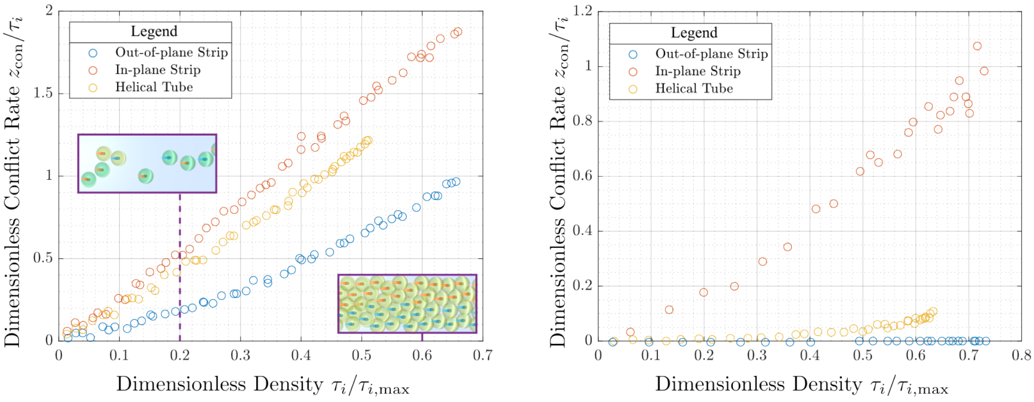
Waypoint Updating

In the simple corner turn experiment, two observations were made. First, the waypoint updating method performed as expected. Second, in all cases, there was an amount of overshoot, which was expected when passing a waypoint, as shown in Figure 5.4. The parameters that contribute to the overshoot are the gain of the geometry rule, $K^{\mathcal{G}}$; the reference velocity, u_{ref}; the maximum acceleration, $a_{\max}$; the rule interval, t_i; and the corner angle, θ_C.

5.1.3 Evaluation of Converging and Diverging Junctions

Evaluating 1D paths

Lanes are the only 1D path. Experiments were performed on lanes as a baseline for the following experiments. Figure 5.5 shows the dimensionless conflict rate for various dimensionless



(a) Convergence

(b) Divergence

Figure 5.6. Dimensionless conflict rate against dimensionless density for various strip convergence and divergence junctions. Insets show simulation results at various densities. The characteristic size of all path cross-section is $d = 10$, and the junction angle is $\theta_j = 20°$. All other variables were consistent with Table 4.1. Simulations were run for a minimum of 1000 agents.

densities in a converging junction. This experiment was repeated for several values of Dimensionless Conflict Distance (DCD). The results show that the dimensionless conflict rate remains consistent for different DCD values. A linear fit is used to show the similarities between the plots.

Diverging lane junctions have no conflicts and hence are not considered any further.

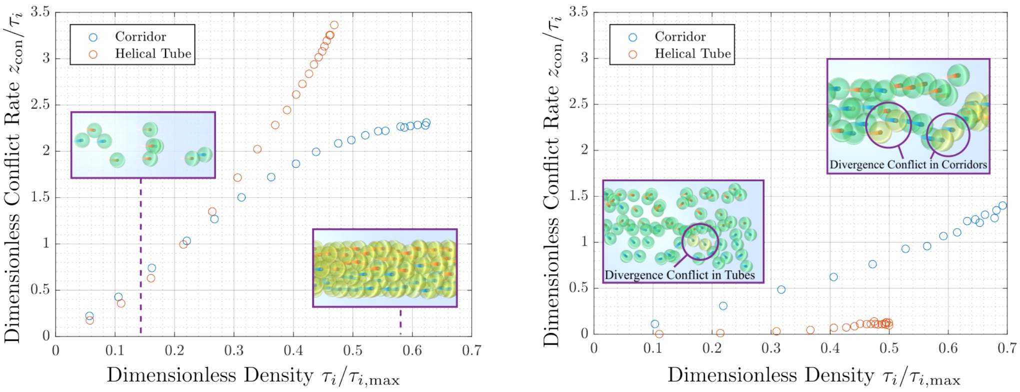
Evaluating 2D Paths

The three junctions evaluated are in-plane strip junctions, out-of-plane strip junctions, and counter-rotating helical tube junctions.

It is clear from Figure 5.6 that the junction with the highest conflict rate is the in-plane strip. This is expected as agents are more likely to cross paths and hence be in conflict with each other. It can also be seen that an out-of-plane strip in a diverging junction is completely conflict-free. This is also consistent with expectations, as paths of agents should never cross when departing from a strip out-of-plane.

Finally, it can be seen that the helical tubes perform in between these two types of strips. In a convergence junction, the conflict rate is very close to an out-of-plane strip junction. A video can be found here[2] that compares out-of-plane strip convergence junctions with helical tubes.

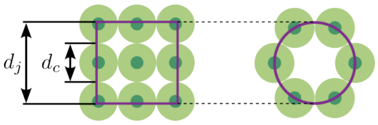
[2] Strip and Helix Junction - https://youtu.be/3fiB6gxLsDY



(a) Convergence

(b) Divergence

Figure 5.7. Dimensionless conflict rate against dimensionless density for corridors and helical tube convergence junctions. Insets show simulation results at various densities. The characteristic size of all path cross-sections is $d = 10$, and the junction angle is $\theta_j = 20°$. All other variables were consistent with Table 4.1. Simulations were run for a minimum of 1000 agents.



(a) Corridor

(b) Tube

Figure 5.8. Comparison of inflows between corridors and tubes. For the same characteristic size of path cross-section, corridors generally are able to inflow more agents than tubes. The result is the ability to achieve larger dimensionless densities in corridors.

Although it is assumed that helical tubes behave similarly to out-of-plane strips, the additional interactions are attributed to the additional confinement caused by the wrapping of the tube.

Evaluating 3D Paths

The two evaluated junctions are corridor junctions and counter-rotating helical tubes.

It can be seen that for convergence junctions, helical tubes and corridors perform very similarly for low dimensionless densities, however, for larger dimensionless densities, helical tubes exhibit significantly larger dimensionless conflict rates, Figure 5.7. Another difference between corridors and helical tubes is that the corridor is able to use more of the volume allocated for the junction, hence reaching larger dimensionless densities.

When considering diverging junctions, it is clear that helical tubes have a much lower conflict rate than corridors. This is because helical tubes behave similarly to out-of-plane strips, which are conflict-free in divergence junctions.

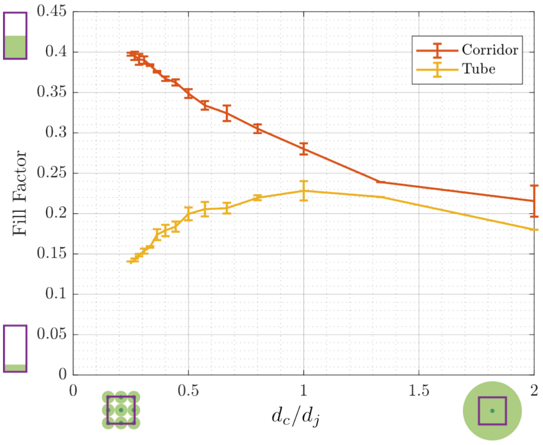


Figure 5.9. Fill factor of corridors and tubes with changing d_c/d_j. For a constant agent conflict distance, if the paths size is increased, hence reducing d_c/d_j, the fill factor of a tube is reduced. In contrast, for a corridor, the fill factor increases.

The discrepancy between the maximum capacity of helical tubes and corridors is due to helical tube inflows being smaller than the corresponding corridor inflow, see Figure 5.8.

In a brief follow-up experiment, fill factor was used to compare the capacity of paths. Fill factor is a dimensionless quantity that measures how full a volume is. It is generally used in physical chemistry when measuring the effectiveness of atom/molecule packing structures. The packing structures of the agents in a path is complex, hence, the fill factor was found experimentally.

The experiment was set up by inflowing the maximum number of agents in a straight path experiment. The fill factor was found by dividing the volume occupied by agents by the total available volume.

The result of the experiment shows that the fill factor of tubes reduces as the d_c/d_j ratio is reduced, Figure 5.9. In contrast, the corridor fill factor increases as the d_c/d_j ratio is reduced. This result shows that as d_c/d_j is reduced in a junction, the maximum dimensionless density for a tube will be reduced.

5.2 Emergent Aerial Routing Networks

5.2.1 Introduction

In this section, emergent aerial routing networks are implemented and evaluated based on transit efficiency, dimensionless collision rate, and dimensionless conflict rate. Initial experiments observed the affects of variables within the procession rule on the motion of individual agents. Next, the structure of emergent aerial routing networks is examined in a uni flow test case. The following experiments measure the performance of the procession rule for various values of a, b and $K^{\mathcal{P}}$. Next, experiments were performed to compare the velocity-based implementation with a destination-based implementation in a simple 2D contra flow test case. Finally, a 3D study of the destination-based implementation was performed in both the cross-flow and uniform random test cases. There are many experiments performed in this section, hence a short summary includes key findings of this work.

Videos for all the following experiments can be found here[3].

5.2.2 Characterising the Procession Rule

Initial characterisation experiments were performed to examine the behaviour of agents for changes in distance weighting variables. Simulation results of a parallel and anti-parallel run can be seen in Figure 5.10.

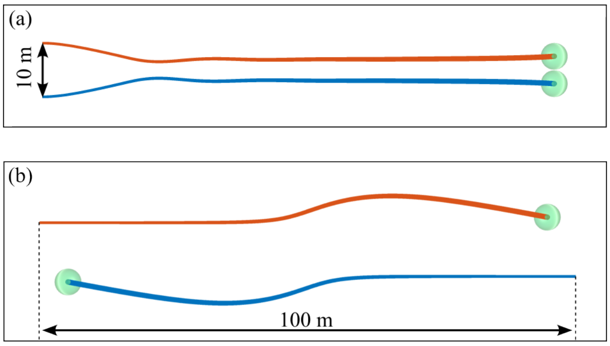


Figure 5.10. Motion of agents using procession rule. (a) Agents travelling in parallel directions are attracted. (b) Agents travelling in antiparallel directions are repelled.

[3] Emergent Aerial Routing Network Video - https://youtu.be/esYjUZKxxjg

Figure 5.11 shows the changes in the paths of the agents for various a, b and $K^{\mathcal{P}}$. This is shown using the anti-parallel case. The parallel cases did not have significant differences between runs. It can be seen that both a and b change the point at which the agent begins to deviate from the straight line path, with smaller a and larger b causing the agents to begin deviating earlier.

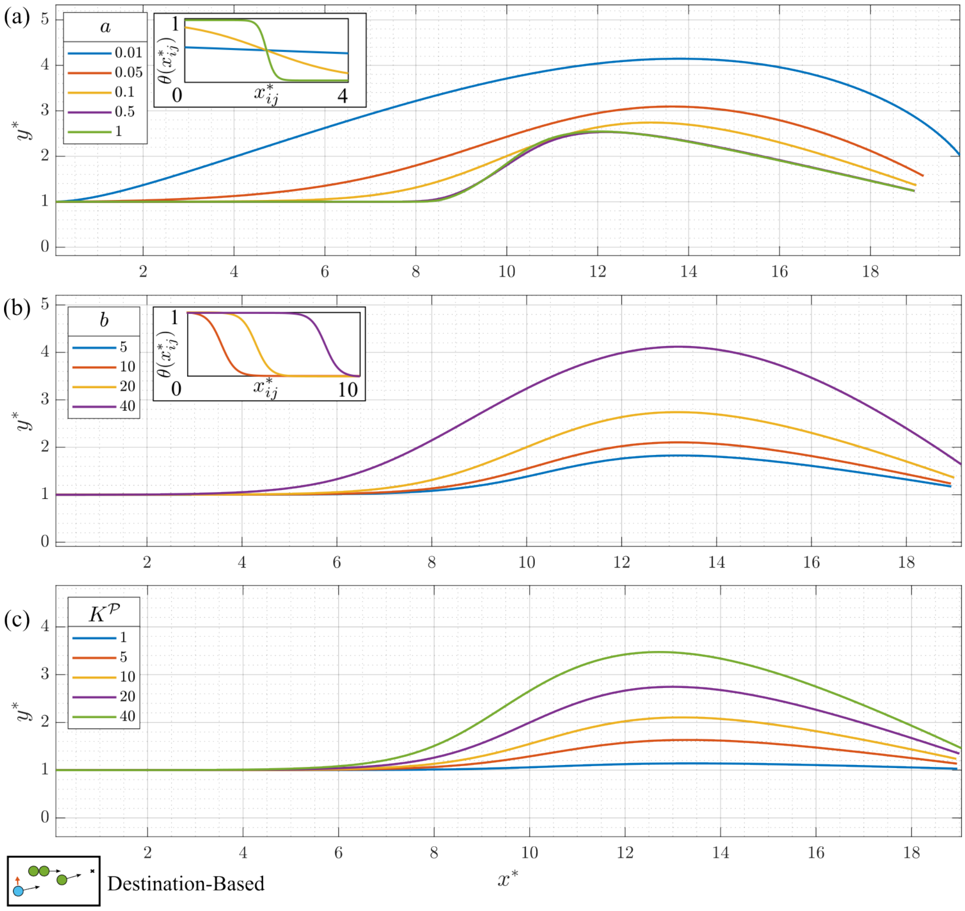


Figure 5.11. Paths of an agent using the procession rule in an anti-parallel experiment, identical to Figure 5.10 (b). (a) Paths of agent with different gradient coefficient of the sigmoid function, a. Inset shows response of sigmoid function for changing a. (b) Similar paths for different distance coefficient of sigmoid function, b. (c) Paths of agents for different procession rule gain. Unless being varied, the variables remain constant with, $a = 0.1$, $b = 10$, $K^{\mathcal{P}} = 10$. All experiments are of an agent travelling anti-parallel to another agent using destination-based implementation. Spacial dimensions are nondimensionalised by the turning radius of an agent.

5.2.3 Characterising Lane Structure in Uni Direction Flow Test Case

Experiments were carried out with all agents travelling in the same direction. The size of the environment had a significant impact on the structure of the lanes; hence the size was increased until independence was reached. This was achieved when the inflow was approximately three times larger than b. This is due to the fact that b is the distance at which the procession rule is

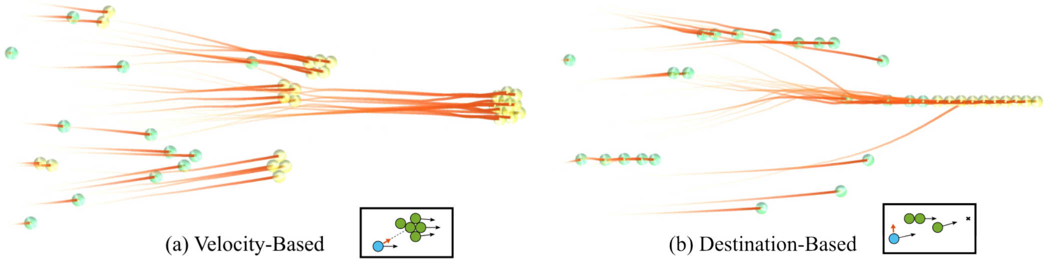


Figure 5.12. Emergent Lane structures formed in uni flow test case using two different implementations. (a) presents clumping formed due to attraction to agents positions. (b) presents lane formation similar to the desired structure.

effective. This study is purely qualitative to compare the lane structures of different implementations. All experiments were performed with variables found in Table 4.2.

Figure 5.12 presents results for velocity-based and destination-based implementations of the procession rule. The velocity-based implementation has an output direction towards other agents positions, which causes clumping effects. This is not desirable because the shape has a large cross-sectional area and causes agents to accelerate and slow down unnecessarily. In contrast, the destination-based implementation forms lanes similar to the desired lanes.

5.2.4 Characterising Distance Weighting Component in 2D Contra Flow Test Case

A multidimensional analysis was performed for a 2D contra flow test case to find the variables that produce the lowest collision rate in emergent aerial routing networks. The simulation results of a contra flow test case can be seen in Figure 5.13. It was found that as a, b, and $K^{\mathcal{P}}$ were all increased, the collision rate reduced, however, with diminishing returns, as seen in Figure 5.14. For each of the surfaces found in Figure 5.14 (a), the minimum and mean were found and

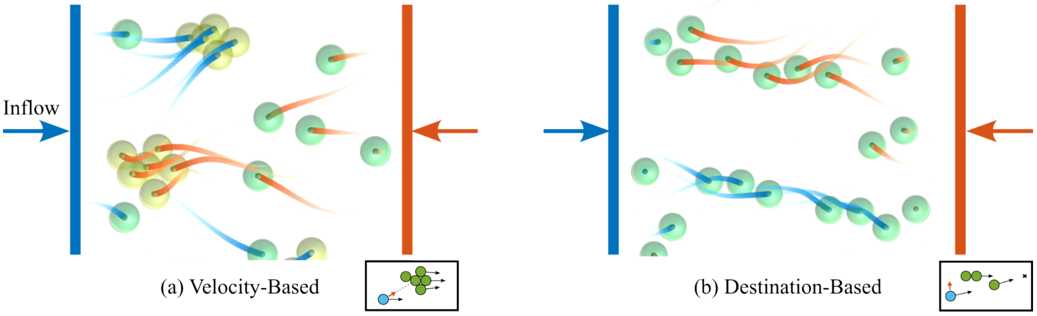
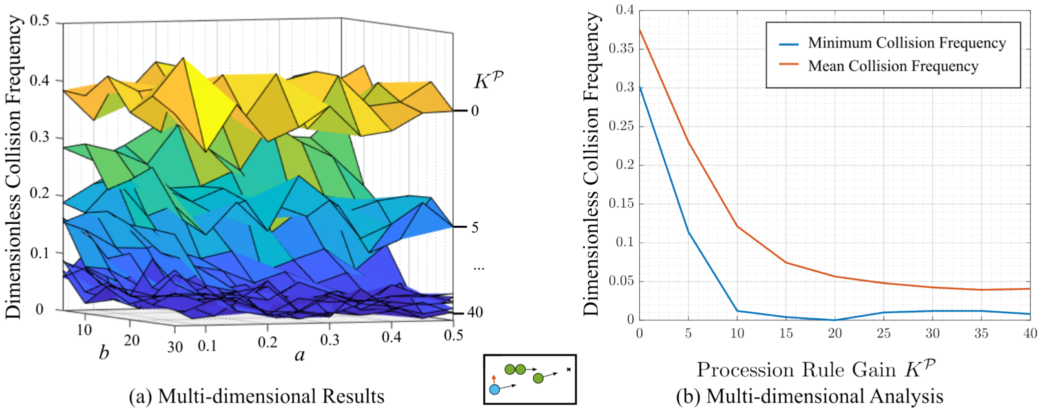


Figure 5.13. Simulation results of emergent lane formation in contraflow test case.



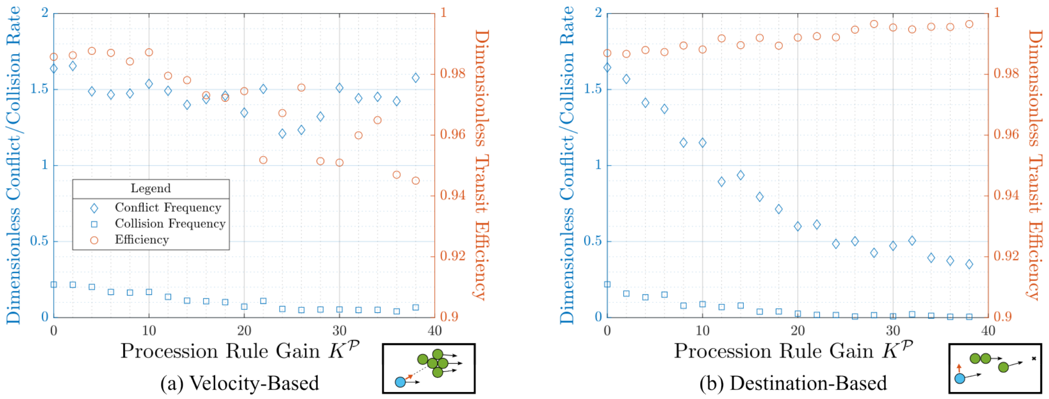
(a) Multi-dimensional Results

(b) Multi-dimensional Analysis

Figure 5.14. Multidimensional experiments of destination-based implementation in 2D contra flow test case. (a) Surfaces with various values of a and b. Each surface has a unique procession rule gain, $K^{\mathcal{P}}$. (b) The minimum and mean of each surface is found to show the diminishing returns for increasing $K^{\mathcal{P}}$. Each experiment was run for at least 1000 agents.

presented in Figure 5.14 (b). This figure particularly illustrates the diminishing returns in the dimensionless collision rate for increasing $K^{\mathcal{P}}$. A similar result is obtained for both a and b. From these figures, the values that will be used in future experiments will be $a = 0.25$, $b = 15$, $K^{\mathcal{P}} = 20$. It is noted that b, although dimensionless already, can be considered the radius of influence, and can be normalised by the turning radius of an agent. Thus, $b/l_c \approx 3$.

It is recognised that any change in the test case, the implementation, or any variable in Table 4.2 will yield different results; however, these variables will be a good starting point for most similar cases.



(a) Velocity-Based

(b) Destination-Based

Figure 5.15. Emergent aerial routing networks in 2D contra flow test case. Measuring dimensionless collision/conflict frequency and transit efficiency against procession rule gain for two implementations.

5.2.5 Comparing Implementations in 2D Contra Flow Test Case

Figure 5.15 shows the results for velocity-based and destination-based implementation in the contra flow test case. First, there is a decrease in transit efficiency with increasing procession rule gain for the velocity-based implementation, whereas, for destination-based implementations, the transit efficiency increases with increasing rule gain. In the velocity-based implementation, this is due to agents slowing down to wait for agents behind to join the ‘clump’, Figure 5.13 (a). Furthermore, as the clumps collide, this causes a significant reduction in transit efficiency. In the destination-based implementation, the reduction in collisions with the opposing flow causes an increase in transit efficiency.

Next, the conflict rate in the velocity-based implementation remains constant with procession rule gain, whereas, it decreases in the destination-based implementation. For the velocity-based implementations, the reduction in conflicts with the opposing flow is countered by the increase in conflicts due to the clumping effects. For destination-based implementation, when lanes are formed, the number of conflicts with agents travelling in similar directions is minimal due to agents forming a linear formation. Conflicts are generally due to agents in front or behind, Figure 5.13 (b). It should be noted that although these conflicts from in front and behind are counted as ‘conflicts’, the risk of collision between these agents is negligible.

Finally, the collision rate for both implementations shows a reduction with increase in rule gain. It should be noted, however, that the destination-based implementation approaches zero sooner and remains at a minimal value, compared to the velocity-based implementation, which has a more gradual reduction.

5.2.6 Performance of Emergent Aerial Routing Networks with Inflow Rate

When the inflow rate increases, emergent aerial routing networks can be seen to improve the performance of a UTM system. Figure 5.16. This result is significant in that it is clear that the reduction in transit efficiency and the increase in dimensionless conflict rates is gradual with increasing inflow rate. In this experiment, the change in inflow rate is directly proportional to

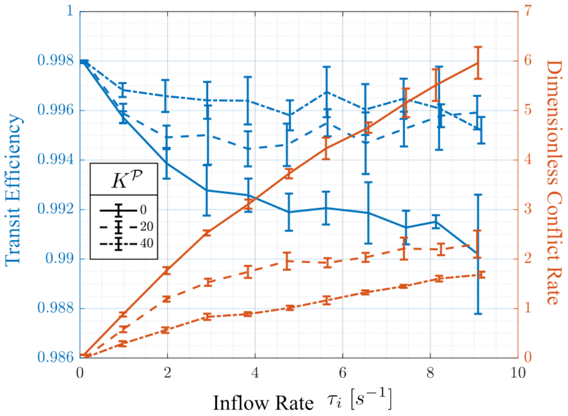


Figure 5.16. Transit efficiency and dimensionless conflict rate with changing inflow rate for various procession rule gains. When $K^{\mathcal{P}} = 0$, the dimensionless conflict rate increases while transit efficiency decreases. As $K^{\mathcal{P}}$ is increased, for high inflows, the transit efficiency remains relatively high and the dimensionless conflict rate relatively low. Experiments were performed in a 2D contra flow test case with $L_{CF}/l_c = W_{CF}/l_c = 20$. Each run was performed 5 times and the standard deviation is presented as error bars.

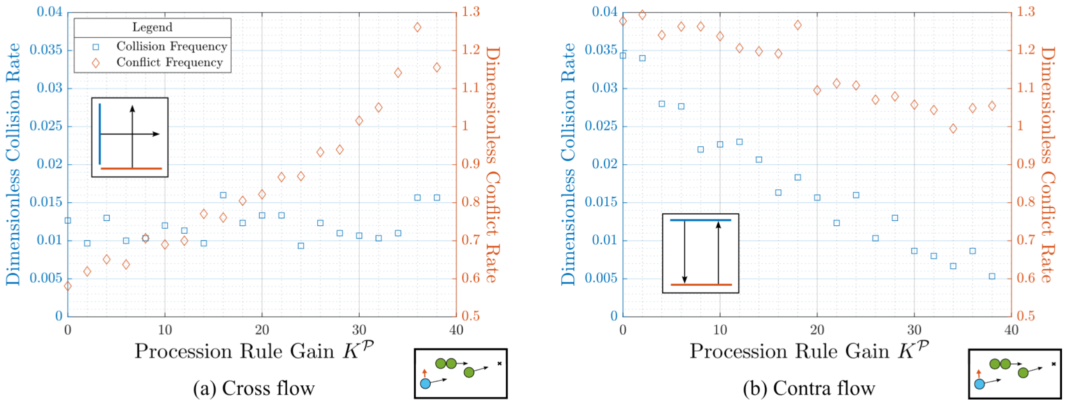
the density in the environment. This means that there is no phase change at a critical density, as exhibited in classic traffic theory.

5.2.7 Performance of Destination-Based Implementation in 3D Test Cases

Experiments were carried out in 3D test cases using the destination-based implementation of the procession rule. The first was the 90° cross flow test case. Figure 5.17 (a) shows that for an increase in procession rule gain, there is an increase in conflicts, but no significant change in the collision rate. This is consistent with expectations, as the procession rule has no effect on agents crossing at 90°. Recall that the procession rule attracts agents travelling in a similar direction and repels agents travelling in opposing directions, but less in between.

The results of the contra flow test case in Figure 5.17 (b) show a decrease in the dimensionless conflict and collision rate with increasing rule gain. This is consistent with the 2D contra flow results.

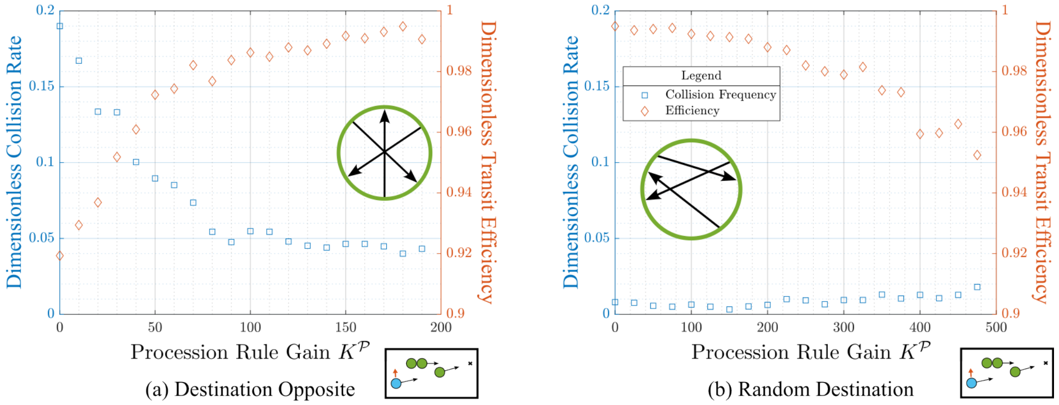
The final experiments were also carried out in uniform random test cases. Figure 5.18 (a) shows that in the opposite destination test case there is a decrease in collisions and an increase in dimensionless transit efficiency with an increase in procession rule gain. This is consistent with expectations, as when lanes are formed, agents no longer merge into the centre of the test region and instead form lanes that avoid each other. This improves transit efficiency, as well as



(a) Cross flow

(b) Contra flow

Figure 5.17. Dimensionless collision and conflict rate for 3D cross flow (90°) and contra flow (180°) test cases. This experiment used the destination-based implementation. The variables for the distance weighting were $a = 0.25$ and $b = 15$. Inflow rate was 5 s^{-1}. Transit efficiency was not affected significantly in any of these experiments.



(a) Destination Opposite

(b) Random Destination

Figure 5.18. Dimensionless collision and conflict rate for a uniform random test case. (a) Results for the destination opposite test case show a reduction in dimensionless collision rate and an increase in transit efficiency with increasing procession rule gain. (b) Result for the uniform random destination test case show insignificant change in collision rate and reduction in transit efficiency with increasing procession rule gain. The destination-based implementation of the procession rule is used with $a = 0.25$ and $b = 15$.

reduces the dimensionless collision rate.

Figure 5.18 (b) shows that in the random destination test case there is minimal change in the dependent variables when $K^{\mathcal{P}} \in [0\ 200]$, the range tested in Figure 5.18 (a). For this reason, the range was increased to $K^{\mathcal{P}} = 500$. It can be seen in this region that the transit efficiency reduces with the procession rule gain, while the dimensionless collision rate remains relatively low. This is due to the fact that agents are increasingly encouraged to form lanes, even at the cost of transit efficiency.

In uniform random simulations, very few lanes are formed, and generally only 2/3 agents join to form a small lane, see Figure 5.19. This is likely due to the fact that there is minimal

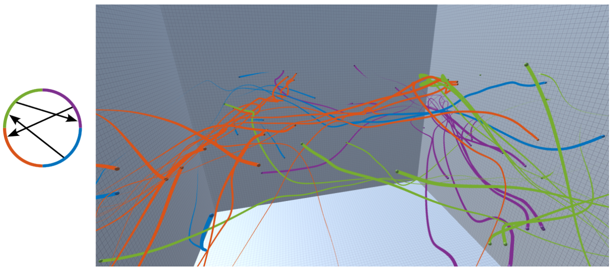


Figure 5.19. Small lanes forming in random destination test case due to minimal ‘bulk’ flow. Agent colours represent the quadrant from which they flowed in from, see inset. Procession rule gain is 100, with $a = 0.25$ and $b = 15$. Diff

‘bulk’ flow which reduces the effect of the procession rule. To increase the length of the lanes, a higher inflow rate is required to increase the density. Furthermore, a large test region is required to allow agents to form lanes. Combining a high inflow rate and a large test region means simulating a large number of agents. The time complexity of these experiments is $\mathcal{O}(n^2)$ where n is the number of agents in the test region. This makes experiments with large numbers of agents take a considerable amount of time to run. For this reason, experiments are limited to a certain size and inflow rate to allow for a high enough density while giving agents enough time to form lanes.

5.2.8 Summary

Emergent aerial routing networks are a useful method to increase transit efficiency and reduce dimensionless collision and conflict rates, with some caveats. First, the rule design must be tested to ensure it performs the desired behaviour. In this case, the velocity-based implementation partially achieved the expected behaviour; however, it did not perform as well as the destination-based implementation. Next, the type of flow has a strong effect on the performance of an emergent aerial routing network. Flows with bulk flows perform better, with the exception of 90° cross flow. Finally, emergent aerial routing networks have a stronger effect with a greater rule gain and inflow rate. This is expected as for greater rule gains and the number of agents, lanes form more readily.

5.3 Comparing Collision Avoidance Methods

5.3.1 Introduction

In this section, the performance of two collision avoidance methods is evaluated. Initial experiments characterise both collision avoidance methods in two agent interactions. The following experiments tested the methods in the uniform random test cases. The performance of each method is compared using dimensionless collision rate and transit efficiency. It was found that simple separation performs better than orthogonal separation when considering collision rates. In contrast, orthogonal separation performs better than simple separation when considering transit efficiency.

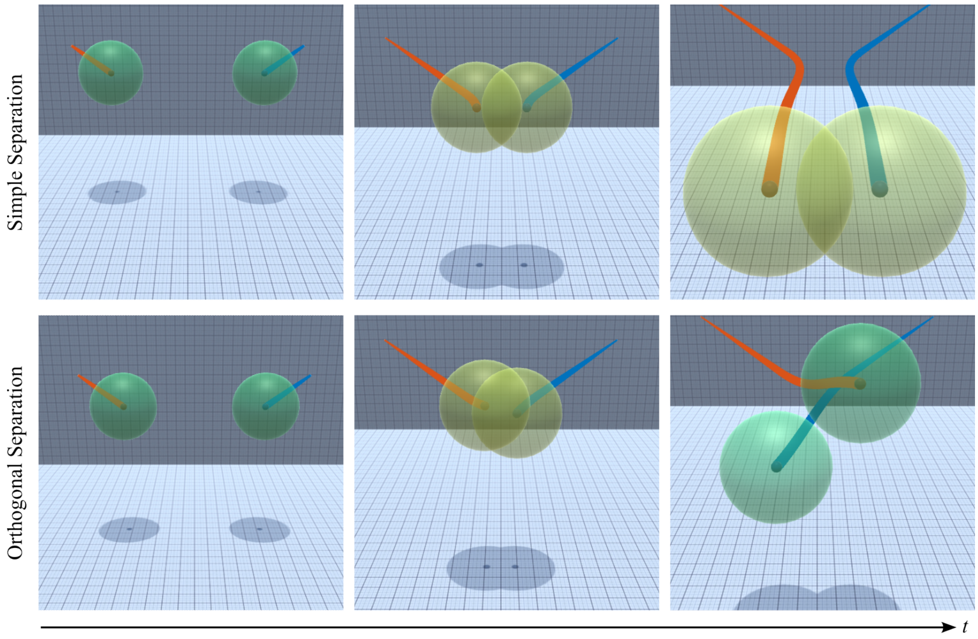
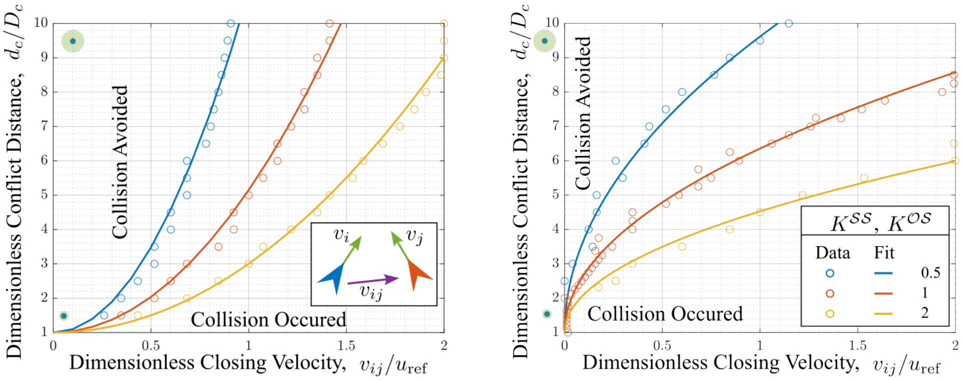


Figure 5.20. Simulation results from simple separation and orthogonal separation. Two agents crossing with an angle of 45° with DCD of 10.

5.3.2 Characteristics of Collision Avoidance Methods in Two Agent Experiments

Figure 5.20 shows the simulation results for both separation methods. The simple separation method shows agents repelling each other and moving off together. This behaviour is uncommon in uniform random experiments, as agents rarely converge with perfect symmetry. However, this behaviour is not uncommon when there is symmetry in agent motion, for example, in junctions,



(a) Simple Separation

(b) Orthogonal Separation

Figure 5.21. Safety envelope based on closing velocity and dimensionless conflict distance for simple separation and orthogonal separation. The scatter plot (circles) indicate the point at which collisions no longer occur. Lines are second order polynomial fits. Inset shows the experimental set up of agents travelling with given velocities with a crossing angle. Closing velocity was changed with crossing angle.

and hence is a valid flaw to emphasise. The orthogonal separation method shows the agent using the vertical axis to separate and pass over each other.

Figure 5.21 shows the performance of both methods based on the ability to avoid a collision. For a given rule gain, if the closing velocity and DCD are on the upper left of the plot, the collision avoidance method successfully avoids the collision. If it is on the lower right, a collision occurs. The trend that both methods exhibit is due to the rule magnitude and rule direction.

Analysing the results of this figure, for small closing velocities, simple separation requires a smaller DCD to avoid collision than orthogonal separation. In contrast, for large closing velocities, orthogonal separation requires a smaller DCD.

In this work, agents are able to move in all directions at maximum acceleration. The result of this is that a spherical conflict volume is fit for purpose. However, in real world applications, vehicles tend to traverse horizontal to the earth surface more often than climbing and descending, hence a more appropriate shape might be a disk.

It is possible to reason more advance shapes when considering the requirements for a conflict volume. When considering the shape of the conflict volume, a main concern is the error in the position and velocity of an agent and the frequency in which an agent communicates with other agents. Due to this, the conflict volume should reflect this. Figure 5.22 shows an agent (blue)

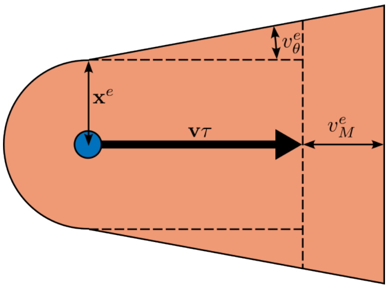
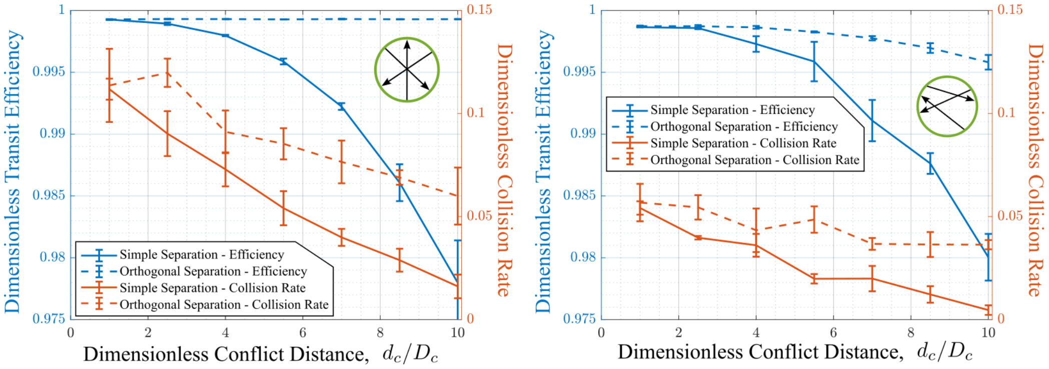


Figure 5.22. Shape of conflict volume when considering position and velocity errors.



(a) Destination Opposite

(b) Random Destination

Figure 5.23. Dimensionless collision rate and efficiency for both collision avoidance methods in destination opposite and random destination test cases. Gain for both rules was 1.

surrounded by a sphere of radius $\mathbf{x}^e$ which considers the position error of the agent. The velocity of the agent follows the $v\tau$ arrow. The size of the arrow is defined by the velocity of the agent, multiplied by the period in which agents communicate, τ. The volume extends further due to the potential error in velocity magnitude v_M^e. Furthermore, a conical shape is given to the volume with angle v_θ^e due to the directional error of the velocity.

5.3.3 Performance of Collision Avoidance Methods in Uniform Random Test Cases

Collision avoidance methods were tested in both uniform random test cases, Figure 5.23. In both test cases, it is clear that orthogonal separation allows for greater efficiency than simple separation; however, simple separation performs better at avoiding collisions for a given DCD.

5.4 Measuring Congestion

5.4.1 Introduction

In this section, the performance of different congestion metrics is evaluated, followed by some example use cases. When comparing the three methods for measuring congestion, the expected collision rate method, derived from gas kinematics, shows the strongest correlation to the actual collision rate. The expected collision rate is then used to perform field measurements of congestion in the cross flow and uniform flow test cases. The results show high congestion when the cross flow test case has a high crossing angle, as expected. Similarly with the uniform random test case, the destination opposite exhibits considerably larger congestion than the random destination, also as expected. Finally, congestion was measured in a procedurally generated city. The results effectively identified regions of high congestion.

A video of the results can be found here[4].

[4] Measuring Congestion - https://youtu.be/3Ii8ilHKseU

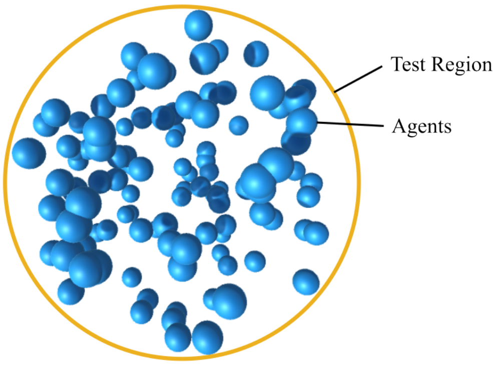


Figure 5.24. Simulation result of agents in the spherical test region.

5.4.2 Comparing the Performance of Different Methods of Measuring Congestion

The three methods of measuring congestion are compared in a standard test environment. A screenshot of the simulation environment is presented in Figure 5.24.The first method tested is the traffic pressure using the velocity direction variance. In the initial experiments, the collision rate was varied by changing the inflow direction variance and the inflow rate. Figure 5.25 shows the relationship between traffic pressure and collision rate. It is clear that there is a relationship between the two measures; however, it is not linear.

The following experiments varied the speed of the agents, as seen in Figure 5.26. It is clear that this congestion construction is not robust against the change in speed. This is due to the fact that the velocity direction variance only considers the direction. The next method, generalised variance, does, however, consider both variance and magnitude.

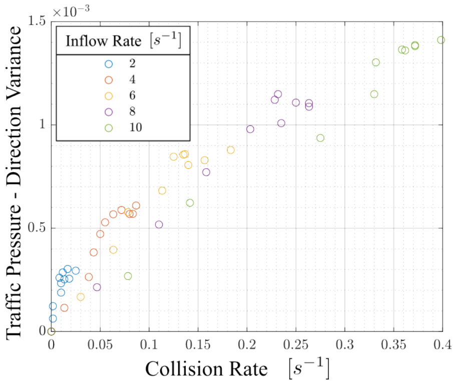


Figure 5.25. Comparing collision rate of agents with traffic pressure using the **velocity direction variance**. The inflow rate and inflow direction variance were $\tau_i \in [2\ 10]$ and $\sigma(\theta) \in [0.2\ 1.4]$, respectively. It is clear some relationship exists between the collision rate and traffic pressure, however further experimentation is required. Each experiment was run for a minimum of 1000 agents.

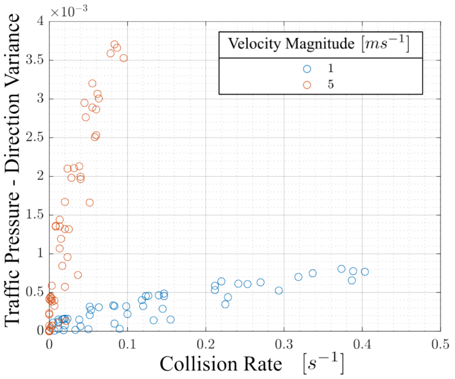


Figure 5.26. Similar to Figure 5.25 however changing the speed at which agents travel. It is clear that this construction of traffic pressure is unable to account for velocity direction variance and magnitude. The inflow rate and inflow direction variance were $\tau_i \in [2\ 10]$ and $\sigma(\theta) \in [0.2\ 1.4]$, respectively. Each experiment was run for a minimum of 1000 agents.

Figure 5.27 shows the collision rate against traffic pressure using the generalised velocity variance. This method is more robust against change in speed, however, is not well defined. The final method to construct the congestion metric is to use the expected collision rate from kinetic theory. Recall that this method is modified to remove the bulk flow and measure the collision rate due to the remaining velocity. This is similar to removing the dynamic pressure and measuring the static pressure.

Figure 5.28 shows a clear linear relationship between the expected collision rate and the actual collision rate. The linear fit gradient is 1.2 ± 0.1. This is hence the best candidate for use as the congestion metric and will be used in the multi-cell measurements in the following sections.

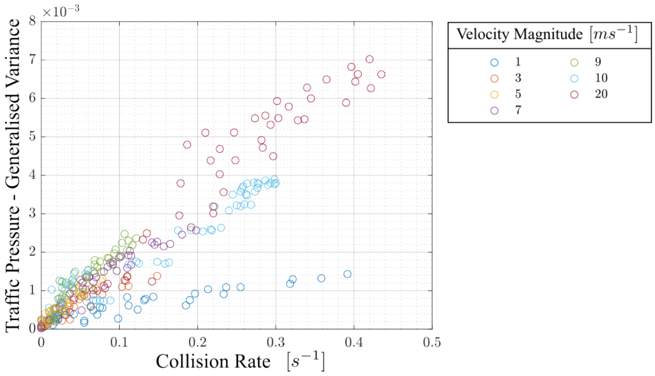


Figure 5.27. Comparing collision rate of agents with traffic pressure using the **generalised variance**. The inflow rate and inflow direction variance were $\tau_i \in [2\ 10]$ and $\sigma(\theta) \in [0.2\ 1.4]$. This method is more robust against changes in speed, however there is a significant difference in the results between the minimum and maximum values of speed. Each experiment was run for a minimum of 1000 agents.

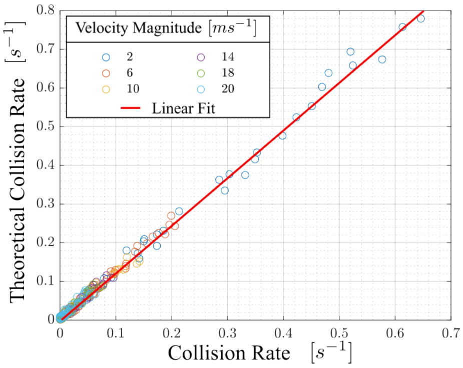
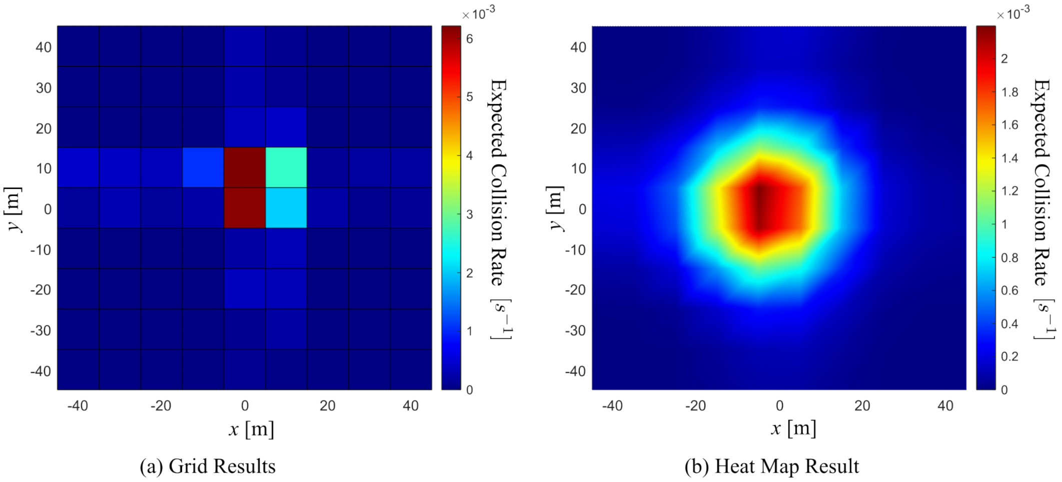


Figure 5.28. Comparing collision rate of agents with expected collision rate from kinetic theory. This method has a clear linear relationship and is robust against inflow rate, inflow direction variance and speed.

5.4.3 Multi-Cell Measurements of Congestion



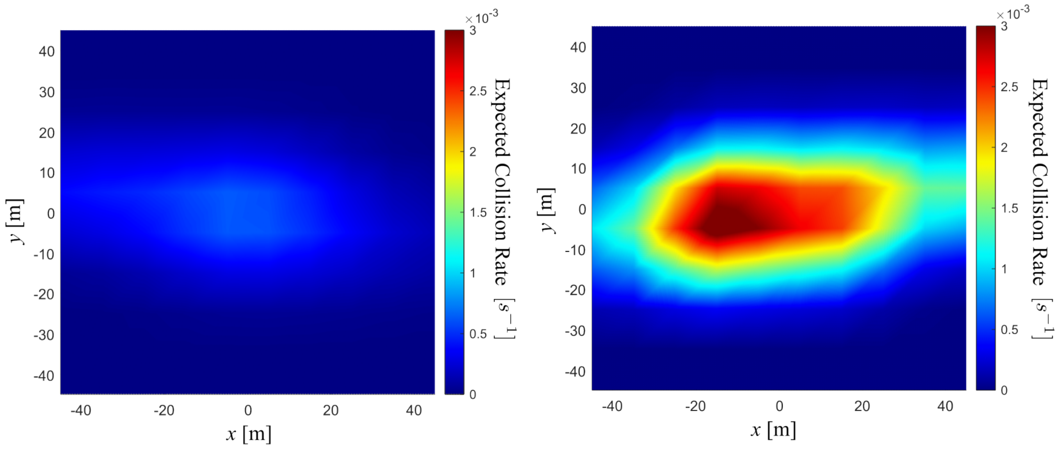
(a) Grid Results

(b) Heat Map Result

Figure 5.29. Congestion multi-cell measurements of a 90° cross flow test case. (a) Congestion in a grid format. (b) Congestion in a heat map by applying a Gaussian filter over the grid result. The test environment was divided into a 10x10 array and the congestion was averaged over all timesteps in each cell.

Congestion was measured in the flow test cases. In the 90° cross flow test case, the results are presented in both grid format and heat map format. The heat map is achieved by applying a Gaussian filter over the grid layout with a standard deviation of 1. The purpose of the visual layout is for qualitative understanding of the environment, and it is the authors opinion that the continuous heat map is more easily understood. This is due to the entire colour range presented between the lowest and highest points instead of discrete jumps. Figure 5.29 is a congestion map of the 90° cross flow test case. All variables remained the standard variables used in Table 4.2.

The experiments were then carried out in the cross flow test case with a crossing angle of

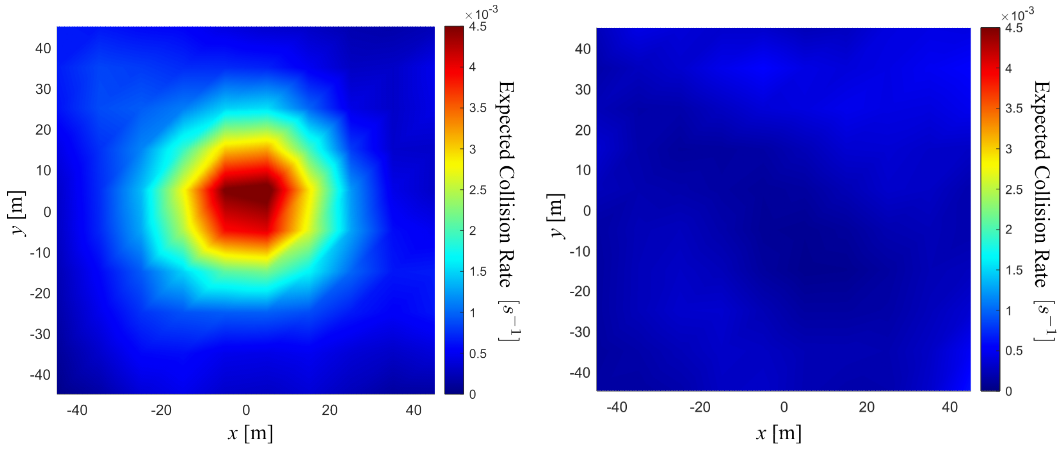


(a) 20 degrees

(b) 160 degrees

Figure 5.30. Congestion multi-cell measurements of cross flow test case with different crossing angles. (a) Crossing angle of 20° has very low congestion. (b) Crossing angle of 160° has significantly higher congestion. The test environment was divided into a 10x10 array and the congestion was averaged over all timesteps in each cell.

20° and 160°. Figure 5.30 presents the results of both test cases in the heat map format. As expected, there is significantly greater congestion in the 160° test case due to the high closing angle between the agents.



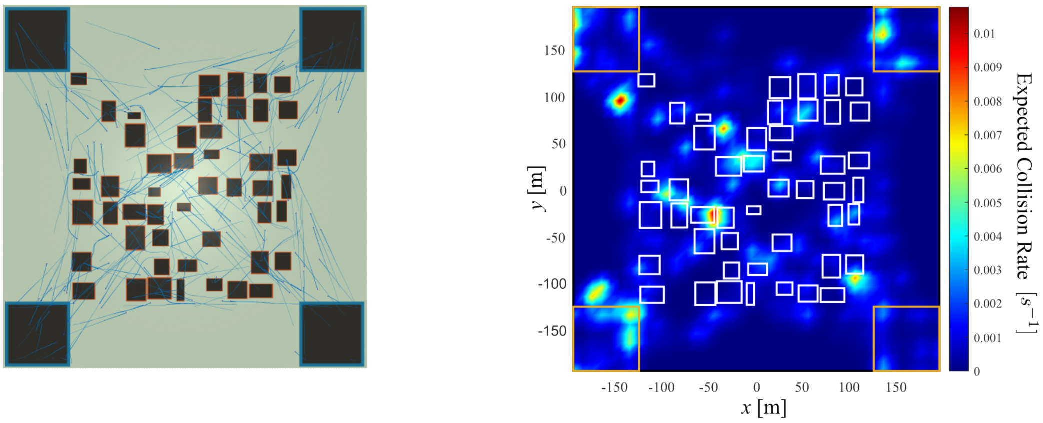
(a) Destination Opposite

(b) Random Destination

Figure 5.31. Congestion heat maps in uniform random test cases. (a) Shows the destination opposite test case with comparatively high congestion. (b) Shows the random destination test case with comparatively low congestion. The test environment was divided into a 10x10 array and the congestion was averaged over all timesteps in each cell.

Finally, experiments were performed in uniform random test cases. Figure 5.31 (a) shows the opposite destination test case with very high congestion, as expected. Figure 5.31 (b) shows the random destination test case with a very low congestion compared to other tests.

Figure 5.32 (a) shows the simulation results of the experiment with four distribution centres located in the corners of the city. The procedurally generated city uses a high-rise profile of a



(a) Simulation Result

(b) Congestion Measurement

Figure 5.32. Measuring congestion in an entire city. (a) Shows the simulation result with the procedurally generated city in the center and the 4 distribution centres on each corner. (b) Shows the congestion heat map after running the experiment for approximately 10 000 agents. Buildings are outlined in white and distribution centres in yellow. The test environment was divided into a 40x40 array and the congestion was averaged over all timesteps in each cell.

city with a building coverage ratio in the range of 0.3 to 0.6 and a floor area ratio in the range of 1 to 5. Figure 5.32 (b) shows the congestion in the city with a silhouette of each building in white and the distribution centres in yellow. This method has effectively identified several regions of high congestion.

Chapter 6

Practical Implementation

Real-world experimentation was performed to implement the theoretical principles within a test environment with realistic communication delays, position and velocity errors, and environmental disturbances, such as wind. The aim was to show that it is possible to implement the theoretical principles with present-day technology. In the following, the technologies used and the software developed to perform experiments are discussed followed by results from several experiments performed in the Snowdonia Aerospace Centre in Llanbedr, Wales. The framework constructed to perform the practical experiments was known as the Helix framework and had contributions from two research assistants. Within this project, I worked as the principal investigator.

6.1 System Overview

6.1.1 System Requirements

The system consists of two parts, the agents and the ground control station.

The requirements for an agent to perform an experiment are the following.

1. The ability to move freely in a 3D environment and hover.
2. Have a flight controller that can take velocity commands and actuate the platform accordingly.
3. Carry an onboard computer for local computing of velocity commands.
4. Capability to communicate with a ground control station and other agents.
5. Carry sensors required for position and velocity estimation.

Additional requirements were identified to ensure that the experiments were practical and safe. First, triple communication redundancy should be used to ensure consistent data links for control and navigation. Second, the agents should be small enough to carry multiple agents to the experimentation sites.

The requirements for the ground control station (GCS) are as follows.

1. Ability to command all agents to perform experiments.
2. Ability to command agents individually in emergency cases.
3. Give an accurate representation of the current system state.
4. Communicate with ranges of at least $100\ m$.

From these requirements, the following subsystems were used. An overall system diagram is presented in Figure 6.1.

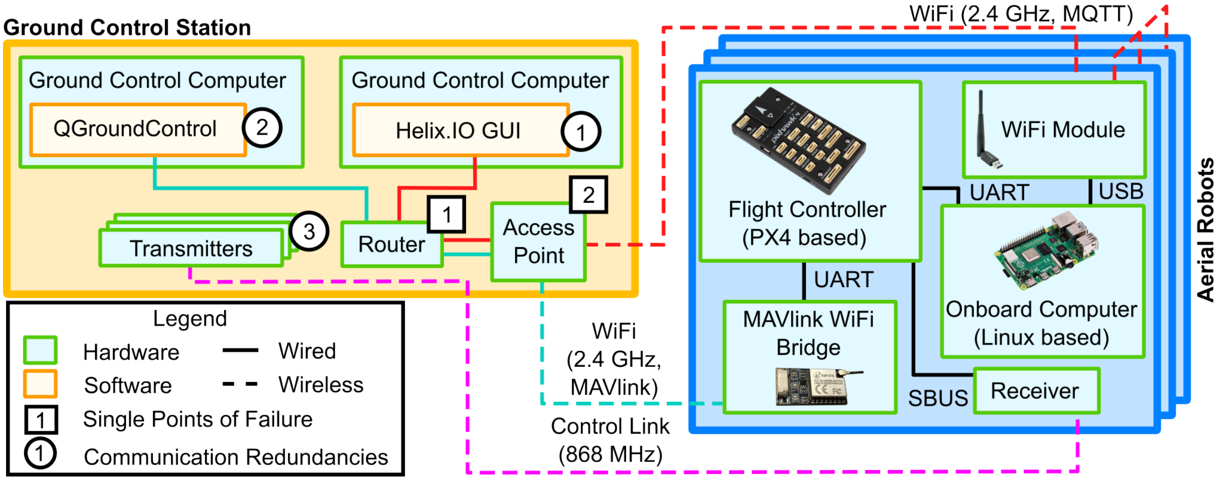


Figure 6.1. Systems diagram of practical demonstrations. Single points of failure within communication network are identified with mitigating redundancies.

6.1.2 Onboard Computation

To make use of the Helix framework, each aerial robot must be equipped with a Linux-based computer. There are two separate components to the software on the onboard computer. The first is the core component, which handles communications with other aerial robots and the Ground Control Station (GCS), as well as sending commands to the flight controller, such as take-off and return to launch. The second is the customisable experiment component, where the high-level control algorithm which forms the basis of the multi-agent experiment is implemented.

The software running on the onboard computer is written in Python. MAVSDK is used to send commands from the onboard computer to the flight controller over a Universal Asynchronous Receiver/Transmitter (UART) connection using the MAVlink protocol.

The control and telemetry available to aerial robots are available through the MAVSDK library. An example of commands available to control an aerial robot includes sending velocity and position commands. An example of telemetry streams available include position, velocity, and battery level of all aerial robots in the system.

The Helix framework provides a software upload tool enabling the customisable experiment component to be transferred to all the aerial robots in the experiment at once over WiFi. Once the customised code is uploaded to each aerial robot, it can be triggered from the ground control station.

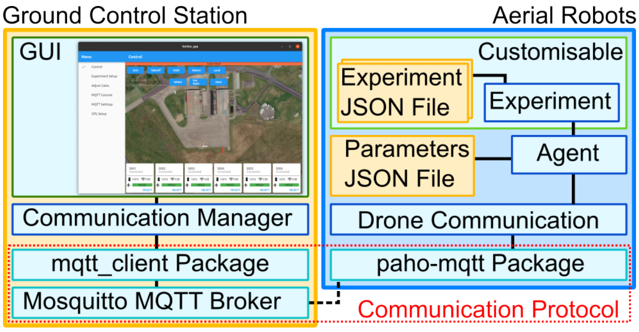


Figure 6.2. Communication systems diagram between GCS and aerial robots. The communication protocol is modular and can be interchanged with another communication protocol.

6.1.3 Communication

MQTT is used for the primary communication link between the Ground Control Computer (GCC) and the onboard computer of each aerial robot. Figure 6.2 shows the communication flow in the framework. MQTT is a low-bandwidth publish/subscribe network protocol. Within the current implementation, communications are centralised through a broker running on the GCC. Each aerial robot connects and communicates with the GCC using local MQTT packages.

In addition to the communication protocol within the framework, each agent is also equipped with a MAVLink WiFi Bridge that provides a secondary telemetry link to a mission control program, such as QGroundControl. It is recommended to use a second GCC to run the mission control program, however, for convenience, it can be run on the same computer as the Helix GUI. Finally, an 868 MHz control link is recommended to provide triple communication redundancy.

Although the communication architecture presented here is centralised with all messages passing through a broker running on the GCC, the proposed framework has been designed so that a mesh network could be implemented in the future for distributed communications.

6.1.4 Human Interface

The Helix framework offers a GUI that runs on a GCC and allows real-time monitoring of the current state of each aerial robot, including position, battery state of charge, Received Signal Strength Indicator (RSSI), and flight mode.

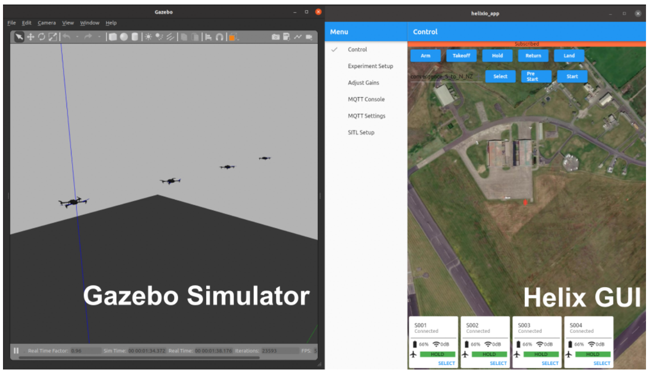


Figure 6.3. SITL experiment running simulation of 4 aerial robots.

The GUI is built using Flutter, a user interface software that allows the development of a single Dart-written code base to be deployed on any desktop, iOS, and Android devices. This enables the development of a Linux-native and Android version of the GUI to provide more flexibility when controlling experiments.

The safety features included in the GUI show when aerial robots are within a critical distance from each other. If the distance between the aerial robots continues to decrease below a predefined minimum distance, all aerial vehicles are automatically commanded to hold position. The values of these distances can be changed depending on the vehicle size and speeds.

6.1.5 Software In The Loop

SITL simulation is provided by PX4 based on the Gazebo robot simulator. The Helix framework includes a tool to launch a SITL simulation with a specified number of aerial robots. The tool launches the specified number of instances of the Helix onboard program, connects them with PX4 Firmware instances using different ports, and spawns the specified number of aerial robots in Gazebo. An example of a SITL experiment being run is in Figure 6.3.

6.1.6 Mission Flow

The Helix framework provides a mission flow to carry out MARS experiments safely. The standard mission flow is as follows: aerial robots are armed, take-off, initialised for experiment, perform experiment, return to launch, and land. Each of these processes are described below.

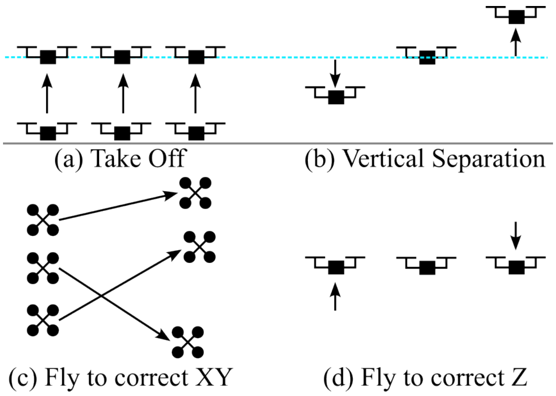


Figure 6.4. Experiment initialisation strategy for deconfliction.

To initiate the flight, an arm command is sent to the aerial robots. Once all aerial robots are armed, the take-off command can be sent. Optionally, a user can have each aerial robot travel to a predefined location in the experiment initialisation phase.

As shown in Figure 6.4, when performing this relocation, the aerial robots first separate vertically by a specified distance. Secondly, aerial robots travel to the specified XY position. Finally, when all aerial robots are at the correct XY position, the aerial robots move to the specified altitude. This method safely deconflicts the travel of aerial robots prior to an experiment. It is important to note that aerial robots cannot be given initialisation positions with the same XY coordinates.

The predefined initialisation positions are defined by the user in a JavaScript Object Notation (JSON) experiment definition file along with other specified parameters to be used by the custom code during the experiment.

Once the experiment has been initialised, the command to start the experiment can be sent. It is important to note that the proposed framework assumes that the aerial robots are flying throughout the experiment, currently making it unsuitable for experiments that require aerial robots to take-off and land.

When the experiment has finished, a return command can be issued. The aerial robots are sorted by altitude and then vertically separated in the same order. Once each aerial robot has returned to its home location, the land command is issued, and the agents land at their take-off position before disarming.

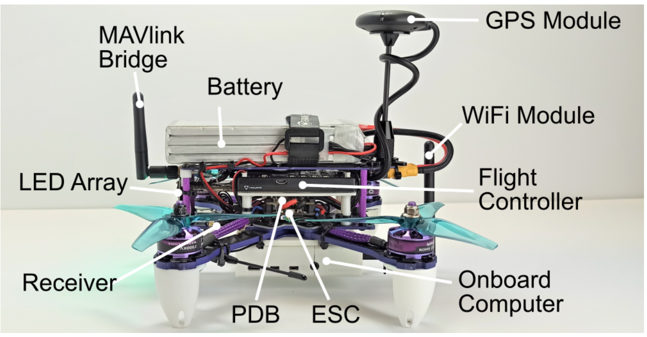


Figure 6.5. Hardware used for implementing framework in case study experiments.

6.2 Hardware Implementation

The MARS consisted of off-the-shelf PX4 Vision Autonomy Development Kits along with custom built quadcopters, requiring only minor modifications to the JSON parameter file for the different platforms. For custom built quadcopters, the frame selected was a small racing type drone designed for 5-inch diameter propellers. The aerial robots were equipped with a Pixhawk 4 flight controller and a Raspberry Pi 4 onboard computer. The MAVlink WiFi bridge used was a Beyond Robotix Kahuna, and the WiFi module was a Dynamode USB WiFi Adapter. The receiver used was a Crossfire Nano that was bound to a unique transmitter with a Crossfire transmitter module. A visual representation of the agents is shown in Figure 6.5.

The two GCCs were Linux PCs with i7 processors and 8GB RAM, one running the latest QGroundControl software and the other running the Helix.IO GUI. The router was a TP-LINK Archer AX90 connected to a Ubiquiti M2 Nanostation Access Point (AP). This AP was used because it was the required 2.4 GHz WiFi band and is directional, giving improved range performance.

6.3 Results

The performance of the framework is evaluated in two ways, the performance of the system during the entire mission flow and the performance of the framework during the case study experiments.

Table 6.1. Junction experiments variables in practical helical convergence and divergence experiments.

Variable	Value	Unit	Variable	Value	Unit
Reference Speed, u_{ref}	5	ms^{-1}	Junction Angle, θ_j	10	$^\circ$
Flow Rule Gain, $K^{\mathcal{F}}$	1	n.d.	Helix Angle, θ_h	45	$^\circ$
Geometry Rule Gain, $K^{\mathcal{G}}$	1	n.d.	Path Size, d_j	10	m
Rule Interval, τ_r	0.1	s	Path Length, L_j	75	m
Max Acceleration, a_{max}	5	ms^{-2}	Collision Distance, D_c	2.5	m
Acceleration Gain, K^a	1	n.d.	Conflict Distance, d_c	5	m
Separation Gain, $K^{\mathcal{SS}}$	2	n.d.			

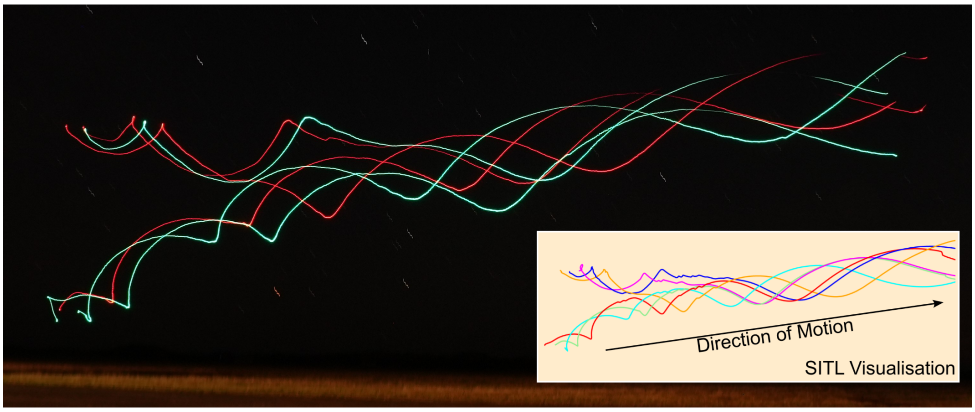


Figure 6.6. Long exposure of 6 aerial robot experiment performing counter-rotating helical converging junction. Agents started south (left of frame) and travelled north (right of frame). The inset figure obtained by the log visualiser tool shows the consistency between SITL and practical experiment. [Credit: Kieran Wood]

Table 6.2. Helix framework issue record.

Problem	Possible Reason	Solution	Outcome
Aerial robot unresponsive to all communication channels	a) Corrupted geofence causing auto return to geofence	a) Always re-upload geofence before flight/don't use flight controller geofence	No further unresponsive aerial robots
Commands not being executed by aerial robots	a) WiFi Link failed - aerial robot disconnected b) Command not received by aerial robot c) Command not sent from GCS	a) Added directional AP for better connectivity b,c) Added a check function to see if command had been actioned yet, if not send command again	Communications became more robust Reliability of agents executing commands increased significantly
Uncommanded recalibration, arm and throttle up in lab	a) Spurious PWM command from flight controller to ESC on startup b) Spurious PWM command from battery being plugged in c) Malfunctioning ESC	a) Replaced flight controller on aerial robot and changed ESC communication protocol to DSHOT b) Listened for recalibration tones on startup, if heard, unplug the battery immediately c) Replaced ESC	No further uncommanded re-calibrations, arming and throttle ups occurred
Aerial robot flip on take off	a) Wind gust as aerial robot became airborne b) One of the legs being caught in a groove	a) Reduced acceptable wind threshold for experiments b) Ensured aerial robots were not positioned in grooves between experiments	No further aerial robots flipped
Uncommanded gyration when in hold flight mode	a) Poorly calibrated compass b) Malfunctioning compass	a) Re-calibrated compass b) Replaced compass	Minimal gyration was observed
Aerial robot failed pre-flight check (Error: EKF HORIZ POS ERROR)	a) Optical flow sensor gives poor data at night	a) Turn off optical flow sensors at night	Aerial robot passed pre-flight checks

6.3.1 System Performance

To evaluate the performance of the system, a complete issue record is presented in Table 6.2 that summarises the problems faced and the solutions implemented to overcome them. Problems generally fall into two categories, hardware problems such as flight controllers and ESCs and communication problems, which were largely resolved by using a directional AP.

Communication is one of the most challenging components of the framework. When using a router without additional AP, the communication success rate was approximately 60-80%. This was due to range limitations and limitations on the number of devices that communicated with the router. When aerial robots travelled more than 150 m from the router, disconnections would occur; however, they would reconnect within a range of 100 m. Regarding the number of aerial robots, it was found that for up to 5 aerial robots, therefore 10 WiFi links to the router, communications were close to 80% successful. This was compared with using up to 10 aerial robots; therefore, with 20 WiFi links, the success rate dropped to 60%. Once the directional AP was installed, the communication success rate increased to more than 95% for all aerial robots. The only communication loss occurred when an aerial robot flew outside the effective beamwidth of the directional AP antenna.

SITL simulations were invaluable to the production of the framework as they allowed tests of the entire mission flow pre-flight. On several occasions, bugs in code caused initialisation of emergency hold procedures due to potential collisions. These bugs were identified early in SITL and corrected before any flight. The result was that there were no emergency hold procedures initialised in-flight.

6.3.2 Experiment Performance

From the initialisation of the experiment, the aerial robots travelled in a consistent manner to the SITL environment, as shown in Figure 6.7. Differences can be attributed to wind, position errors, and communication delays. This presents SITL as a useful estimator on how aerial robots will behave in a practical experiment.

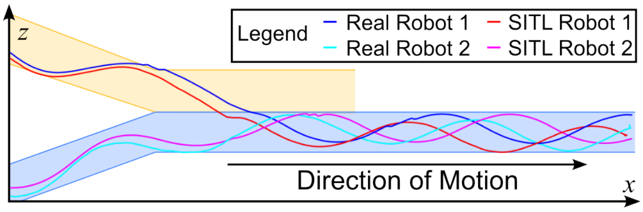


Figure 6.7. Comparison of practical experiment (Real) and SITL of converging in counter-rotating cylindrical transport surfaces by using the Multi visualiser tool. Differences between the paths are likely due to random changes in the environment, for example wind, that have caused the paths to less consistent. Jittering on the paths is caused by a deconfliction method.

A long exposure of a convergence experiment was captured in Figure 6.6. In this figure, there are two cylindrical transport surfaces converging on the vertical axis. Aerial robots on the upper cylindrical transport surface begin with a counter-clockwise helical motion from the perspective in the figure and switch to clockwise when converging onto the lower cylindrical transport surface. The inset shows the visualisation of the same experiment run in SITL and demonstrates the consistency between the two methods.

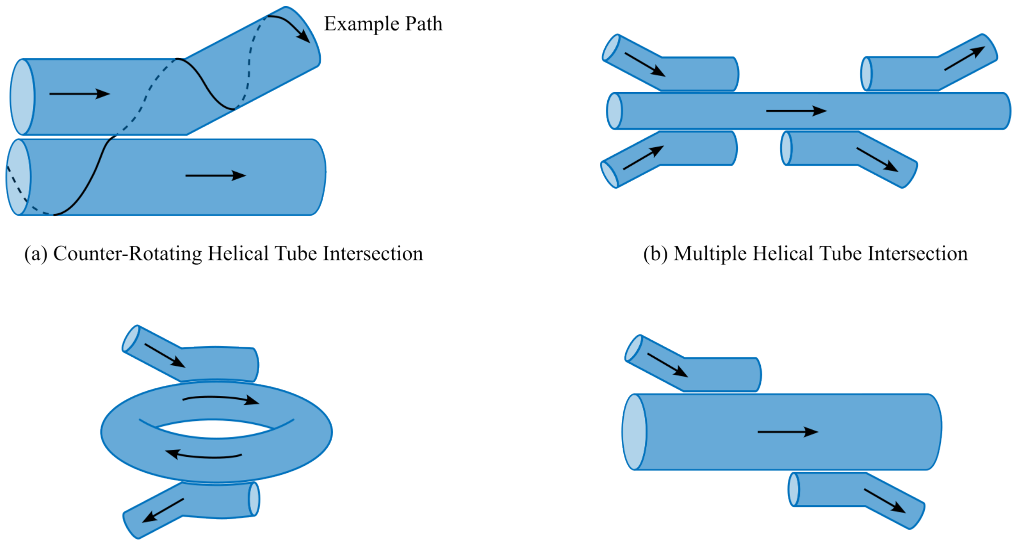
6.4 Conclusion

In this work, we propose a framework for enabling the experimentation of distributed multi-aerial robotic systems in large outdoor spaces of the order of 100 m. The implemented technologies give sufficient measurement resolution using standard open-source flight controller sensor hardware (inertial, GPS, and barometric), and acceptably robust communications between vehicles and the ground control station. The framework was designed with ease of use in mind, employing well-documented open-source programming languages, libraries, and protocols. The framework has been shown to be hardware agnostic, since it was implemented on two different hardware architectures with consistent results. Furthermore, the framework has been shown to be scalable up to an order of 10 aerial robots. However, it is hypothesised that link quality will degrade with an increasing number of aerial robots. In future work, a mesh network as the primary control link may eliminate this degradation and further make the framework distributed in both control and communication.

Chapter 7

Advanced Applications of Aerial Routing Networks

In this chapter, an overview of potential advanced applications for aerial routing networks is presented. This work has not been tested numerically, however, based on experience in the previous work, should yield favourable results. These applications are focused on the use of helical tubes due to their low conflict rates in diverging junctions and unique ability to generate multiple simultaneous converging and diverging junctions in any radial direction from the centerline. These intersections fall into three categories. Tube-tube intersections, similar to the junctions experimented with in this work, are any type of junction with counter rotating tubes. Terminating tube intersections are intersections where multiple tubes terminate and agents arrive on an inflowing tube, navigate through the intersection and outflow from an outgoing tube. Finally, Tube-Ribbon intersections allow agents to transfer from a tube to a ribbon to form more complex intersections.



(a) Counter-Rotating Helical Tube Intersection

(b) Multiple Helical Tube Intersection

(c) Circular Helical Tube Intersection

(d) Variable Scale Helical Tubes

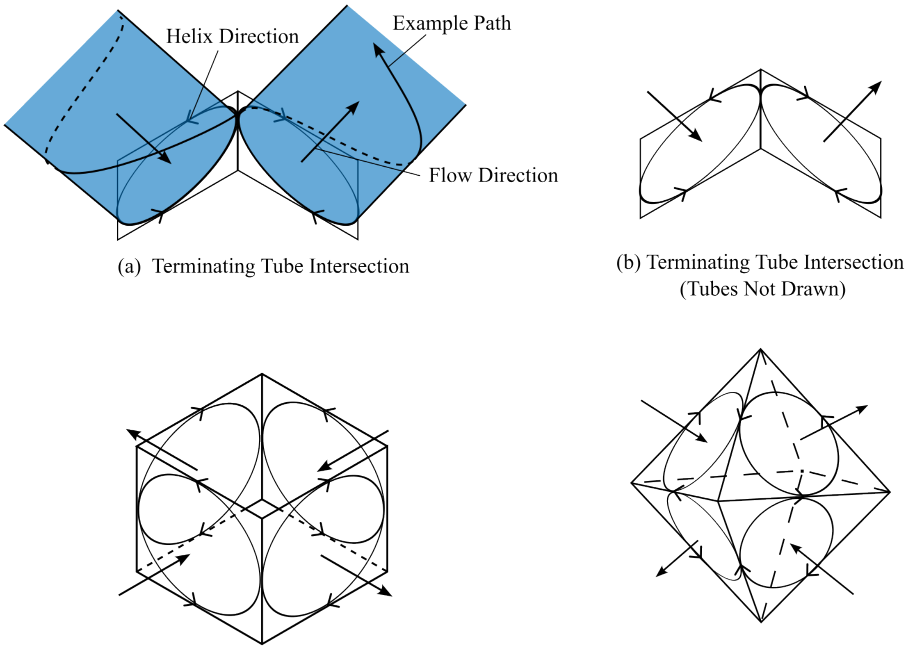
Figure 7.1. Tube-Tube Intersections

7.1 Tube-Tube Intersections

Following from the counter-rotating tube intersections found in this thesis, several other structures are proposed. Figure 7.1 (b) presents several counter-rotating helical tube intersections. The only requirement is that the centerlines are parallel for the distance required for agents to join or leave the skyway, and the distance between the centerlines causes the tubes to coincide along another line parallel between the two centerlines. This means that the centerlines can be at any angle around each other, as long as they are parallel. Figure 7.1 (c) shows tubes intersecting a circular helical tube that acts like a roundabout. This circular helical tube can have an infinite number of helical tubes converging and diverging from it, as long as the circular tube is large enough. Finally, Figure 7.1 (d) shows how tubes can be of variable sizes to accommodate more or less agents.

7.2 Terminating Tube Intersections

The following terminating tube intersections are found when the tubes terminate and form roundabouts on a surface that are interlaced with other terminating tubes. Agents arrive in an inflowing tube and navigate between the interlacing intersections until they reach the desired



(a) Terminating Tube Intersection

(b) Terminating Tube Intersection (Tubes Not Drawn)

(c) Uncapped Cube Terminating Tube Intersection

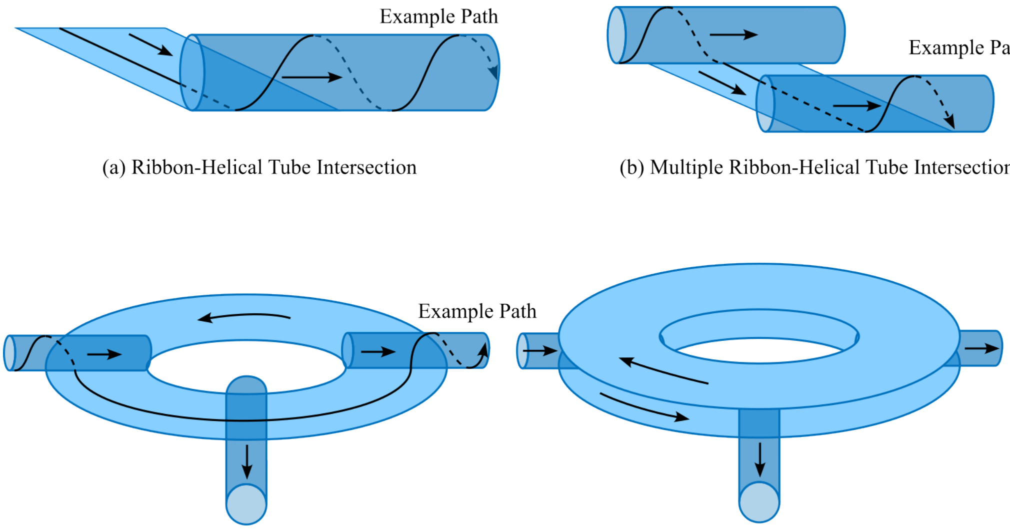
(d) Octahedron Terminating Tube Intersection (Octabout)

Figure 7.2. Tube Terminating Intersections

outflowing tube of the agent. Figure 7.2 (a) shows an example path of an agent entering from one tube and navigating to the outflowing tube on another surface. Figure 7.2 (b) shows the same as (a) with the inflowing and outflowing tubes not drawn. This is the format used in the following two figures. Figure 7.2 (c) shows the same as (b) with four interlaced tube termini on an ‘uncapped’ cube. Using 3D shapes is useful to help understand the intersection topology as each of the faces is where a tube terminates, and each of the sides of the shape can be used to describe the common direction between neighbouring tubes. Figure 7.2 (d) shows an octahedron with 8 tube termini, one on each face. Going into the geometry of using 3D shapes, the only closed 3D shapes that can be used for constructing these types of tube terminating intersections have an even number of sides joining at every corner.

7.3 Tube-Ribbon Intersections

Tube-ribbon intersections use a combination on various surfaces to allow agents to travel between tubes. Figure 7.3 (a) shows a ribbon merging with a tube. Figure 7.3 (b) shows how a surface can be used by agents to transfer from one tube to another. Figure 7.3 (c) shows a



(a) Ribbon-Helical Tube Intersection

(b) Multiple Ribbon-Helical Tube Intersection

(c) Rotating Ribbon Intersection

(d) Multiple Counter-Rotating Ribbon Intersection

Figure 7.3. Tube-Ribbon Intersections

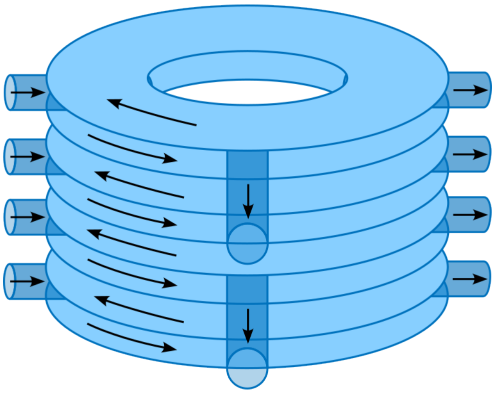


Figure 7.4. Several counter-rotating disks staked in a large tube surface intersection.

similar concept; however, the surface is a disk, and agents can enter using one tube, follow the surface until the agent reaches its desired outflowing tube. Figure 7.3 (d) shows the same as (c) with a counter-rotating disk above the tubes. These structures can be stacked vertically on top of each other to form a large tube intersection with alternating clockwise and counterclockwise disks, as shown in Figure 7.4.

7.4 Best Practices in Intersection Design

From the studies performed in this body of work, it has been found that there are several patterns found in good intersection design. Firstly, in this work it has been found that some junction types are conflict-free. These types of junctions should be used as much as possible in intersection design.

Secondly, large closing velocities between agents are the main cause of collision. Closing velocities are considered large when they are significantly higher than the reference velocity or the maximum acceleration multiplied by the computation time step.

Finally, intersections must be designed to fit the environments they are used in. An example of poor intersection design for road traffic would be putting a roundabout or traffic lights in the middle of a motorway. A good design should consider the most common traffic patterns and construct an intersection for these patterns.

Chapter 8

Conclusions and Future Work

In this chapter, conclusions and future work of this study are presented.

8.1 Conclusions

The concept of aerial routing networks is found in much of aerial traffic management literature; however, there exists little literature on their practical implementation. We defined aerial routing networks as a directed graph of paths and intersections. Furthermore, paths were categorised based on the number of dimensions they occupied, lanes (1D), strips (2D) and corridors (3D), with a novel type called a tube (2.5D). Finally, we defined two aerial routing network categories: pre-defined and emergent. A rule-based algorithm was developed to implement aerial routing networks. The proposed method for pre-defined aerial routing networks uses two rules, the flow rule and the geometry rule, to control agents. In the Unity test environment, agents were shown to behave as expected. The only exception is that the effective radius of the helical tubes was larger than expected because of a ‘centrifugal-like’ effect causing agents to reach an equilibrium between the flow rule acting away from the surface of the tube and the geometry rule acting towards the surface of the tube. It was found that for all experiments performed using helical tubes, the effective radius was within 10% of the target radius when using helix angles of less than 45° and a tube radius greater than the turning radius of the agent.

One of the main contributions of this work is characterising the fundamental junctions, converging and diverging. The metric used to measure the performance of each junction was the dimensionless conflict rate, as junctions that require less deconfliction are inherently safer, everything being equal. In these studies, when comparing ribbons and tubes, out-of-plane ribbons had the lowest conflict rate in convergence and divergence junctions. When comparing tubes and corridors, helical tubes had a similar conflict rate to corridors in a convergence junction, with low inflows and were almost conflict-free in diverging junctions. The main shortcoming of the helical tube is the reduced capacity compared to a corridor.

The other use of rule-based algorithms was to design and implement an emergent aerial routing network. The procession rule was designed to attract agents travelling in similar directions, and hence form emergent lanes. Two implementations of this were tested; the first, the velocity-based implementation, which resulted in agents clumping, and the second, the destination-based implementation, which formed lanes as required. Experiments showed that the procession rule was able to significantly reduce the collision and conflict rate while maintaining high transit efficiency in contra flow test cases. When considering uniform random test cases, the procession rule increased the performance of the destination opposite test case but had an insignificant effect on the random destination test case. The main conclusion is that the procession rule is effective when there are several predominant bulk flows, with the only exception that two bulk flows are exactly 90° from each other.

There are many methods proposed for collision avoidance in a distributed control system. The classic method in robotic control theory is the use of a form of repulsion. In this work, this method, called simple separation, is compared to a novel orthogonal separation rule. Orthogonal separation, similar to TCAS, causes agents to use the third dimension to avoid a collision. Results showed that the performance of each method depended on DCD and flow type. For low closing velocities, simple separation required smaller values of DCD to avoid collisions. In contrast, for large closing velocities, orthogonal separation required smaller DCD to avoid a collision.

Next, there have been several instances in the literature where a concept analogous to congestion is described and an attempt to quantify it has been made. However, it was found that the proposed concepts were incomplete or over-simplified. In this work, the concept of congestion is synonymous with the expected risk of conflict between two agents. Three methods were identified to find the expected risk of collision, two of which derive from the concept of traffic pressure, found in traffic theory, and one derived from gas kinematics. The three methods were compared in a test case in which agents flow with a velocity direction variance and a given density. For a range of these independent variables, the congestion metric was compared with the actual conflict rate. It was found that the gas kinematics method had the best correlation. This method was then used to perform field examinations of different flow types, including within a procedurally generated city. Heat maps of each of these flow types was presented and effectively identifies regions of high congestion.

Practical experiments were performed to show aerial routing networks applied in a real-world environment. Agents were composed of a small 6-inch racing drone frame with a flight controller, an on-board computer for decentralised computation, and triple redundant communication. Several experiments were performed for converging and diverging counter-rotating helical tube junctions. The experiments showed that the paths of the agents were consistent with those in the simulation. Minor deviations were found in the paths, which are likely due to position errors and wind. The main challenges of the practical experiments were the typical communication latency of approximately 0.2 s and the communication dropouts.

Finally, advanced applications of aerial navigation structures were proposed. All proposed structures take advantage of the low closing velocity and low conflict rate of helical tubes. Firstly, helical tube intersections use several counter-rotating converging and diverging junctions to make a complex intersections. Terminating helical tubes allow agents to navigate between paths in intersections with more inflows and outflows than a standard converging or diverging junction. Tube-ribbon intersections are scalable intersections that allow agents to transfer between helical tubes. The intersection scale can increase or decrease depending on the number of joining or departing paths.

8.2 Future Work

This work touched on various fields, hence has generated more questions than answers, as much research does. The future work generated from these questions have been categorised into major and minor investigations. The major investigations continue on the main topics, including aerial routing networks and multi-cell congestion. Minor investigations are suggested improvements on methods implemented in this work.

8.2.1 Major - Development of Large Test Cases for Development of Traffic Management Design

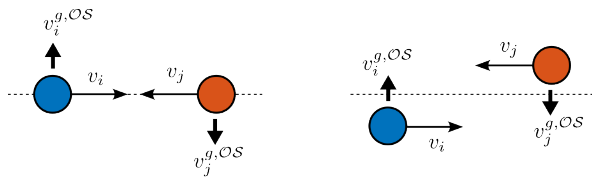
This work has evaluated the characteristics of paths and junctions built with rule-based algorithms in simple small-scale test cases. In future work, these components can be joined to make a full aerial routing network in a large-scale test case. Within the literature, there exist some work in this field. In [7], a aerial routing network with an undirected graph of waypoints is proposed. In [115], an aerial routing network that follows streets, also in an undirected manner, is proposed. The shortcoming of this research is that it does not approach the problem from a design perspective which is iterated. Instead, the research selects an idea and reports on its performance with inconsistent metrics and methods. The aim of the work should be to evaluate traffic management design by testing several implementations of aerial routing networks. Research should be performed in an environments that are an accurate representation of where this technology will be used. For example between distribution centres and residential areas, or from rural outposts to other rural outposts. Results should evaluate the transit efficiency and dimensionless conflict rates of an entire flight for different implementations of aerial routing networks. This research can be further continued by comparing aerial routing networks to other navigation structures like layers and zones, as well as comparisons with emergent navigation structures.

8.2.2 Major - Multi-cell Congestion

In this work, congestion was used to identify regions of higher and lower risk of collision. Arbitrary values, on the order of l_c, were chosen for the cell size when measuring congestion over a larger test case. Furthermore, a fixed grid was chosen. In future work, a method for finding a effective values for the cell size should be developed and experiment with different grid structures. The aim of the work should be to compare regions of high congestion with locations of actual collisions. Using the results of this work, it will be possible to measure congestion over a large airspace, and examine the risk of collision. If a free-flight model is being used, regions of high congestion can have speed limits put in place or navigation structures to increase coordination and thus, reduce congestion.

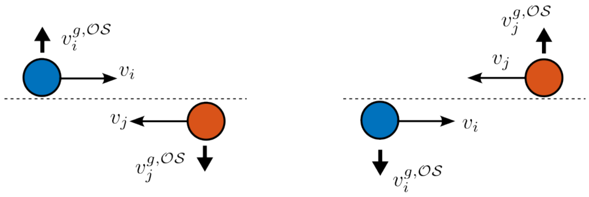
8.2.3 Minor - Improvements to the Orthogonal Separation Rule

The current implementation of the orthogonal separation rule causes agents to separate in the orthogonal direction from the transit plane. In an ideal case, 8.1 (a), agents are co-planar to the



(a) Co-planar velocities (b) Non co-planar velocities

Figure 8.1. Current implementation of orthogonal separation. (a) The ideal case of conflict is where agents are co-planar with the transit plane. Agents then separate orthogonal to this plane. In most cases, however, agents are not co-planar with the transit plane. (b) Shows when agents are on the favourable side of the transit plane, orthogonal separation is required less. (c) Shows when agents are on the unfavourable side of the transit plane, agents will collide.



(a) Standard Cross product (b) Opposite Cross Product

Figure 8.2. Future improvement of orthogonal separation rule to ensure agents always separate away from the transit plane, regardless of orientation.

transit plane. Recall that the transit plane is the plane that is orthogonal to the two destination error vectors of the agents and coincides with the midpoint of the relative position vector of the agents $\mathbf{x}_{ij}$. In a favourable case, the agents in Figure 8.1 (b) are positioned so that the orthogonal separation goal velocity is away from the transit plane. However, in the unfavourable case, the agents in Figure 8.1 (c) will move towards the transit plane and in some cases will cause a collision.

In future work, orthogonal separation could be improved to always apply a goal velocity away from the transit plane, as seen in Figure 8.2.

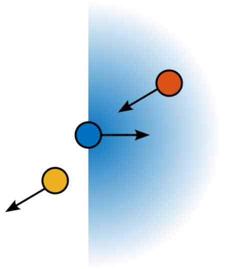


Figure 8.3. Look ahead rule magnitude for orthogonal separation rule. The shaded region represents the positional magnitude weighting of neighbouring agents for the orthogonal separation rule. Agents in the darker region have a greater magnitude for the orthogonal separation rule. In this example, for the classic orthogonal separation implementation, the yellow agent would have a greater effect on the blue agent as it is closer. This is clearly counterproductive as there is no risk of collision. The proposed implementation of the orthogonal separation rule only applies to agents ahead of a given agent. This means that only the orange agent affects the blue agent.

Another improvement to the orthogonal separation rule would only apply a velocity goal for neighbouring agents that are ahead of a given agent. Currently, the maximum effect of the rule is when two agents are at the closest point; however, after the collision has been avoided, the rule still applies a significant goal velocity, even though there is no longer a threat of collision. This is clearly seen in Figure 8.3 as the yellow agent has passed the blue agent, but it will still have a larger affect on the velocity goal than the orange agent. To remove this effect, another magnitude weighting could be implemented to apply the orthogonal separation rule only when the position of the neighbouring agent is ahead of a given agent. This is shown in the figure by showing the rule weighting as a shaded semi-sphere.

Using these two improvements, research should be performed to find the transit efficiency and collision rate for various test cases. The outcomes should provide insight into the performance of these improvements in comparison to the implementation used here.

8.2.4 Minor - Closing Velocity Magnitude Weighting in the Simple Separation Rule

After performing this work, an obvious improvement to the separation rule is a closing velocity rule weighting. The construction of the simple separation rule currently only has a magnitude based on distance and a repulsive output direction. It is hypothesised that a magnitude weighting based on closing velocity, similar to the one used in orthogonal separation, would greatly improve the performance of the separation rule. The main outcomes of this work would compare the performance of all test cases performed in this thesis with the improved separation rule.

8.2.5 Minor - Local Communications

One of the main assumptions in this work is the ability of agents to communicate with each other with low latency, high frequency, and minimal dropouts. The method used in practise was with Wi-Fi modules connected to an access point. This method of communication was successful for the purposes of a small practical demonstration but has many potential flaws in a large-scale application. There are two potential architectures for a large-scale application, (1) a large-scale communication network to which all agents are connected in a centralised manner, and (2) a local mesh-style communication network. There is a large body of work that dictates the theoretical capabilities of communication networks [116]–[118]; however, there exists minimal practical evidence to show the advantages and weaknesses of different communication architectures when used in UTM environments. Research in this field would test the two architectures for various ranges, agent densities, and communication protocols. The results of these experiments should compare the quality of service, latency, dropouts, and effectiveness of each communication architecture.

8.2.6 Minor - Other Emergent Topologies

In this work, emergent aerial routing networks were examined. Within the literature, there are several other predefined topologies that could be explored within an emergent context. The benefits of emergent behaviours are that they are only implemented when they are required. Therefore, for low congestion, airspace can remain free of topologies, allowing agents to travel the shortest route. When congestion increases, emergent behaviours will begin to prevail and generate coordination between agents. It is hypothesised that this coordination will enable the reduction of conflict distances and therefore increase the airspace capacity as required. The outcome of the studies would compare various implementations of different navigation structures. A good performing implementation exhibits low congestion while maintaining high efficiency. The low congestion results in reduction in required conflict distance and therefore higher capacity.

References

[1] Airbus, “Blueprint for the Sky,” Airbus, Tech. Rep., 2018.

[2] T. McCarthy, L. Pforte, and R. Burke, “Fundamental elements of an urban UTM,” *Aerospace*, vol. 7, no. 7, 2020, ISSN: 22264310. DOI: 10.3390/AEROSPACE7070085.

[3] N. S. Labib, G. Danoy, J. Musial, M. R. Brust, and P. Bouvry, “Internet of unmanned aerial vehicles—a multilayer low-altitude airspace model for distributed UAV traffic management,” *Sensors (Switzerland)*, vol. 19, no. 21, 2019, ISSN: 14248220. DOI: 10.3390/s19214779.

[4] Federal Aviation Administration, “Unmanned Aircraft System (UAS) Traffic Management (UTM) Concept of Operations, V2.0,” Tech. Rep., 2020.

[5] V. Lappas, G. Zoumponos, V. Kostopoulos, H. Y. Shin, A. Tsourdos, M. Tantarini, D. Shmoko, J. Munoz, N. Amoratis, A. Maragkakis, T. MacHairas, and A. Trifas, “EuroDRONE, A European UTM Testbed for U-Space,” *2020 International Conference on Unmanned Aircraft Systems, ICUAS 2020*, pp. 1766–1774, Sep. 2020. DOI: 10.1109/ICUAS48674.2020.9214020.

[6] J. Hoekstra, S. Kern, O. Schneider, F. Knabe, and B. Lamiscarre, “Metropolis – Concept design,” vol. 341508, pp. 1–56, 2015.

[7] E. Sunil, J. Hoekstra, J. Ellerbroek, F. Bussink, D. Nieuwenhuisen, A. Vidosavljevic, and S. Kern, “Metropolis: Relating airspace structure and capacity for extreme traffic densities,” *Proceedings of the 11th USA/Europe Air Traffic Management Research and Development Seminar, ATM 2015*, no. June, 2015.

[8] F. Rossi, S. Bandyopadhyay, M. Wolf, and M. Pavone, “Review of Multi-Agent Algorithms for Collective Behavior: a Structural Taxonomy,” *IFAC-PapersOnLine*, vol. 51, no. 12, pp. 112–117, 2018, ISSN: 24058963. DOI: 10.1016/j.ifacol.2018.07.097. [Online]. Available: https://doi.org/10.1016/j.ifacol.2018.07.097.

[9] J. Alonso-Mora, T. Naegeli, R. Siegwart, and P. Beardsley, “Collision avoidance for aerial vehicles in multi-agent scenarios,” *Autonomous Robots*, vol. 39, no. 1, pp. 101–121, 2015, ISSN: 15737527. DOI: 10.1007/s10514-015-9429-0. [Online]. Available: http://dx.doi.org/10.1007/s10514-015-9429-0.

[10] F. Ho, R. Geraldes, A. Goncalves, M. Cavazza, and H. Prendinger, “Improved Conflict Detection and Resolution for Service UAVs in Shared Airspace,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1231–1242, 2019, ISSN: 00189545. DOI: 10.1109/TVT.2018.2889459.

[11] A. Bauranov and J. Rakas, “Designing airspace for urban air mobility: A review of concepts and approaches,” *Progress in Aerospace Sciences*, vol. 125, p. 100 726, 2021, ISSN: 03760421. DOI: 10.1016/j.paerosci.2021.100726. [Online]. Available: https://doi.org/10.1016/j.paerosci.2021.100726.

[12] H. Grotius, *Mare Liberum*, 1st ed. Dutch Republic: Lodewijk Elzevir, 1609.

[13] J. E. G. De Montmorency, “The Control of Air Spaces,” *Problems of the War*, vol. 3, no. Vol. 3, pp. 61–69, 1917.

[14] International Civil Aviation Organization (ICAO), *Convention on Civil Aviation (”Chicago Convention”)*, 1944.

[15] Department of Transport and Civil Aviation Authority, “Upgrading UK airspace: strategic rationale,” Tech. Rep., 2017. [Online]. Available: https://www.gov.uk/government/publications/upgrading-uk-airspace-strategic-rationale.

[16] P. Kopardekar, J. Rios, J. Robinson, M. Johnson, T. Prevot, J. J. C. Ippolito, C. Belcastro, and L. Glaab, "Safely enabling low-altitude airspace operations: Unmanned Aerial System Traffic Management (UTM)," Tech. Rep., 2015.

[17] CAA, "A Unified Approach to the Introduction of UAS Traffic Management," pp. 1–30, 2019. [Online]. Available: `https://publicapps.caa.co.uk/docs/33/CAP1868UTMInnovationHub.pdf`.

[18] Connected Places Catapult, "Implementing an Open-Access UTM Framework for the UK," Tech. Rep. April, 2021.

[19] Single European Sky ATM Research 3 Joint Undertaking, "U-Space: Blueprint," Publications Office of the European Union, Tech. Rep., 2017. DOI: `10.2829/614891`.

[20] R. Shrestha, I. Oh, and S. Kim, "A Survey on Operation Concept, Advancements, and Challenging Issues of Urban Air Traffic Management," *Frontiers in Future Transportation*, vol. 2, no. April, pp. 1–20, 2021. DOI: `10.3389/ffutr.2021.626935`.

[21] Federal Aviation Administration and NASA, "Urban Air Mobility (UAM) Concept of Operations v1.0," FAA, Tech. Rep., 2020.

[22] B. Greenshields, "The Photographic Method of Studying Traffic Behavior," in *Proceedings of the 13th annual meeting of the highway research board*, 1934, pp. 382–399.

[23] F. Kessels, *Traffic Flow Modelling: Introduction to Traffic Flow Theory Through a Genealogy of Models*. Springer, 2019, p. 130, ISBN: 9783319786957. [Online]. Available: `http://www.springer.com/series/13840`.

[24] G. B. Whitham and M. J. Lighthill, "On kinematic waves II. A theory of traffic flow on long crowded roads," *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 229, no. 1178, pp. 317–345, May 1955, ISSN: 0080-4630.

DOI: 10.1098/rspa.1955.0089. [Online]. Available: https://royalsocietypublishing.org/doi/10.1098/rspa.1955.0089.

[25] L. C. Edie, “Discussion of traffic stream measurements and definitions,” Port of New York Authority New York, 1965.

[26] Wilhelm Leutzbach, *Introduction to the theory of traffic flow*, 2-3. 1991, vol. 25, pp. 143–144, ISBN: 9783642648052. DOI: 10.1016/0191-2607(91)90134-c.

[27] D. Helbing, “Gas-kinetic derivation of Navier-Stokes-like traffic equations,” *Physical Review E Volume*, vol. 53, no. 3, pp. 2366–2381, Mar. 1996, ISSN: 1063651X. DOI: 10.1103/physreve.53.2366. [Online]. Available: https://journals.aps.org/pre/pdf/10.1103/PhysRevE.53.2366%20https://journals.aps.org/pre/abstract/10.1103/PhysRevE.53.2366.

[28] ——, “Agent-Based Modeling,” in *Social Self-Organization: Agent-Based Simulations and Experiments to Study Emergent Social Behavior*, D. Helbing, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 25–70, ISBN: 978-3-642-24004-1. DOI: 10.1007/978-3-642-24004-1{_}2. [Online]. Available: https://doi.org/10.1007/978-3-642-24004-1_2.

[29] J. Lee, T. Kim, J. H. Chung, and J. Kim, “Modeling lane formation in pedestrian counter flow and its effect on capacity,” *KSCE Journal of Civil Engineering*, vol. 20, no. 3, pp. 1099–1108, 2016. DOI: 10.1007/s12205-016-0741-9.

[30] ICAO, *Annex 2 - Rules of the Air - Tenth Edition*, July. 2005, pp. 1–74, ISBN: 5149548022. [Online]. Available: http://www.icao.int.

[31] Federal Aviation Administration, *Code of Federal Regulations. Title 14—Aeronautics and Space*, 1999.

[32] CAA, "CAP 1036: Global Fatal Accident Review 2002 to 2011.," Tech. Rep., 2013, pp. 1–134.

[33] C. L. Tallec, P. L. Blaye, C. L. Tallec, P. Le, B. Low, L. Rpas, T. Identification, M. Eucass, C. L. Tallec, and P. L. Blaye, "Low Level RPAS Traffic Identification and Management," in *7TH EUROPEAN CONFERENCE FOR AERONAUTICS AND SPACE SCIENCES (EUCASS)*, 2019.

[34] M. Doole, J. Ellerbroek, V. L. Knoop, and J. M. Hoekstra, "Constrained urban airspace design for large-scale drone-based delivery traffic," *Aerospace*, vol. 8, no. 2, pp. 1–22, 2021, ISSN: 22264310. DOI: 10.3390/aerospace8020038.

[35] M. Grote, A. Pilko, J. Scanlan, T. Cherrett, J. Dickinson, A. Smith, A. Oakey, and G. Marsden, "Sharing airspace with Uncrewed Aerial Vehicles (UAVs): Views of the General Aviation (GA) community," *Journal of Air Transport Management*, vol. 102, Jul. 2022, ISSN: 09696997. DOI: 10.1016/J.JAIRTRAMAN.2022.102218.

[36] Amazon, "Revising the Airspace Model for the safe Integration of sUAS," no. 1, pp. 2–5, 2015.

[37] EmbraerX, "Flight plan 2030," Tech. Rep., 2019.

[38] V. Duchamp, L. Sedov, and V. Polishchuk, "Density-adapting layers towards PBN for UTM," *13th USA/Europe Air Traffic Management Research and Development Seminar 2019*, 2019.

[39] CAA, *CAP 1535S: The Skyway Code (Version 3)*, 2021.

[40] DLR, "DLR Blueprint - Concept for Urban Airspace Integration," no. December, pp. 1–26, 2017.

[41] *North Atlantic Tracks - Wikipedia*. [Online]. Available: https://en.wikipedia.org/wiki/North_Atlantic_Tracks.

[42] *Fastest jet stream on Earth | Guinness World Records*. [Online]. Available: https://www.guinnessworldrecords.com/world-records/613902-fastest-jet-stream-on-earth.

[43] *Federal Airways*. [Online]. Available: https://www.faa.gov/air_traffic/publications/atpubs/pham_html/chap20_section_3.html.

[44] *Next Generation Air Transportation System (NextGen) | Federal Aviation Administration*. [Online]. Available: https://www.faa.gov/nextgen.

[45] *The future of airspace - NATS*. [Online]. Available: https://www.nats.aero/airspace/future/.

[46] K. H. Low, L. Gan, and S. Mao, "A preliminary study in managing safe and efficient low-altitude unmanned aircraft system operations in a densely built-up urban environment," *Air Traffic Management Research Institute, School of Mechanical and Aerospace Engineering Nanyang Technological University*, 2014.

[47] SESAR, "European ATM Master Plan : Roadmap for the safe integration of drones into all classes of airspace," pp. 1–33, 2018. [Online]. Available: https://www.sesarju.eu/sites/default/files/documents/reports/European%20ATM%20Master%20Plan%20Drone%20roadmap.pdf.

[48] D. Sacharny and T. C. Henderson, "Optimal policies in complex large-scale UAS traffic management," *Proceedings - 2019 IEEE International Conference on Industrial Cyber Physical Systems, ICPS 2019*, pp. 352–357, 2019. DOI: 10.1109/ICPHYS.2019.8780127.

[49] ——, "A lane-based approach for large-scale strategic conflict management for UAS service suppliers," *2019 International Conference on Unmanned Aircraft Systems, ICUAS 2019*, pp. 937–945, 2019. DOI: 10.1109/ICUAS.2019.8798157.

[50] D. Sacharny, T. C. Henderson, M. Cline, B. Russon, and E. Guo, "FAA-NASA vs. Lane-Based Strategic Deconfliction," *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, vol. 2020-Septe, pp. 13–18, 2020. DOI: 10.1109/MFI49285.2020.9235239.

[51] D. Sacharny, T. C. Henderson, and M. Cline, "Large-Scale UAS Traffic Management (UTM) Structure," *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, vol. 2020-Septe, pp. 7–12, 2020. DOI: 10.1109/MFI49285.2020.9235237.

[52] L. A. Tony, A. Ratnoo, and D. Ghose, "CORRIDRONE: Corridors for Drones An Adaptive On-Demand Multi-Lane Design and Testbed,"

[53] ——, "Lane Geometry, Compliance Levels, and Adaptive Geo-fencing in CORRIDRONE Architecture for Urban Mobility," *2021 International Conference on Unmanned Aircraft Systems, ICUAS 2021*, pp. 1611–1617, 2021. DOI: 10.1109/ICUAS51884.2021.9476745.

[54] V. R. Challa, M. Gupta, A. Ratnoo, and D. Ghose, "Multiple Lane UAV Corridor Planning for Urban Mobility System Applications," *2021 International Conference on Unmanned Aircraft Systems, ICUAS 2021*, pp. 1003–1009, 2021. DOI: 10.1109/ICUAS51884.2021.9476854.

[55] A. A. Bhise, S. Garg, A. Ratnoo, and D. Ghose, "Signed Distance Function based Geofencing for UAV Corridors," *AIAA Science and Technology Forum and Exposition, AIAA SciTech Forum 2022*, 2022. DOI: 10.2514/6.2022-1507.

[56] S. R. Nagrare, L. A. Tony, A. Ratnoo, and D. Ghose, "Multi-Lane UAV Traffic Management with Path and Intersection Planning," *AIAA Science and Technology Forum and Exposition, AIAA SciTech Forum 2022*, pp. 1–13, 2022. DOI: 10.2514/6.2022-1505.

[57] D. D. Nguyen, J. Rohacs, and D. Rohacs, “Autonomous Flight Trajectory Control System for Drones in Smart City Traffic Management,” *ISPRS International Journal of Geo-Information*, vol. 10, no. 5, p. 338, 2021, ISSN: 22209964. DOI: 10.3390/ijgi10050338.

[58] M. Gharibi, Z. Gharibi, R. Boutaba, and S. L. Waslander, “A density-based and lane-free microscopic traffic flow model applied to unmanned aerial vehicles,” *Drones*, vol. 5, no. 4, pp. 1–25, 2021, ISSN: 2504446X. DOI: 10.3390/drones5040116.

[59] D. S. Jang, C. Ippolito, S. Sankararaman, and V. Stepanyan, “Concepts of airspace structures and system analysis for UAS traffic flows for urban areas,” *AIAA Information Systems-AIAA Infotech at Aerospace, 2017*, no. January, pp. 1–15, 2017. DOI: 10.2514/6.2017-0449.

[60] N. S. Labib, G. Danoy, J. Musial, M. R. Brust, and P. Bouvry, “A multilayer low-altitude airspace model for UAV traffic management,” *DIVANet 2019 - Proceedings of the 9th ACM Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications*, pp. 57–63, 2019. DOI: 10.1145/3345838.3355998.

[61] Q. Quan and M. Li, “Sky Highway Design for Dense Traffic,” Oct. 2020. [Online]. Available: http://arxiv.org/abs/2010.09159.

[62] T. W. M. Randal W. Beard, “Small Unmanned Aircraft Theory and Practice,” Tech. Rep., 2012.

[63] F. Kendoul, “Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems,” *Journal of Field Robotics*, vol. 29, no. 2, pp. 315–378, Mar. 2012, ISSN: 1556-4967. DOI: 10.1002/ROB.20414. [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1002/rob.20414%20https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.20414%20https://onlinelibrary.wiley.com/doi/10.1002/rob.20414.

[64] M. Johnson and J. Larrow, “UAS Traffic Management Conflict Management Model,” *FAA-NASA UTM Research Transition Team: Sense and Avoid Working Group*, 2020.

[65] F. Ho, R. Geraldes, A. Goncalves, B. Rigault, A. Oosedo, M. Cavazza, and H. Prendinger, “Pre-Flight Conflict Detection and Resolution for UAV Integration in Shared Airspace: Sendai 2030 Model Case,” *IEEE Access*, vol. 7, pp. 170 226–170 237, 2019, ISSN: 21693536. DOI: 10.1109/ACCESS.2019.2954987.

[66] K. Burwell, “Multi-Agent Pathfinding for Unmanned Aerial Vehicles,” Ph.D. dissertation, BARCELONA SCHOOL OF INFORMATICS, 2019.

[67] Q. Sajid, R. Luna, and K. E. Bekris, “Multi-agent pathfinding with simultaneous execution of single-agent primitives,” *Proceedings of the 5th Annual Symposium on Combinatorial Search, SoCS 2012*, pp. 88–96, 2012.

[68] A. Felner, R. Stern, S. E. Shimony, E. Boyarski, M. Goldenberg, G. Sharon, N. Sturtevant, G. Wagner, and P. Surynek, “Search-based optimal solvers for the multi-agent pathfinding problem: Summary and challenges,” *Proceedings of the 10th Annual Symposium on Combinatorial Search, SoCS 2017*, vol. 2017-Janua, no. SoCS, pp. 29–37, 2017.

[69] P. Surynek, A. Felner, R. Stern, and E. Boyarski, “Efficient SAT approach to multi-agent path finding under the sum of costs objective,” *Frontiers in Artificial Intelligence and Applications*, vol. 285, pp. 810–818, 2016, ISSN: 09226389. DOI: 10.3233/978-1-61499-672-9-810.

[70] F. Ho, A. Goncalves, A. Salta, M. Cavazza, R. Geraldes, and H. Prendinger, “Multi-Agent path Finding for UAV traffic management,” *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, vol. 1, pp. 131–139, 2019, ISSN: 15582914.

[71] F. Ho, A. Goncalves, B. Rigault, R. Geraldes, A. Chicharo, M. Cavazza, and H. Prendinger, "Multi-Agent Path Finding in Unmanned Aircraft System Traffic Management With Scheduling and Speed Variation," *IEEE Intelligent Transportation Systems Magazine*, no. August 2021, pp. 2–15, 2021, ISSN: 19411197. DOI: 10.1109/MITS.2021.3100062.

[72] B. Nebel, "On the computational complexity of multi-agent pathfinding on directed graphs," *Proceedings International Conference on Automated Planning and Scheduling, ICAPS*, vol. 30, no. Icaps, pp. 212–216, 2020, ISSN: 23340843. DOI: 10.1609/icaps.v30i1.6663.

[73] Y. I. Jenie, E. J. Van Kampen, J. Ellerbroek, and J. M. Hoekstra, "Taxonomy of Conflict Detection and Resolution Approaches for Unmanned Aerial Vehicle in an Integrated Airspace," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 3, pp. 558–567, 2017, ISSN: 15249050. DOI: 10.1109/TITS.2016.2580219.

[74] P. Brooker, "Introducing unmanned aircraft systems into a high reliability ATC system," *Journal of Navigation*, vol. 66, no. 5, pp. 719–735, 2013, ISSN: 03734633. DOI: 10.1017/S0373463313000337.

[75] A. D. Zeitlin and M. P. McLaughlin, "Safety of cooperative collision avoidance for unmanned aircraft," *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*, pp. 1–7, 2006. DOI: 10.1109/DASC.2006.313737.

[76] G. Fasano, D. Accardo, A. Moccia, G. Carbone, U. Ciniglio, F. Corraro, and S. Luongo, "Multi-sensor-based fully autonomous non-cooperative collision avoidance system for unmanned air vehicles," *Journal of Aerospace Computing, Information and Communication*, vol. 5, no. 10, pp. 338–360, 2008, ISSN: 15429423. DOI: 10.2514/1.35145.

[77] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*

- *SIGGRAPH '87*, vol. 21, New York, New York, USA: ACM Press, 1987, pp. 25–34, ISBN: 0897912276. DOI: `10.1145/37401.37406`. [Online]. Available: `http://portal.acm.org/citation.cfm?doid=37401.37406`.

[78] S. A. Quintero, G. E. Collins, and J. P. Hespanha, "Flocking with fixed-wing UAVs for distributed sensing: A stochastic optimal control approach," in *Proceedings of the American Control Conference*, 2013, pp. 2025–2031, ISBN: 9781479901777. DOI: `10.1109/acc.2013.6580133`.

[79] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006, ISSN: 00189286. DOI: `10.1109/TAC.2005.864190`.

[80] B. Crowther, "Flocking of autonomous unmanned air vehicles," *The Aeronautical Journal*, vol. 107, no. 1069, pp. 99–109, 2003, ISSN: 0001-9240. DOI: `10.1017/s0001924000013774`. [Online]. Available: `https://www.cambridge.org/core/journals/aeronautical-journal/article/flocking-of-autonomous-unmanned-air-vehicles/B8B3FE99C99582AE088`

[81] W. J. Crowther, "Rule-based guidance for flight vehicle flocking," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 218, no. 2, pp. 111–124, 2004, ISSN: 09544100. DOI: `10.1243/0954410041322005`.

[82] M. Chen, H. Chu, and X. Wei, "Flocking control algorithms based on the diffusion model for unmanned aerial vehicle systems," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 3, pp. 1271–1282, 2021, ISSN: 24732400. DOI: `10.1109/TGCN.2021.3074073`.

[83] C. Virágh, G. Vásárhelyi, N. Tarcai, T. Szörényi, G. Somorjai, T. Nepusz, and T. Vicsek, "Flocking algorithm for autonomous flying robots," *Bioinspiration and Biomimetics*, vol. 9, no. 2, 2014, ISSN: 17483190. DOI: `10.1088/1748-3182/9/2/025012`.

[84] C. Virágh, M. Nagy, C. Gershenson, and G. Vásárhelyi, “Self-organized UAV traffic in realistic environments,” in *IEEE International Conference on Intelligent Robots and Systems*, vol. 2016-Novem, Institute of Electrical and Electronics Engineers Inc., Nov. 2016, pp. 1645–1652, ISBN: 9781509037629. DOI: 10.1109/IROS.2016.7759265.

[85] G. Vásárhelyi, C. Virágh, G. Somorjai, N. Tarcai, T. Szörényi, T. Nepusz, and T. Vicsek, “Outdoor flocking and formation flight with autonomous aerial robots,” in *IEEE International Conference on Intelligent Robots and Systems*, Institute of Electrical and Electronics Engineers Inc., Oct. 2014, pp. 3866–3873, ISBN: 9781479969340. DOI: 10.1109/IROS.2014.6943105.

[86] G. Vásárhelyi, C. Virágh, G. Somorjai, T. Nepusz, A. E. Eiben, and T. Vicsek, “Optimized flocking of autonomous drones in confined environments,” *Science Robotics*, vol. 3, no. 20, pp. 1–14, 2018, ISSN: 24709476. DOI: 10.1126/scirobotics.aat3536.

[87] S. Hauert, S. Leven, M. Varga, F. Ruini, A. Cangelosi, J. C. Zufferey, and D. Floreano, “Reynolds flocking in reality with fixed-wing robots: Communication range vs. maximum turning rate,” *IEEE International Conference on Intelligent Robots and Systems*, pp. 5015–5020, 2011. DOI: 10.1109/IROS.2011.6048729.

[88] Y. I. Jenie, E. J. Van Kampen, C. C. De Visser, J. Ellerbroek, and J. M. Hoekstra, “Selective velocity obstacle method for deconflicting maneuvers applied to unmanned aerial vehicles,” *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 6, pp. 1140–1145, 2015, ISSN: 15333884. DOI: 10.2514/1.G000737.

[89] J. W. Park, H. D. Oh, and M. J. Tahk, “UAV collision avoidance based on geometric approach,” *Proceedings of the SICE Annual Conference*, pp. 2122–2126, 2008. DOI: 10.1109/SICE.2008.4655013.

[90] J. Montewka, T. Hinz, P. Kujala, and J. Matusiak, “Probability modelling of vessel collisions,” *Reliability Engineering and System Safety*, vol. 95, no. 5, pp. 573–589, 2010,

ISSN: 09518320. DOI: 10.1016/j.ress.2010.01.009. [Online]. Available: http://dx.doi.org/10.1016/j.ress.2010.01.009.

[91] Allianz, "Safety and Shipping Review 2021," no. December 2014, p. 36, 2021. [Online]. Available: http://www.agcs.allianz.com/assets/PDFs/Reports/Shipping-Review-2015.pdf.

[92] C. C. Morris, "Midair collisions: Limitations of the see-and-avoid concept in civil aviation," *Aviation Space and Environmental Medicine*, vol. 76, no. 4, pp. 357–365, 2005, ISSN: 00956562.

[93] World Health Organisation, *Road traffic injuries*, 2021. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries.

[94] *Mid-Air Collision | SKYbrary Aviation Safety*. [Online]. Available: https://www.skybrary.aero/articles/mid-air-collision.

[95] S. Leven, J.-C. Zufferey, and D. Floreano, "A minimalist control strategy for small UAVs," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Oct. 2009, pp. 2873–2878, ISBN: 978-1-4244-3803-7. DOI: 10.1109/IROS.2009.5354465. [Online]. Available: http://ieeexplore.ieee.org/document/5354465/.

[96] S. Kristiansen, "Maritime transportation: Safety management and risk analysis," *Maritime Transportation: Safety Management and Risk Analysis*, vol. 4, no. 1, pp. 1–503, 2005. DOI: 10.4324/978080473369.

[97] S. Leven, J. C. Zufferey, and D. Floreano, "Dealing with midair collisions in dense collective aerial systems," *Journal of Field Robotics*, vol. 28, no. 3, pp. 405–423, May 2011, ISSN: 15564959. DOI: 10.1002/rob.20385.

[98] H. D. Young and R. A. Freedman, "Kinetic molecular model of an ideal gas," *University physics*, pp. 700–701, 2004.

[99] P. T. Pedersen, "Collision and Grounding Mechanics," *Proceedings of WEGEMT'95*, no. March, pp. 125–157, 1995.

[100] R. Golding, "Metrics to characterize dense airspace traffic," no. June, 2018.

[101] K. B. Kramer and G. J. Wang, "Social distancing slows down steady dynamics in pedestrian flows," *Physics of Fluids*, vol. 33, no. 10, p. 103 318, Oct. 2021, ISSN: 1070-6631. DOI: 10.1063/5.0062331. [Online]. Available: https://aip.scitation.org/doi/abs/10.1063/5.0062331.

[102] EUROCONTROL, *Free route airspace (FRA)*. [Online]. Available: https://www.eurocontrol.int/concept/free-route-airspace.

[103] L. Boulton, *Introducing Free Route Airspace into the UK skies - NATS Blog*, Dec. 2021. [Online]. Available: https://nats.aero/blog/2021/12/introducing-free-route-airspace-into-the-uk-skies/.

[104] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 500–505, 1985, ISSN: 10504729. DOI: 10.1109/ROBOT.1985.1087247.

[105] S. Kocherlakota and K. Kocherlakota, *Generalized Variance*, Oct. 2004. DOI: https://doi.org/10.1002/0471667196.ess0869. [Online]. Available: https://doi.org/10.1002/0471667196.ess0869.

[106] W. Feller and V. Feller, *An Introduction to Probability Theory and Its Applications*, ser. An Introduction to Probability Theory and Its Applications v. 1-2. Wiley, 1957, ISBN: 9780471257097. [Online]. Available: https://books.google.co.uk/books?id=K7kdAQAAMAAJ.

[107] *Mean Free Path, Molecular Collisions*. [Online]. Available: http://hyperphysics.phy-astr.gsu.edu/hbase/Kinetic/menfre.html.

[108] *Mean free path & collision frequency - tec-science*. [Online]. Available: https://www.tec-science.com/thermodynamics/kinetic-theory-of-gases/mean-free-path-collision-frequency/.

[109] Unity, *Introduction to collision*, 2022. [Online]. Available: https://docs.unity3d.com/Manual/CollidersOverview.html.

[110] A. S. Aweiss, B. D. Owens, J. L. Rios, J. R. Homola, and C. P. Mohlenbrink, "Unmanned Aircraft Systems (UAS) Traffic Management (UTM) National Campaign II," *AIAA Information Systems-AIAA Infotech at Aerospace, 2018*, no. 209989, pp. 1–16, 2018. DOI: 10.2514/6.2018-1727.

[111] C. Verdonk, H. Jia, F. Javier, and S. Nieto, "H2020-GALILEO-GSA-2017 Innovation Action Galileo-EGNOS as an Asset for UTM Safety and Security Design of UTM System Operational Concept," no. 2019, pp. 1–60, 2020.

[112] J. C. Hill, J. K. Archibald, W. C. Stirling, and R. L. Frost, "A multi-agent system architecture for distributed air traffic control," *Collection of Technical Papers - AIAA Guidance, Navigation, and Control Conference*, vol. 3, no. August, pp. 1936–1946, 2005. DOI: 10.2514/6.2005-6049.

[113] 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," 2009.

[114] C. Lee, *FAR vs BCR*, 2019. [Online]. Available: https://commons.wikimedia.org/wiki/File:FAR_vs_BCR.svg.

[115] V. Polishchuk, E. Sunil, V. d. Vries, K. D. C. Flores, and J. v. Ham, "Concept Design Report Deliverable," Tech. Rep. January 2021, 2021.

[116] A. Chakrabarty, C. Ippolito, J. Baculi, K. Krishnakumar, and S. Hening, “Vehicle to vehicle (V2V) communication for collision avoidance for multi-copters flying in UTM-TCL4,” *AIAA Scitech 2019 Forum*, no. January, 2019. DOI: 10.2514/6.2019-0690.

[117] Y. Zeng, R. Zhang, and T. J. Lim, “Wireless communications with unmanned aerial vehicles: Opportunities and challenges,” *IEEE Communications Magazine*, vol. 54, no. 5, pp. 36–42, May 2016, ISSN: 01636804. DOI: 10.1109/MCOM.2016.7470933.

[118] T. Zeng, M. Mozaffari, O. Semiari, W. Saad, M. Bennis, and M. Debbah, “Wireless Communications and Control for Swarms of Cellular-Connected UAVs,” in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, IEEE, Oct. 2018, pp. 719–723, ISBN: 978-1-5386-9218-9. DOI: 10.1109/ACSSC.2018.8645472. [Online]. Available: https://ieeexplore.ieee.org/document/8645472/.

Appendices

Appendix 1: Publication Plan

Three publications are planned as a result of this work. The following titles and abstracts give an overview of each.

Journal Article: Evaluation of rule-based traffic structures for distributed aerial robotic systems

Using transport structures to organise high-density aerial robotic systems is an area of active research yet little implementation strategy is found in the literature. In this work, rule-based algorithms are proposed to construct distributed transport structures. The proposed methodology is capable of generating simple structures that are comparable to roads and railways, and novel structures such as helical tubes. Of particular interest is how to best create junctions between transport structures to reduce the risk of collision. Experiments were carried out in Unity to compare performance between various structures. The results showed acceptably stable paths of agents while using the structures, and the use of appropriate junction design methodologies are the largest influence on the risk of collision between agents within a junction.

Journal Article: Method for rapid collision risk evaluation in multi agent systems

Within the aerial robotics and multi agent system (MAS) literature, there are several occurrences where risk of collision is alluded to, however not effectively measured. In this study, a particular method for measuring congestion is used to evaluate the risk of collision within a MAS environment. The method can be used on a global scale to measure the congestion globally, or an environment can be divided into smaller cells to measure the congestion locally. This method is evaluated on an entire city model. Results present congestion in a useful heat map format that is able to discern regions of high risk and low risk. This method can be applied to any MAS moving. Future work would develop specific methods of dividing an environment effectively, and applying time and spatial averaging to generate greater resolution risk maps.

Journal Article: Evaluation of emergent rule-based algorithms in multi agent systems

Biological systems evolve and learn behaviours that are advantageous to their survival. In this work, two types of emergent behaviour are analysed and implemented in multi-agent aerial robotic systems using rule-based algorithms. The first, emergent lane formation, has been observed in many biological systems and is implemented to improve flow performance in contra flowing aerial robotic systems. The second, orthogonal separation, is not observed in biological systems, however, causes a helical emergent behaviour, which reduces the risk of collision for incoming agents. The experiment was performed in Unity and compared with a control case in which the emergent rules were not implemented. The results show that the performance in both cases showed a significant improvement in overall system performance.

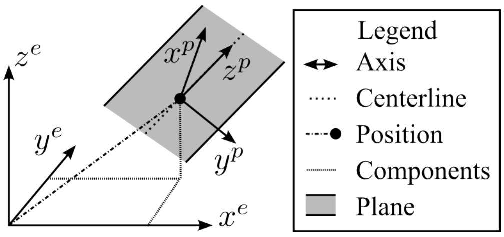


Figure 4. Axes of a plane relative to earth frame of reference.

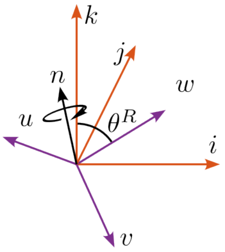


Figure 5. Earth and skyway axes. Step 1, rotate earth axes around n so that k is parallel to w.

Appendix 2: Rotation matrix for skyway to earth conversion

This section outlines the rotation matrix required to convert velocities in the frame of reference of the skyway to earth coordinates.

For simplicity, we will first find the rotation matrix from the earth axes to the skyway axes. From there, the inverse matrix can be used to perform the reverse operation of generating the velocity in earth axes from the skyway axes. This is significantly easier to understand than the reverse.

The rotation matrix has two steps.

- Rotate the z-axis of the earth axes so that it is parallel to $\mathbf{p}_i$, or z^p in Figure 4.
- Rotate the x-axis of the earth axes so that it is parallel to the vertical axis of the skyway, or x^p in Figure 4.

For step 1, let ijk represent the earth reference frame, uvw represent the skyway reference frame $\theta^R = \cos^{-1}(\hat{k} \cdot \hat{w})$, $n = \hat{k} \times \hat{w}$, $\hat{n} = \frac{n}{|n|}$, and $n = (n_x, n_y, n_z)$.

To find the rotation matrix to make k colinear with w, it is as simple as finding n which is orthogonal to the two vectors and rotating the axes about n by angle θ^R. We define the following

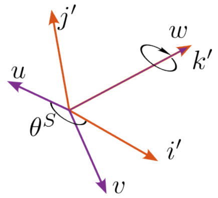


Figure 6. Rotated earth ijk' and skyway axes. Step 2, rotate ijk' around w so that i' and u are parallel.

to do so.

$$a = \cos(\theta^R/2) \tag{1}$$

$$b = \sin(\theta^R/2)n_x \tag{2}$$

$$c = \sin(\theta^R/2)n_y \tag{3}$$

$$d = \sin(\theta^R/2)n_z \tag{4}$$

Hence, the rotation matrix is

$$R_n = \begin{matrix} a^2+b^2+c^2+d^2 & 2(bc-ad) & 2(bd+ac) \\ 2(cb+ad) & a^2-b^2+c^2-d^2 & 2(cd-ab) \\ 2(db-ac) & 2(dc+ab) & a^2-b^2-c^2+d^2 \end{matrix} \tag{5}$$

Now, using this matrix, we transform the coordinates of any point in the earth reference frame into a reference frame where k and w are colinear.

For step 2, we must now rotate i so that it is parallel to u. This rotation is now around w. Let $\theta^S = \cos^{-1}(\hat{i} \cdot \hat{u})$ and $w = (w_x, w_y, w_z)$. We define the following to compute the second rotation matrix.

$$a = \cos(\theta^S/2) \tag{6}$$

$$b = \sin(\theta^S/2)w_x \tag{7}$$

$$c = \sin(\theta^S/2)w_y \tag{8}$$

$$d = \sin(\theta^S/2)w_z \tag{9}$$

$$R_w = \begin{matrix} a^2+b^2+c^2+d^2 & 2(bc-ad) & 2(bd+ac) \\ 2(cb+ad) & a^2-b^2+c^2-d^2 & 2(cd-ab) \\ 2(db-ac) & 2(dc+ab) & a^2-b^2-c^2+d^2 \end{matrix} \tag{10}$$

We can finally compose our rotation matrix by multiplying these matricies together.

$$R = R_n \cdot R_w \tag{11}$$

Hence a position in earth coordinates is represented in skyway coordinates as.

$$x^s = x^e R \tag{12}$$

Therefore, the velocities from skyway coordinates to earth coordinates are computed using the following.

$$v^e = v^s R^{-1} \tag{13}$$

Appendix 3: Research Materials

All data, collection and analysis tools can be found at the following link: Dropbox

There is also a unity package which includes all components. In essence, it is possible to open this package and press play to see experiments in action.