University of Glasgow

Development of a Formation Control Algorithm to Coordinate Multiple Biomimetic AUVs in the presence of Realistic Environmental Constraints

Jonathan McColgan

Submitted in fulfilment of the requirements for the
Degree of Doctor of Philosophy

Aerospace Sciences Research Division
School of Engineering
College of Science and Engineering
University of Glasgow

November 2018

Declaration

I declare that this thesis is my own work and that due acknowledgement has been given by means of complete references to all sources from which material has been used for this thesis. I also declare that this thesis has not been presented elsewhere for a higher degree.

Jonathan McColgan

Glasgow, November 2018

Abstract

Biomimetic Autonomous Underwater Vehicles (BAUVs) are a class of Uncrewed Underwater Vehicle (UUV) that mimic the propulsive and steering mechanisms of real fish. However, as with all UUVs, the range and endurance of these vehicles remains limited by the finite energy source housed on board the vehicle. Unsurprisingly, a consequence of this finite energy source is that BAUVs/UUVs are incapable of completing the large-scale oceanographic sampling missions required to drastically improve our understanding of the Earth's oceans and its processes. To overcome this limitation, this thesis aims to investigate the feasibility of deploying a self-coordinating group of BAUVs capable of completing the aforementioned oceanic surveying missions despite the constraints of the local operating environment.

To achieve this, the work presented in this thesis can be separated into four distinct parts. The first of which is the development of a suitable mathematical model that accurately models the dynamics of the RoboSalmon BAUV designed and built at the University of Glasgow. As well as ensuring the models validity, its ability to efficiently simulate multiple vehicles simultaneously is also demonstrated.

The design and implementation of the formation control algorithm used to coordinate the vehicles is then presented. This process describes the alterations made to a biologically-inspired algorithm to ensure the required parallel line formation required for efficient oceanic sampling can be generated. Thereafter, the implementation of a realistic representation of the underwater communication channel and its debilitating effect on the algorithms ability to coordinate the vehicles as required is presented.

The thesis then describes the incorporation of two methodologies designed specifically to overcome the limitations associated with the underwater communication channel. The first of which involves the implementation of tracking/predictive functionality while the second is a consensus based algorithm that aims to reduce the algorithms reliance on the communication channel. The robustness of these two methodologies to overcoming not only the problematic communication channel but also the inclusion of additional external disturbances is then presented.

The results demonstrate that while the tracking/predictive functionality can overcome the problems associated with the communication channel, its efficiency significantly reduces when the external disturbances are taken into consideration. The consensus based methodology meanwhile generates the required formation regardless of the constraints imposed by both the communication channel and the additional external disturbances and therefore provides the more robust solution.

Acknowledgements

I would like to express my sincere thanks to Dr Euan McGookin for his supervision and support throughout not only my postgraduate research but since 2008 when I first became a student at the University.

I am grateful to the Engineering and Physical Sciences Research Council (EPSRC) for providing the funding for my James Watt Scholarship without which, this research would not have happened.

I would like to take this opportunity to thank Dr Murray Ireland, Dr Kevin Ferguson, Dr John Walker and Dr. Aldo Vargas for their words of wisdom and more importantly, friendship throughout the past 4.5 years. And to Dr Shaun Skinner whose support, guidance and friendship has made this whole journey a lot more fun. I genuinely miss the afternoon strolls for a coffee.

Finally, getting to this point would never have been possible if it weren't for the continued support of my friends and family. In particular, to Mum, Dad, Julie-Ann, Jude, Mark, Emma and Claire – I will never be able to thank you enough for all the support, love and patience you have shown to me throughout this time. I love you all!

Dedicated to the memory of my Brother, Mark Cloherty

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

"How inappropriate to call this planet Earth when clearly it is Ocean."

Arthur C. Clarke

Considering that approximately seventy percent of the Earth's surface is comprised of ocean, it is easy to understand the rationale of British author, Arthur C. Clarke (Lovelock 1990). Perhaps an alternative interpretation of the above quote is that without the oceans, life on this planet would cease to exist. Regardless of one's interpretation, the words of Clarke poignantly highlight Earth's reliance on these vast expanses of water. However, as well as creating life, the oceans are also capable of causing immense devastation through natural disasters such as the recent tsunamis in Indonesia (Matsumaru, Nagami and Takeya. 2012) and Japan (Raby et al. 2015).

Yet, despite its irrefutable influence to life on Earth, the current map of the entire ocean floor is only to a resolution of 5km (Copley 2014). To put this figure into context, NASA's Magellan spacecraft orbited and mapped (in the early 1990's) the vast majority of the surface of Venus – a planet millions of miles from Earth - to a resolution of 100m (Copley 2014; NASA 1994). While there are a number of reasons for this disparity in available resolution, the overriding one is the difficulties associated with radio wave propagation in the underwater environment (Chen et al. 2010). As a result, the ability to use satellite technology similar to that used by the Magellan spacecraft to explore subsurface oceanic features is, and will continue to be, extremely limited (Copley 2014).

With the applicability of satellite technology limited, the vast majority of modern day oceanic exploration (oceanography) is undertaken through a combination of (crewed and uncrewed) Surface Vehicles as well as a group of underwater vehicles collectively referred to as Uncrewed Underwater Vehicles (UUVs) (see Figure 1.1).

Figure 1.1 - Combination of Surface Vehicles and Uncrewed Underwater Vehicles used in Oceanography. (a) - Uncrewed Surface Vehicle (USVs), (b) - Remotely Operated Vehicle (ROV), (c) Autonomous Underwater Vehicle (AUV), (d) - Autonomous Underwater Glider (AUG) and (e) Hybrid AUV/AUG (NOC 2017). (Figure has been removed due to copyright restrictions).

While crewed surface vessels are likely to remain an integral part of ocean exploration for the foreseeable future, their daily operational costs ($20k-$50k) (German et al. 2012) do not make them an economically viable method for the large scale data collection required to drastically improve our understanding of the oceans (Petillo et al. 2012) (Leonard et al., 2007).

UUV's, on the other hand, provide a far more cost effective solution that can result in savings of up to 60% (Chance et al. 2000). These savings are due primarily to the ability of these vehicles to operate without requiring continuous human supervision (Griffiths et al. 2011; Tan et al. 2007). Furthermore, these vehicles are also capable of capturing data at far higher resolutions due to their ability to travel below the ocean's surface (Wynn et al. 2014). However, with these vehicles operating untethered, all necessary operational equipment must be carried on board, including the finite power source. While battery technology has improved drastically since the inception of the first AUV (SPURV) in the 1950's, modern, state of the art UUVs and most notably AUVs are still only capable of covering distances measured in the 100's of kilometres over a period of a few days (Maritime 2017b; Maritime 2017a; Maritime 2018). When this limited range and endurance is coupled with the typical modern AUV mission trajectory shown in Figure 1.2, it is apparent that AUV deployments, in their current guise, do not present an economically viable solution for the large-scale

oceanic mapping discussed above (Rumson 2018). Furthermore, regardless of the particular type of UUV used, the lawnmower trajectory shown in Figure 1.2 also inherently prevents the accurate monitoring of spatiotemporal phenomena such as oil plume tracking (Petillo et al. 2012).

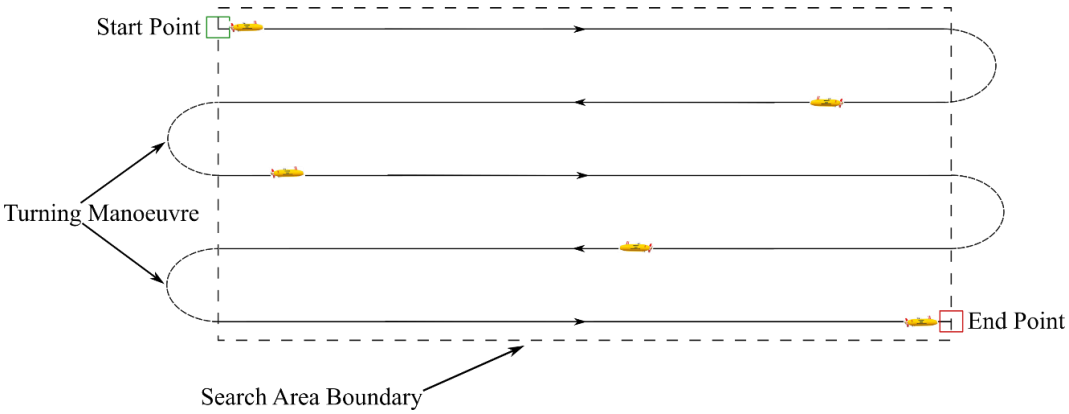


Figure 1.2 - Typical lawnmower trajectory pattern used throughout current AUV missions.

The only option available to overcoming these limitations and drastically improving the efficiency of current UUV missions is to deploy multiple collaborating UUVs (Tena 2018) capable of self-coordinating and generating the parallel line formation shown below in Figure 1.3.

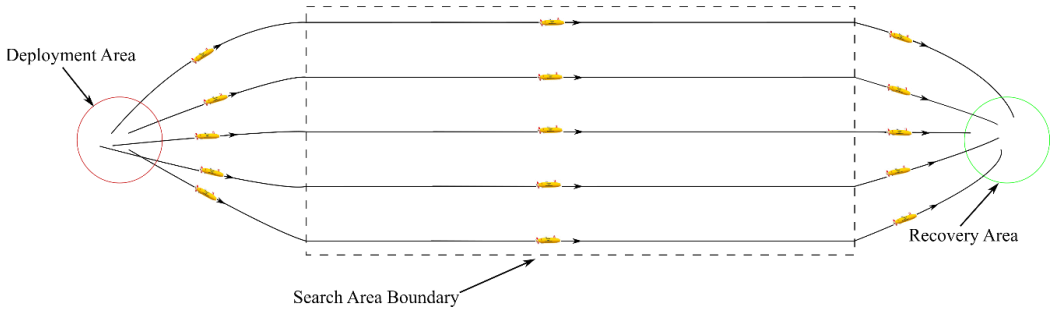


Figure 1.3 - Envisioned mission overview with multiple vehicles generating parallel line formation to complete efficient oceanic mapping.

However, before a deployment such as the one shown above can become a reality, it is necessary to design, test and optimise the necessary algorithms that will allow the various vehicles to self-coordinate and organise themselves into the parallel line formation shown within the *Search Area Boundary* of the above figures. In the field of cooperative robotics, this problem is known as the *formation control problem* and has been studied extensively over the past two decades for air, land, sea and space applications as reviewed in (Murray 2007; Oh et al. 2015) and the references therein.

However, as a result of the Earth's watery veil preventing the successful propagation of radio waves, vehicles operating underwater do so in an environment with a communication channel characterised by low bandwidth, large delays and unreliability (Akyildiz et al. 2004). These problems are further complicated by the presence of unknown and unpredictable external disturbances such as obstacles and oceanic currents.

Nevertheless, since the start of the 21st century, not only has the feasibility of deploying multiple UUVs simultaneously being demonstrated but also the advantages of doing so (Ramp et al. 2009). Included in these projects is the seminal work completed throughout the first Autonomous Ocean Sampling Network (AOSN) collaboration (Fiorelli et al. 2006). In 2003 this work culminated with the successful deployment of three AUVs moving in formation to monitor upwelling and relaxation in the Monterey Bay area (Fiorelli et al. 2006). While multiple projects have since demonstrated the ability to successfully deploy multiple UUVs (Sotzing, C.C, Lane 2010)(Brignone et al. 2009), the majority of these projects have done so by the vehicles periodically surfacing. These surfacing periods permitted the individual vehicles to receive a GPS positional fix as well as new heading commands from onshore base stations. This requirement to periodically surface is particularly inefficient as it means the vehicles are expending a significant amount of energy and time manoeuvring to the surface when they could be continuing mapping operations. Furthermore, the maximum number of vehicles used during these projects has been limited to six vehicles (Leonard et al. 2010). The likely reason for such few vehicle being used in these studies is the costs associated with purchasing the individual vehicles costing approximately $135,000 each (Herkewitz 2013).

Recently, to combat this problem, smaller and far more affordable UUVs have been produced. Included in these, is the Biomimetic AUV, Jeff (Mintchev et al. 2014) which, with a unit cost of only €1200 is significantly cheaper than the vehicles used in the projects discussed above. As discussed in (Tena 2016), this philosophy of designing and developing smaller, more affordable vehicles and deploying them as part of a large collaborative group is a current research and development trend within the UUV community both in industry and academia.

Consequently, based on the above discussion, it is apparent that there is presently scope to investigate the feasibility of designing a formation control algorithm capable of coordinating groups of these smaller, more affordable UUVs into a formation similar to that shown in Figure 1.3. Furthermore, in order to maximise the group's efficiency, the algorithm designed should create the desired formation without requiring the vehicles to

periodically resurface. This requirement will entail the algorithm having to be tolerant to the limited and problematic underwater communication channel (Akyildiz et al. 2004).

To investigate the feasibility of designing such an algorithm, the work presented in this thesis uses a validated mathematical model of the RoboSalmon Biomimetic AUV shown below in Figure 1.4 which has previously been designed, manufactured and tested at the University of Glasgow (Mazlan 2015).

Figure 1.4 - RoboSalmon BAUV designed at the University of Glasgow. (Figure has been removed due to copyright restrictions).

Like the RoboSalmon vehicle, the formation control algorithm designed and presented in this thesis takes inspiration from nature and is based on the behavioural mechanisms of fish partaking in schooling behaviour. As discussed in (Aoki, 1981) and shown below in Figure 1.5, these mechanisms result in each fish manoeuvring in either a *repulsive, orientating* or *attractive* manner depending on the distance between themselves and their nearest neighbour.

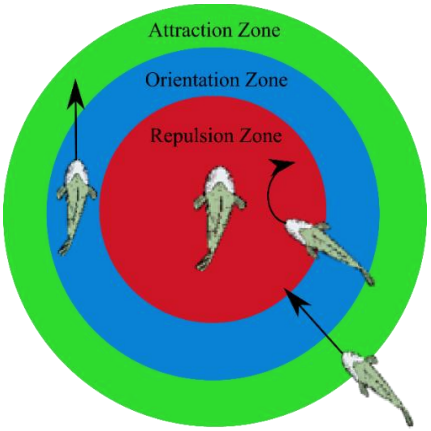


Figure 1.5 - Diagrammatic representation of the behavioural mechanisms used by fish while schooling.

Analysing these behaviours and comparing them with the mission profile outlined in Figure 1.3, it is apparent that they are analogous to the *Deployment*, *Mapping* and *Recovery* phases of the proposed mission profile for multi-vehicle deployments. On initial inspection therefore, these behaviours appear to be a particularly suitable formation control method.

Therefore, the work presented in this thesis details the design process of a formation control algorithm based on the behavioural mechanisms of fish in schooling structures. This study is achieved through simulation studies of the school while completing mission profiles similar to the scenario presented in Figure 1.3. The development of the algorithm has been carried out under realistic environmental constraints by including a credible representation of the underwater environment. This representation includes the incorporation of external disturbances such as ocean currents, subsea obstacles and most importantly, the aforementioned restrictions associated with underwater communications.

1.2 Aims & Objectives

As discussed above, although a number of projects have demonstrated the ability to coordinate a group of UUVs, they have done so by compromising mission efficiency and autonomy by requiring the vehicles to periodically resurface to receive new heading commands and positional fixes. Therefore, it is the aim of this research to design a formation control algorithm capable of coordinating a group of UUVs to complete efficient oceanic mapping missions despite the nuances associated with the underwater environment. However, in order to achieve this, a number of objectives must be set. These objectives are described below:

- Implement an appropriate mathematical model that accurately represents the dynamics of the Biomimetic AUV (BAUV) – RoboSalmon - while maintaining a practical execution time when multiple vehicles are simulated simultaneously;
- Design a suitable formation control algorithm based on the behavioural mechanisms of fish in schooling structures that satisfies the requirements of the mission profile shown in Figure 1.3;
- Implement a realistic representation of the underwater communication channel as well as external disturbances such as oceanic currents and obstacles;
- Design and incorporate suitable functionality within the formation control algorithms that mitigates the nuances associated with the underwater communication channel;

- Complete simulations to illustrate both the effect of implementing the nuances associated with the underwater environment as well as the ability of the functionality implemented to overcome these nuances.

These are the aims and objectives of the work presented in this thesis.

1.3 Contribution of Research

The research presented in this thesis contributes mainly to the fields of cooperative robotics and autonomous systems as applied to the maritime environment. The specific contributions of the work are listed below:

- Implementation, comparison and testing of two system identification techniques to reduce the fidelity of a validated, high-fidelity mathematical model of the BAUV, RoboSalmon;
- Development and implementation of a formation control algorithm based on the behavioural mechanisms of fish within school structures;
- Improvement of the above formation control algorithm's design to ensure the required parallel line formation shown in Figure 1.3 is generated;
- Application and comparison of predictive and consensus based control methodologies to successfully mitigate the effect of the underwater communication channel; and
- Creation of a formation control algorithm that operates as required despite the nuances of the underwater communication channel as well as the presence of external disturbances such as obstacles and oceanic currents.

This work contributes to autonomous systems and cooperative robotics within the maritime environment as it demonstrates the validity of using predictive and consensus based control methodologies to overcome the constraints of the underwater communication channel. Furthermore, the work also demonstrates the creation of a formation control algorithm that is capable of operating with minimal human supervision and without the need to periodically resurface.

At present, the following publications have resulted from the work completed within this thesis:

- McColgan, J. and McGookin, E.W., (2015), " Analysis of the group structure of a school of biomimetic AUVs coordinated using nearest neighbour principles", *6th*

International Conference on Automation, Robotics and Applications (ICARA), Queenstown, New Zealand, Feb 17-19, pp. 312-317.

- McColgan, J. and McGookin, E.W., (2015), "A low fidelity mathematical model of a biomimetic AUV for multi-vehicle cooperation", *OCEANS 15,* Genova, Italy, 18-21 May, pp. 1-10.
- McColgan, J. and McGookin, Euan W. (2016). "Coordination of Multiple Biomimetic Autonomous Underwater Vehicles Using Strategies Based on the Schooling Behaviour of Fish." *Robotics* 5, no. 1: 2.
- McColgan, J. and McGookin, E.W. (2017). "Effect of Communication Delays on the Successful Coordination of a group of Biomimetic AUVs", *OCEANS 17*, Aberdeen, Scotland, 19-22 June 2017.

Other publications are in preparation and will be published in due course.

1.4 Thesis Outline

The main focus of this work is to design a formation control algorithm that is capable of coordinating a group of vehicles into the required formation to ensure efficient oceanic mapping missions can be undertaken. The algorithm design is also required to generate this formation regardless of the nuances associated with the underwater environment. The development and improvements incorporated in the algorithm are presented in stages throughout this thesis. With the exception of Chapter 2, each chapter presents the theory and associated functionality implemented within the mathematical model before presenting and analysing the results obtained from the simulations completed at each stage of the algorithm's development.

Chapter 2 provides an overview of the relevant literature associated with the work completed in this thesis. This includes a review of the current state of the art in both UUV design and autonomy and why it is important to the work completed in this thesis. Thereafter, this chapter presents a brief overview of the different projects associated with the deployment of multiple collaborating UUVs and how they relate to the work completed in this thesis. Lastly, this chapter presents the different types of formation control methodologies available and which ones have been incorporated recently within studies associated with AUVs.

Chapter 3 begins by providing a brief overview of the RoboSalmon vehicle and the mechanisms used to allow it to imitate the swimming properties of the North Atlantic salmon. Thereafter, this chapter describes in detail the original, high-fidelity mathematical model of the BAUV, RoboSalmon. This chapter then analyses the suitability of this model

to be used to model multiple vehicles simultaneously. The subsequent changes made to the model are described in detail and suitable validation tests are completed to prove the performance of these reduced fidelity models.

Chapter 4 begins by briefly describing the behavioural mechanisms associated with fish in schooling structures. In terms of coordinating the vehicles, these behaviours are replicated into a number of heading and velocity control laws that when combined, make up the initial formation control algorithm used in this work. Once the formation control algorithm has been presented, its suitability to successfully coordinate the vehicles as required is presented.

Chapter 5 describes the modifications made to the algorithm presented in Chapter 4 in order to ensure that it operates as efficiently as possible and is able to generate the required parallel line formation. The modified formation control algorithm is then tested in a similar manner to the original algorithm of Chapter 4.

Chapter 6 begins by describing the nuances associated with the underwater communication channel and the methods used to include them within the work completed in this thesis. Simulations are then completed to test how varying parameters associated with the underwater communication channel effects the algorithms ability to operate as required.

Chapter 7 presents and describes the predictive and consensus based methodologies incorporated within the formation control algorithm in order to overcome the nuances of the underwater communication channel. The results obtained from the various simulations completed are presented and a determination of whether or not the methodologies implemented operate as expected is made.

Chapter 8 begins by describing the functionality included to ensure a realistic representation of both oceanic currents and external obstacles are included. Thereafter, the exact same simulations completed in Chapter 7 are repeated and based on the results obtained, the most suitable control method (predictive or consensus based) is chosen. This chapter then describes the results obtained from additional simulations completed to demonstrate the robustness of the chosen algorithm.

Chapter 9 presents the results obtained from simulations completed to demonstrate that the formation control algorithm designed and developed from Chapters 4 through to 8 operates as required and regardless of the local operating environment during realistic oceanic sampling missions.

Chapter 10 concludes this thesis by given a summary of each chapter and the conclusions drawn from the results obtained. Areas of possible future work are also given at the end of this chapter.

Chapter 2

Literature Review

2.1 Introduction

Autonomous Underwater Vehicles (AUVs) like all Uncrewed Underwater Vehicles (UUVs) are extremely complicated robotic systems and are made up of many smaller interacting subsystems. These subsystems involve the merging of several areas of technology including (but not limited to) guidance, control and navigation techniques, communication systems and their associated protocols, computer science, sensor technology, data fusion as well as the vehicle design itself (Finn & Scheding 2010). As a result, it is easy to comprehend that once the formation control aspect of multi-AUV collaboration is taken into consideration, the literature available to review is exhaustive. Therefore, only the subsystems pertinent to the work completed in this research will be reviewed in this chapter, i.e. the guidance, control and navigation systems, vehicle design, the communication subsystems as well as the different formation control methodologies available.

Therefore, this chapter provides a review of the relevant literature in the following manner. Section 2.2 contains a brief summary of the evolution of AUV development from their inception to the modern day. Section 2.3 provides a detailed review of the current state of the art in AUV technology. Section 2.4 provides an overview of the various subsystems associated with an AUV that are pertinent to the work completed in this thesis. Finally, Section 2.5 provide a summary of the literature reviewed in this chapter.

2.2 AUV Development

The inception of AUV technology can be traced back to the early 1950's when researchers within the Applied Physics Laboratory at the University of Washington began developing what would become in the 1960's, the first operational AUV (Alt 2003).

The Special Purpose Underwater Research Vehicle (SPURV) as it was named was capable of operating at a maximum depth of 3000m for up to five and half hours (Widditsch 1973) while using the on board temperature and conductivity sensors to record data relating to diffusion studies (Drones 2010).

In the following two decades, AUV development continued, albeit at a gradual pace, due primarily to the required technology being unavailable resulting in a lack of interest from those out with the relevant research communities (Blidberg 2001). However, with the production of higher density batteries and the availability of ever more compact computer processing units, interest in the development of AUV technology was rejuvenated towards the end of the 20th century. This rejuvenation resulted in the development and deployment of a number of new, more sophisticated AUVs by a number of institutions including Massachusetts Institute of Technology (MIT), Woods Hole Oceanographic Institution (WHOI) and Southampton's Oceanography Centre (SOC) as highlighted below in Table 2.1.

Table 2.1 - Comparison of Different AUVs (Alt, 2003).

Name	Institution	Depth Rating (km)	Max Endurance (hrs)	Weight (kg)
Odyssey	MIT	6	6	195
ABE	WHOI	5	34	68
AutoSub	SOC	1	50	1700

The subsequent missions completed by these vehicles resulted in a number of world firsts including the successful laying of 175km of fibre optical cabling under the arctic ice (Ferguson et al. 1999) and surveying subsea lava flows (Yoerger et al. 1998).

The reliability and capability demonstrated by the above vehicles combined with the need for the hydrocarbon industry to start exploring deeper waters resulted in the subsequent commercialisation of AUV technology. Furthermore, with governments across the world also beginning to realise the potential for AUVs to be used as efficient maritime reconnaissance platforms (U.S. Navy 2004), demand and investment in AUV technology grew rapidly in the decade following the turn of the century (Nicholson, J.W. Healey 2008). This new found demand resulted in the number of operational AUVs available increasing dramatically from only a handful in the mid 1990's (Bellingham & Rajan 2007) to the point whereby in 2007, the number of AUVs used exclusively for military purposes was equal to 82 (Moline, Dana L. Woodruff, et al. 2007). Since then, continued investment in the relevant technology has resulted in an increased confidence in the ability of AUVs to reliably and successfully meet the operational requirements demanded from them by the scientific, military and commercial sectors. As a result, it is predicted that by 2020, the global AUV

market will increase by 49% resulting in the demand for over 900 units (Douglas-Westwood 2016).

Although the demand for AUVs has increased significantly since their inception in 1960, one facet of their operation that has remained largely unaltered is their *modus operandi*. AUV missions, since the deployment of the SPURV vehicle have all operated by traversing through a number of pre-programmed waypoints (Tena 2013). Typically, these waypoints will result in the vehicle following a lawnmower pattern trajectory similar to that shown below in Figure 2.1.

Figure 2.1 - Typical lawnmower trajectory used during AUV missions (Moline, Woodruff, & Evans, 2007b). (Figure has been removed due to copyright restrictions).

The picture presented above represents the trajectory of the popular REMUS AUV during a mission to collect bathymetric data in Sequim Bay in 2007 (Moline, D.L. Woodruff, et al. 2007). The trajectory presented demonstrates the typical profile of a modern AUV mission including the vehicle being deployed from a surface vessel (yellow circle), the vehicle manoeuvring between the different pre-programmed waypoints (blue line) and finally, the vehicle manoeuvring to a specified location (white circle) to be recovered.

The mission profile outlined in Figure 2.1 highlights a number of limitations associated with current AUV operational capability. The first and perhaps most obvious is highlighted by the vehicles trajectory (blue line) and the fact that when compared with the surrounding ocean, the area mapped is relatively small. Secondly, the lawnmower trajectory presented in Figure 2.1 also highlights the limited autonomy present in modern AUV missions in that although the vehicle can operate unsupervised and manoeuvre between the different waypoints, it doesn't have the autonomy to make changes to the mission plan. This means that potential areas of interest can only be identified after the mission has ended and the data analysed.

Based on the above discussion and that presented in the literature (Tangorra et al. 2011), it is apparent that there is currently two factors affecting the operational capability of AUVs; firstly, limited range and endurance due to the finite energy source and secondly, limited vehicle autonomy. If, as highlighted within (German et al. 2012), advances can be made in these areas, the operational capability of AUVs will improve significantly in the next decade.

2.3 State of the Art in AUV Technology

As discussed above, increased range and vehicle autonomy have been highlighted as two key areas of AUV development that if improved, could improve the efficiency of modern AUV missions. The following sections will now describe in detail the current state of the art associated with these two areas of AUV research and development.

2.3.1 Vehicle Design

As shown below in Figure 2.2, since their inception in the late 1960's, the aesthetics of AUVs has remained largely unchanged with a cylindrical shaped hull that optimises the trade-off between hydrodynamic efficiency, structural integrity and the need to maximise the interior space available (Gao et al. 2016). The pictures presented also demonstrate that AUVs all use propellers for propulsion and a rudder for *lateral* manoeuvrability. Finally, *vertical* manoeuvrability is achieved by either external control surfaces positioned at the front or rear of the vehicle (Figure 2.2 (b) and (c)) or by altering the vehicles buoyancy using ballast tanks (Figures 2.2 (a) and (d)).

Analysing the mass and range values associated with each AUV in Figure 2.2, it is apparent that in order to increase the range of an AUV, the size (mass) of the vehicle must be increased significantly. While technology has improved over the decades to provide batteries with greater energy densities (Wang et al., 2012), it remains a design restriction of traditional propeller based AUVs that significant improvements in range and endurance can only be achieved by increasing the number of batteries housed within the vehicle.

(a) SPURV AUV. Operational - 1959 – 1979. Mass – 484kg. Range – 29km

(b) Thesus AUV. Operational – 1992-1996. Mass - 8600kg. Range -1360km

(c) REMUS 100 AUV. Operational 2001 – Present. Mass – 82kg. Range – 72km

(d) ecoSUBμ5. Operational 2015 – Present. Mass – 4kg. Range 50km

Figure 2.2 - Evolution of AUV Design (EcoSUBRobotics, 2018; Ferguson et al., 1999; Gafurov & Klochkov, 2015; Kongsberg, 2018). (Figure has been removed due to copyright restrictions).

This is highlighted below in Table 2.2 where the near identical HUGIN 1000 and 3000 AUVs are compared in terms of their mass and range. As the data demonstrates, The HUGIN 1000

weighs 850kg and can cover a distance of approximately 177km whereas the HUGIN 3000, weighing 550kg heavier provides a range of 442km.(Kongsberg Maritime, 2017).

Table 2.2 - Mass and range data associated with the Hugin 1000 and 3000 AUVs.

Vehicle	Mass (kg)	Range (km)
Hugin 1000	850	177km
Hugin 3000	1400	442

As shown below in Figure 2.3, this relationship between an AUV's mass and its associated range is common amongst the various commercially available AUVs (Kongsberg Maritime, 2017)(ECA Group 2018; Generaly Dynamics 2018) presently on the market. Based on this data, it is apparent that the current state of the art in vehicle design and battery technology only allows for a maximum range value of just under 600km. However, as highlighted by (Griffiths et al. 2004) simply increasing the size of AUVs to increase their range isn't feasible or practical as it would require a vehicle weighing approximately 30 tonnes to produce a long endurance AUV based on current battery technology. In order to overcome this limited range problem, a number of organisations have designed vehicles with an entirely different design philosophy. These vehicles will now be discussed in the following sections.

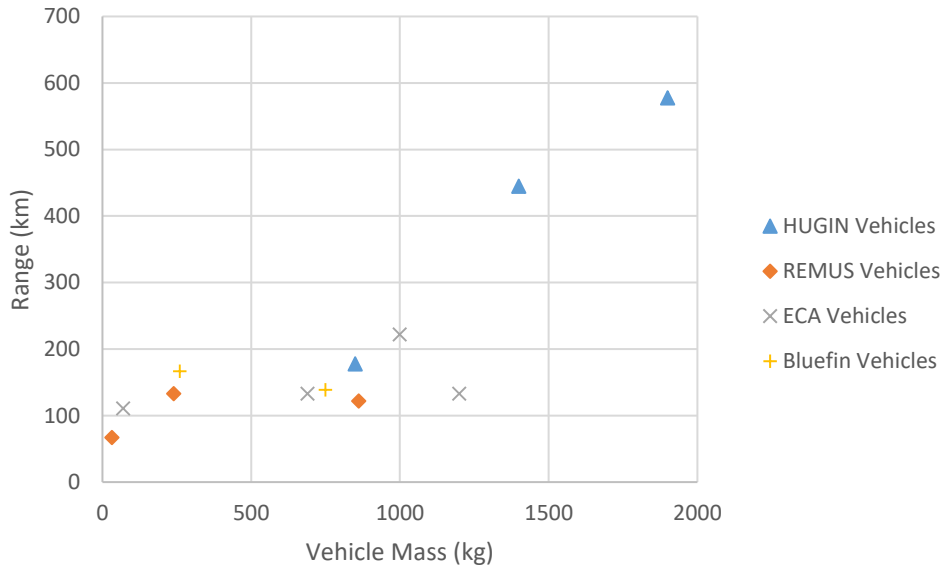


Figure 2.3 - Comparison of range and endurance values for various families of AUVs.

2.3.1.1 Autonomous Underwater Gliders (AUGs)

Borne out of the desire to produce a vehicle capable of completing long endurance missions, a number of projects where initiated whose purpose was to replace or dramatically improve the range and endurance of traditional propeller based AUVs (Claus & Bachmayer 2012).

The result was the creation of an entirely new type of UUV that completely removed the need for a propeller based propulsion system and instead, this new vehicle, used a combination of buoyancy control and wings to allow the vehicle to 'fly' through the water. As well as using a more efficient propulsive system, Autonomous Underwater Gliders (AUGs) as they are commonly referred to as, also adopt far less power intensive sensors and on board control systems (Eriksen et al. 2001). The combination of these factors results in a vehicle that is capable of operating continuously for months at a time while monitoring oceanic characteristics over 1000's of kilometres (Webb et al. 2001). An example of two commercially available AUGs, Seaglider and Slocum are shown below in Figure 2.4.

(a)

(b)

Figure 2.4 - Two examples of commercially available AUGs (a) Seaglider (Kongsberg, 2018) and (b) Slocum. (Figure has been removed due to copyright restrictions).

While the range and endurance figures associated with AUGs satisfy the requirements for long endurance missions, there are also a number of operational limitations associated with these vehicles. The most notable of these is the fact that their maximum forward velocity is limited to between 0.25 m/s and 0.45 m/s (Davis et al. 2002). As a result, one of the disadvantages of operating UGSs is that their trajectories are susceptible to being dictated by the direction of ocean currents (Eriksen et al. 2001).

2.3.1.2 Hybrid Autonomous Underwater Vehicles (HAUVs)

In a bid to combine the best design facets of both traditional propeller based AUVs and AUGs, a team from the National Oceanographic Centre in Southampton have designed the AutoSub Long Range (Furlong et al. 2007) vehicle shown below in Figure 2.5.

The AutoSub LR vehicle can be considered a hybrid AUV as it contains technology normally used exclusively for either an AUV or an AUG. This includes the adoption of a propeller based propulsive system commonly seen on an AUV and a set of wings that as discussed above, are an integral design feature of AUGs.

Figure 2.5 - AutoSub Long Range designed by the Marine Autonomous and Robotics Systems team at the National Oceanographic Centre (NOC 2017). (Figure has been removed due to copyright restrictions).

While normally the inclusion of a propeller based propulsion system would consume significant amounts of energy, the AutoSub LR vehicle has been designed specifically to operate optimally at velocities in the region of 0.5 m/s and 1 m/s and thus, the power consumption of the vehicle has been significantly reduced (Furlong et al. 2012). Furthermore, the vehicle's on board control and sensor systems have been chosen specifically because of their low power consumption characteristics (Wynn et al. 2012). As a result of these two design features, the AutoSub LR is capable of operating continuously for up to six months while covering a distance of approximately 7,000 km (Marshall 2016).

2.3.1.3 Biomimetic Autonomous Underwater Vehicles (BAUVs)

So far, the discussions presented above have demonstrated that in order to improve their range and endurance, the design of underwater vehicles has had to evolve and as result, three categories of underwater vehicle now exist; *AUVs*, *AUGs* and *Hybrid AUVs*. While the design of these three vehicles have focussed primarily on improving their range and endurance, a fourth category now exists that focusses on improving their manoeuvrability characteristics.

In order to achieve this improved manoeuvrability, these vehicles take inspiration from nature and copy the propulsive and steering mechanisms of real fish (Bar-Cohen 2011). Unsurprisingly, this particular type of AUV is commonly referred to as a Biomimetic AUV (BAUV). As it is the type of vehicle used throughout the work presented in this thesis, the following section will describe in detail the current state of the art associated with this particular type of AUV.

Although the concept of imitating nature is not new, its application to underwater vehicles has a relatively short history with the production of *RoboTuna* at MIT in the mid-to-late 1990's (Barrett 1996) signalling the start of biomimetic underwater vehicle research. As with the recent FILOSE project (Salumäe 2013), the original purpose of *RoboTuna* was to facilitate the collection of data associated with the hydrodynamic properties of fish like swimming. Nevertheless, the overriding success of the project resulted in the production of the world's first biologically inspired underwater vehicle, the *vorticity-controlled unmanned underwater vehicle* (VCUUV) (Anderson & Chhabra, 2002) shown below in Figure 2.6.

The VCUUV was the result of a collaboration between MIT and Draper laboratories aimed at producing the first mission ready, biologically inspired underwater vehicle. Aesthetically, the vehicle's design is a scaled up replica of the aforementioned *RoboTuna*. However, to ensure the vehicle could operate untethered, its design was split into two sections (Cho 1998): a forward hull provided the necessary space to house the various systems required for the vehicles operation while a hydraulically actuated multi-segmented tail section allowed the swimming gait of the tuna fish to be replicated. The experimental results (Anderson & Chhabra 2002) for the vehicle demonstrated its exceptional turning performance, with turning rates of up to 75 degrees per second being achieved.

(a) (b)

Figure 2.6 - Vorticity-Controlled Unmanned Underwater Vehicle (VCUUV) when (a) assembled and (b) undertaken a turning manoeuvre with full body deflection. (Cho 1998) (Figure has been removed due to copyright restrictions).

As shown below in Table 2.3, the general trend of BAUVs is to possess extremely good manoeuvrability characteristics but poor straight line speed capability. This lack of forward speed is believed to be a result of the design trade-off that takes place between accurately replicating the swimming gait of the fish while trying to keep the complexity of the vehicle's design to a minimum. As stated by (Valdivia y Alvarado 2007), in order to accurately replicate the swimming gait of fish using the multi-segment approach, a large number of degrees of freedom (segments) are required. However, by increasing the number of degrees of freedom within a system, the sophistication of the necessary control techniques increases (Yu et al. 2004) as does the challenges associated with keeping the electrical equipment waterproof (Kumph 2000). It is for these reasons that the majority of the projects in Table 2.3 decided to use between one and three joints in their design. As a result, the vehicles were incapable of producing an inaccurate replication of the fish's swimming gait (Roper et al. 2011). Furthermore, the focus of these projects on only using the posterior section of vehicle to replicate the swimming gait resulted in an excessive lateral oscillation of thrust at the tail of the vehicle, producing an undesirable and detrimental excessive yawing motion (Clapham 2015; C. M. Watts 2009; Mazlan 2015).

Table 2.3 - Summary of Prototype Biomimetic Underwater Vehicles.

Name	**Straight Line Speed** (m/s)	**Turning Capability**	**Reference**
RoboPike	0.25	17.5 deg/s	(Kumph 2000)
PF-300	0.20	36.0 deg/s	(Hirata et al. 2000)
PF-2001	0.97	-	(Hirata & Kawai 2001)
G9	0.80	130.0 deg/s	(Liu et al. 2005) (Hu 2006)
UWFUV	0.60	-	(Morgansen et al. 2007) (Triplett 2008)
RoboSalmon V2	0.20	50 deg/s	(C. Watts 2009)
SPC-III	1.36	1 Body Length	(Bibuli et al. 2011)
BIOSwimmer	2.50	< 1 Body Length	(Conry et al. 2013)
iSplash-I	0.70	0	(Clapham & Hu 2014a)
iSplash-II	3.70	0	(Clapham & Hu 2014b)
RoboSalmon V3	0.19	0.64 Body Length	(Mazlan 2015)

In an attempt to alleviate this problem, the G9 BAUV shown below in Figure 2.7 (a) altered the approximation of the swimming gait to take into account the motion of the anterior section of the vehicle (Liu & Hu 2010). Although the results produced an improvement in the swimming speed, the vehicle was still unable to achieve a realistic swimming speed. Furthermore, (Watts & McGookin 2013) (Figure 2.7 (b)) and (Mazlan 2015) (Figure 2.7 (c)) both designed vehicles with significantly more joints as well as including an actuated

anterior section in order to investigate whether or not the recoiling motion of the vehicle could be reduced. The results produced where conflicting with Watts suggesting that an improvement in performance could be obtained while Mazlan's results demonstrated a 26% reduction in straight line swimming speed. Regardless, both projects demonstrated that even with increasing the number of segments within the posterior section, the straight line speed obtained was still below that achieved in nature. It is believed that the complexity of the above designs resulted in increased mechanical losses and as a result, unforeseen early saturation of the actuators.

(a) University of Essex – G9 (b) University of Glasgow – RoboSalmon V2

(c) – University of Glasgow – RoboSalmon V3 (d) – University of Essex - iSplash-II

(a) – Boston Engineering - BIOSwimmer

Figure 2.7 - Various BAUV designs.(Liu et al. 2005) (Watts 2009) (Mazlan 2015) (Clapham & Hu 2014a)(Boston Engineering 2014) (Figure has been removed due to copyright restrictions).

However, importantly, the work completed by Watts was the first attempt at comparing the efficiencies of a biomimetic AUV with that of a similarly sized traditional propeller based AUV. Although the Biomimetic AUV was unable to match the traditional AUV in

terms of speed, the experimental results demonstrated that when both vehicles travel at the same speed, the RoboSalmon vehicle is more efficient by a factor of three (McGookin & Watts 2012)(Watts, 2009). Similar results have been presented more recently in (Bibuli et al. 2011). These results also demonstrate that for the same forward speed, the biologically propelled vehicle not only consumed approximately 7% less power but its turning radius was also 40% smaller than that of the propeller based vehicle.

Therefore, while BAUVs provide improvements in both manoeuvrability and overall system efficiency, their inability to achieve forward velocities greater than one metre per second have resulted in their applicability to realistic ocean sampling scenarios being questioned (Clapham & Hu 2014b; Valdivia y Alvarado 2007). This is due to the fact that at these velocities and similarly to AUGs, it is likely that ocean currents will dictate the vehicles trajectory. However, as shown in Table 2.3 there are two vehicles capable of achieving significantly greater forward velocities, the iSplash-II vehicle designed at the University of Essex and the BIOSwimmer produced from Boston Engineering.

The work completed by (Clapham 2015) culminating in the iSplash (Figure 2.7 (d)) series of vehicles created a novel mechanical drive system combining a single electrical motor with a crankshaft to replicate the full-body undulatory motion of the *carangiform* carp fish. This new design methodology resulted not only in a dramatic increase in the vehicles ability to accurately replicate the swimming gait of the carp fish but also the frequencies of the tail's undulatory motion. As a result, for the first time, a BAUV could travel faster than the species of the fish it was based on (Clapham & Hu 2014b). While the design excelled at propelling the vehicle in a straight line, it didn't include a method to allow the vehicle to move in either the horizontal or vertical planes and as Clapham himself concluded, this ability must be incorporated before the real world applicability of the vehicle is realised.

The BioSwimmer design philosophy meanwhile is more pragmatic in dealing with the straight line speed limitations of BAUVs. As shown in Figure 2.7 (e), similarly to the other BAUVs, it still uses a multi-segmented tail section to provide the superior manoeuvrability characteristics (in yaw) while simply attaching a traditional propeller to the end of the tail section to provide the additional thrust required to achieve sufficient straight line speed (Conry et al. 2013). Additionally, motion in the vertical plane is achieved through the actuation of a pair of pectoral fins on either side of the hull (Conry et al. 2013).

The above work has described the current state of the art in AUV vehicle design. The literature reviewed has demonstrated that presently, there are a four different types of underwater vehicle that can be deployed for oceanic mapping missions. While not

explicitly stated, it is apparent that based on the mission requirements, a particular type of AUV will be more suited than the others. For example, traditional propeller based AUVs with their superior velocity are more suited to search and recovery missions where the rapid assessment of a particular area is required. Conversely, AUGs and hybrid AUVs are more likely to be used in scientific mapping missions were ocean processes that evolve over large spatiotemporal domains need to be monitored and therefore, the long endurance capabilities of AUGs and hybrid AUVs are more suited. Finally, BAUVs with their superior manoeuvrability characteristics are more suited to be used for missions completed in confined environments such as harbours, off shore structures or shipwrecks.

However, regardless of the particular type of vehicle used and as discussed previously, underwater vehicles presently have very limited autonomy. Therefore, it is widely believed within the AUV community that improvements in vehicle autonomy will drastically improve the efficiency of oceanic mapping missions more so than any future iterations/improvements in vehicle design. The following section describes in detail the current state of the art in underwater vehicle autonomy.

2.3.2 Vehicle Autonomy

Autonomy within the robotics community is used to describe the level to which an uncrewed vehicle can make mission critical decisions without the permission of a human controller. Presently, the definition of these different levels is somewhat ambiguous with various bodies – mainly governmental - offering alternative definitions. Among these, is the definitions provided by the Office of Naval Research (Button, Kamp, Curtin, & Drydon, 2009), the DoD's Unmanned Aerial Vehicle Roadmap (Clough 2002) and finally those from the Autonomy Levels for Unmanned Systems (ALEFUS) working group (Huang et al. 2005). However, only the DoDs Aerial Vehicle Roadmap provide definitions that encompass both solo and group autonomy and as a result, it will be these definitions that will be used in the following work.

As shown below in Table 2.4, the roadmap defines twelve different autonomous control levels (ACLs) that encompass systems containing zero autonomy (Remotely Operated Vehicles (ROVs)), partial autonomy (AUVs with obstacle avoidance capability) and finally, a fully autonomous system which, if existed would represent an AUV capable of completing any mission objective without requiring any human input (Suresh & Ghose 2010).

Table 2.4 - Description of different Autonomous Control Levels (ACLs).

Autonomous Control Level (ACL)	Name	Description
0	Remotely Piloted	Human operator controls all aspects of vehicle's operation.
1	Remotely Guided	Human acting in supervisory role, vehicle is utilising on board control systems to follow pre-programmed trajectory.
2	Real-Time Health Diagnosis	Vehicle is capable of processing relevant sensor data to monitor, diagnose and identify – in real time - faults within essential vehicle subsystems. Human operator decides necessary corrective action.
3	Adapt to Environmental and Fault Disturbances	Based on the detection of faults utilising ACL 2 functionality as well as monitoring external disturbances such as ocean currents, vehicle decides if and what necessary adjustments are required to successfully complete mission
4	Situational Awareness System	Vehicle has full mission awareness including knowledge of mission objective (ACL 1), is capable of monitoring and adjusting for faults or disturbances within the system (ACL 2 & 3) and is also capable of monitoring and assessing the local operating environment to determine its effect on the vehicle being able to achieve its mission objectives.
5	On board Route Re-plan	Vehicle has the authority to utilise data obtained from ACL 4 to alter its pre-programmed route. An example scenario would be detecting and avoiding obstacles.
6	Group Coordination	First level of group autonomy. Includes the ability to organise multiple vehicles into specific formation depending on mission objectives.
7	Group Tactical Re-plan	Similar to ACL 5 where vehicles react to an unforeseen external situation and make necessary adjustments to their planned trajectory but applied to a group scenario.
8	Group Tactical Goals	At ACL 8, the group of vehicles are able to efficiently reorganise themselves based on significant events in their local operating environment such as one of the vehicles within the group failing.
9	Decentralised Mission Control	This level of autonomy requires the vehicles to be able to have the autonomy of ACL 6-8 but in a decentralised manner. To achieve this level of autonomy intra-vehicle communication must be established.
10	Group Strategic Control	Long-term, cross platform communication and strategy formulation to successfully create and carry out mission plan.
11	Fully Autonomous Systems	Ability to successfully complete ACLs 0-10 but without the supervision of human operator.

As discussed in Section 2.2 and demonstrated in Figure 2.1, the majority of AUV missions are completed by vehicles travelling unsupervised between a number of pre-programmed waypoints (Hagen et al. 2007). As such, based on the descriptions presented in Table 2.4, it can be stated that presently, the majority of AUVs can be considered to be operating at ACL 1. However, due to their ability to undertake real time health monitoring as described in (Kongsberg Maritime, 2017a) , the state of the art REMUS AUVs can be considered as having ACL 2 autonomy.

Nevertheless, although the autonomy for the vast majority of commercially available AUVs is still at ACL 1, a number of recent studies have been focussed on introducing adaptive sampling techniques to improve the overall efficiency of AUV missions. Adaptive sampling techniques, as the name suggests, uses the on board systems to process, analyse and interpret the data obtained from the vehicles sensors in real time and if necessary, make alterations to the vehicle's trajectory to ensure the data obtained is done so in an optimal manner (Chen, Pandey, & Pompili, 2012). The most popular mission scenario to use adaptive sampling is feature tracking whereby methods such as gradient ascent are used to guide the vehicle based on local fluctuations in the concentration of a measured substance (Fiorelli et al. 2006). Practical examples include chemical plume tracking and detection using the REMUS 100 vehicle (Farrell, Member, Pang, & Li, 2005) as well as tracking the evolution of the thermocline regions (Cruz & Matos, 2010b).

While the implementation of adaptive sampling improves the autonomy of AUVs considerably to ACL 5, the widespread practical implementation of real, or near real time adaptive sampling methodologies has been limited to a handful of projects (Cruz & Matos, 2010b; Farrell, Pang, Li, & Arrieta, 2003; Petillo & Schmidt, 2014; Zhang et al., 2010). The reason for this is due to two factors. The first and primary reason is related to the fact that the development of the necessary algorithms is still in its infancy and at present, the algorithms are too processor intensive (Petillo 2015) and therefore negatively impact the range of the vehicle.

The second reason, as discussed in (Cruz & Anibal Matos 2010) is related to the fact that during adaptive sampling operations, the trajectory of the vehicle is created "on the fly". As a result, there is no way to ascertain if the vehicle is operating as expected and therefore, the safe operation and recovery of the vehicle cannot be guaranteed. Furthermore, as discussed in (Petillo et al. 2012), regardless of whether or not adaptive sampling methodologies can be implemented in a safe and efficient manner, there is still an inherent problem associated with modern AUV missions – the deployment of a single vehicle. Oceanographic processes such as the development of the chemical plumes discussed above

can occur over spatiotemporal scales measured in the tens of days and thousands of kilometres (Petillo 2015). As a result, the use of a single vehicle (with a range limit of 600km at best) to track the entire plume isn't possible.

To overcome this limitation, a number of studies have investigated the feasibility of deploying a self-coordinating group of AUVs. In doing so, these groups would be able to accurately map and monitor features over spatiotemporal scales that are simply not achievable when using a single vehicle (Das et al. 2016). The following section will now provide a summary of the main outcomes from this research.

2.3.2.1 Formation Control of AUVs – Practical Experiments

The main advantage of using a group of self-coordinating AUVs is the ability it provides to allow ocean sampling to occur over spatiotemporal scales that presently cannot be completed. It is no surprise then that the Autonomous Ocean Sampling Network (AOSN) was the first collaboration to successfully deploy multiple underwater vehicles. The experiments carried out in August 2003 and explained in (Fiorelli et al. 2006) resulted in the deployment of 3 AUGs to monitor upwelling and relaxation in the Monterey Bay area.

The results demonstrated the ability of a formation control algorithm based on the Virtual Bodies and Artificial Potentials (VBAP) method to coordinate the three gliders to remain in a particular formation while maintaining a prescribed distance throughout a 16 hour deployment. The algorithms were evaluated by an external computer which used navigational data received from the gliders during periods of surfacing to produce a number of waypoints for each glider to navigate to during the subsequent submerged periods. This pattern of periodically surfacing to receive new waypoints occurred every two hours and as a result there was no communication between the individual gliders throughout the entire mission.

A second experiment was completed in Monterey Bay in August 2006 (Leonard et al. 2010) which utilised the lessons learnt from the previous deployment to increase the number of gliders in the fleet from 3 to 6. Furthermore, the mission endurance was increased from less than a day in the previous deployment to almost a full month which allowed a distance of 3270km to be covered. The coordination methodology was an adapted version of the one discussed above and is described in detail in (Zhang et al. 2007) (Paley et al. 2008).

Thereafter, in 2008, a team from Herriot-Watt University managed to use predictive methods to successfully overcome the limited communication problem to coordinate two AUVs to complete a simulated mine countermeasure mission (Sotzing, Lane 2010). The

trials were carried out in Loch Earn and demonstrated the ability of the predictive algorithm to correctly estimate the position of each vehicle in the group and thus optimise each vehicles movement. The results from the trials demonstrated that as one AUV was completing a lawnmower search pattern, the other vehicle was in a holding pattern until a simulated mine had been detected. Once detected, the AUV changed from its holding pattern to navigate to the position of the mine while the other vehicle continued its search pattern.

In 2009 the European Project GREX completed a number of sea trials relating to the coordinated movement of multiple heterogeneous AUVs using inter-vehicle communication. Although the initial sea trials described in (Kalwa 2010) highlighted the unreliable nature of the underwater communication channel with success rates varying between 18-72%, the final trials (Brignone et al. 2009) successfully demonstrated the coordination of two AUVs to complete a number of simple tasks. Furthermore, in 2010, (Smith et al. 2010) used two AUGs to cooperatively complete the tracking of a phytoplankton bloom over a period of three days.

While the projects discussed above have proved the feasibility of deploying a self-coordinating group of AUVs, there was one main limiting feature of these projects – the relatively small number of vehicles used in each deployment. The majority of the projects used only two AUVs with the exception being the AOSN projects which used six vehicles. While there is likely many reasons for this, the most prominent was likely related to budget limitations and the fact that each AUV/AUG would have cost hundreds of thousandths of dollars (Herkewitz 2013).

To combat this problem, a number of projects in the past five years have successfully designed, manufactured and produced a number of much smaller yet more manoeuvrable and affordable AUVs. Included in these vehicles is the biologically inspired vehicle, Jeff (Mintchev et al. 2014) which, with a unit cost of only €1200 is significantly cheaper than the vehicles used in the projects discussed above. Jeff was designed as part of the ambitious CoCoRo (Schmickl et al. 2011) project which, since its completion in 2014, holds the record for the largest deployment of cooperating AUVs with over forty vehicles cooperating using biologically inspired algorithms. However, it should be noted that the work completed in the CoCoRo project was done so under laboratory conditions. Nevertheless, the success of the CoCoRo project resulted in additional funding for the ongoing subCULTron project which aims to deploy a swarm of heterogeneous AUVs to complete large-scale, long-term monitoring of the canals of Venice (Thenius et al. 2016). However, despite the numerous milestones achieved by these projects, the methods used to achieve inter-vehicle

communication can be considered a limiting feature. This is due to the requirement of the vehicles to be within 1.4m of one another before successful communication can take place (Schmickl et al. 2011). Therefore, the ability to use these vehicles to monitor oceanic features occurring over large spatiotemporal scales is impractical.

Based on the above review, it is apparent that projects have been investigating and testing the feasibility of deploying groups of collaborating AUVs since the start of the 21st century. However, more importantly, in order to make large-scale, multi-vehicle deployments of AUVs economically viable, the most recent projects have replaced the large, traditional propeller driven/ buoyancy controlled vehicles shown in Figures 2.2-2.5 with smaller, less expensive vehicles such as the aforementioned Jeff vehicle and the RoboSalmon BAUV used in this thesis.

While being smaller and less expensive means these vehicles will have less individual range, their ability to operate successfully as part of a multi vehicle collaboration will allow them to cover far greater distances and map phenomena not possible with a single vehicle. As discussed in (Tena 2016), this philosophy of designing and developing smaller, less capable vehicles and deploying them as part of a collaborative group is a current research and development trend within the AUV market.

2.4 AUV System Architecture

As highlighted in (Finn & Scheding 2010), all modern AUV missions rely on the successful interaction of a number of smaller subsystems to ensure the overall mission objective can be accomplished.

For example, for the typical single vehicle AUV mission outlined in Figure 2.1, the *Guidance System* will contain a list of waypoints that the vehicle is required to sequentially manoeuvre through during the course of the vehicles deployment. At the start of the mission, the first waypoint will be used to calculate the *Demanded Heading* angle of the vehicle. This angle is then passed to the vehicles *Control System* to calculate the required *Control Surface Deflection* angle that will ensure the vehicle manoeuvres towards the desired waypoint. Throughout this process, the *Sensor* and *Navigation* subsystems continue to update the *Guidance System* with the vehicles current position and orientation. Thereafter, the *Guidance & Control* subsystems will continue to monitor whether or not any further changes to the vehicles heading angle are required and hence, any changes to the vehicles control surface. This process will continue until the vehicle is within a certain range of the first waypoint. After which, the *Guidance System* will begin to use the second point on the

pre-programmed list and this entire process will be repeated until the end of the mission. However, where appropriate and only after a human operator has decided to do so, changes to the mission plan can be made via the *Communication System*.

The above description details how the various subsystem operate together to ensure a modern AUV can safely complete its mission objective. However, for a group of vehicles to complete the mission outlined in Figure 1.3, it is apparent that instead of relying on a pre-programmed list of waypoints, the *Guidance System* will have to contain a formation control algorithm. This algorithm will have to use the positional information received from the other members of the group to ensure the parallel line formation presented in Figure 1.3 is created. Subsequently, the *Communication System* will now be vital in generating this required formation. However, before the different types of formation control algorithm are discussed, the following sections will now describe the purpose and functionality associated with the various subsystems discussed above.

2.4.1 Navigation & Sensor Subsystems

As alluded to above, the purpose of any navigation system is to provide the pilot (human or otherwise) with an accurate representation of the vehicle's position and velocity (Drapher et al. 1965) in space.

Above the Earth's watery veil, this is easily and accurately accomplished through the implementation of GPS technology (Chao et al. 2010). However, due the rapid attenuation of radio waves through seawater, GPS technology cannot be used as a reliable navigational tool in the underwater environment (Paull et al. 2014). As a result, AUVs are forced to use different navigational techniques which, as shown below, can be separated into three groups: (Stutters et al. 2008).

1. *Inertial Navigation* – This technique involves using sensors mounted within the vehicle to measure its translational and rotational accelerations. These accelerations are then integrated within a dead reckoning algorithm to obtain an estimate for the vehicle's velocity and positional data (Hegrenaes & Hallingstad 2011).
2. *Acoustic Navigation* – Navigation using acoustic techniques involves measuring the Time of Flight (TOF) of acoustic signals between the AUV and transponders. Depending on the particular type of acoustic navigation being used, these transponders will be attached to either a surface vessel or rigidly attached to the seafloor (Jakuba et al., 2008).
3. *Geophysical Navigation* – Involves using *a priori* information of geophysical parameters such as bathymetry or magnetic field strength to create an

environmental map. This map is then used to match data obtained from the vehicles sensors to ascertain the position of the vehicle (Leonard & Bahr, 2016).

While each of the above methods have their own advantages and disadvantages, the navigation technique implemented within this work is the first method described above, i.e. *Inertial Navigation*. The main reason for this choice is due to the fact that the RoboSalmon vehicle already contains the necessary hardware (Mazlan 2015) required to implement this method. Furthermore, as discussed above, acoustic navigation relies on the continuous use of acoustic modems to operate. This would only further complicate the process of using acoustic communication to allow the individual vehicles to communicate with one another as is proposed in this work. Finally, *geophysical* navigation was also ruled out due to the fact that it relies on *a priori* information to operate.

In order to incorporate a realistic representation of the inertial navigation technique described above in the work completed throughout this thesis, a mathematical model of the various sensors should be incorporated into the mathematical model of the RoboSalmon vehicle. As described in (Groves 2013), a sensor can be modelled by adding a number of error terms to the true value of the parameter being monitored. These errors include sensor biases, cross-coupling errors as well white noise. Consequently, a sensor can be modelled using the following equation.

$$\tilde{\boldsymbol{\omega}} = \boldsymbol{b}_a + \left(\mathbf{I}_3 + \mathbf{M}_g\right)\boldsymbol{\omega} + \boldsymbol{w}_g \tag{2.1}$$

Where $\boldsymbol{\omega}$ is the vector representing the *true* values of the vehicle's translational and rotational velocities, $\boldsymbol{w}_g$ represents the white noise component of the error, $\boldsymbol{b}_a$ is the biases component, $\mathbf{I}_3$ is the identity matrix, $\mathbf{M}_g$ is the cross-coupling error and finally, $\tilde{\boldsymbol{\omega}}$ is the vector of outputs from the sensor model that represents the *measured* velocities of the vehicle.

However, by incorporating a sensor model such as the one shown in Equation (2.1) would only result in the need to incorporate a suitable *state-estimation* technique to remove the errors introduced. As described in (Bryne 2017), state-estimation techniques such as the Kalman filter have been studied since the 1970's and are today considered a well-established and high performing technology capable of providing an accurate representation of a vehicle's motion based on sensor data. Therefore, by assuming that in reality, a Kalman filter could be implemented to produce an accurate representation of the vehicle's position, velocity and attitude based on sensor data, the incorporation of a suitable sensor model was deemed unnecessary. As a result, for the work completed in this thesis,

the values obtained from the mathematical model described in Chapter 3 are passed directly to the *Guidance* Subsystem.

2.4.2 Guidance & Control Subsystem

As discussed above, the main purpose of the guidance system is to process the data obtained from the navigation and sensor subsystems and determine if/what changes need to be made to the vehicles direction in order to ensure it is travelling in the correct direction (Drapher et al. 1965). In order to accomplish this, the *Guidance System* contains an *Autopilot* which houses an automatic control strategy. For example, in modern AUV missions, the *Autopilot* will contain a list of waypoints through which the vehicle must travel. Therefore, the automatic control strategy will perform the necessary calculations to ascertain whether or not the vehicle has reached the desired waypoint and if so, what changes to the vehicle's heading angle are required to manoeuvre it in the direction of the next waypoint. However, for the work completed in this thesis, the automatic control strategy will be more complicated and will involve the implementation of a formation control algorithm. The implementation of this formation control algorithm will be discussed in detail in Chapter 4.

Nevertheless, regardless of the particular automatic control strategy implemented, the *Guidance System* is required to produce a desired heading angle which the *Control System* will subsequently use to generate the required deflection angle of the vehicles control surface, i.e. RoboSalmon's caudal fin.

As presented in (Mazlan 2015), the particular controller used within the RoboSalmon vehicle to achieve this is a standard PI controller that can be represented using the following equation:

$$\delta_{TC} = K_p \Delta\psi + K_i \int \Delta\psi \, dt \tag{2.2}$$

Here, K_p and K_i are the proportional and integral gains while $\Delta\psi$ represents the difference between the desired and current heading angle of the vehicle.

At this point, it is important to note that throughout the work completed in this thesis only motion in the horizontal plane is taken into consideration. The reason for this is due to a number of factors. The first of which is that although the RoboSalmon vehicle has a pair of pectoral fins attached to it, the vehicle's relatively slow velocity results in the forces produced from these fins being insufficient to generate any significant pitching moment. As discussed in (Mazlan 2015), testing of the RoboSalmon vehicle was restricted to the

horizontal plane only and as a result, no validation data for motion in the vertical plane exists. Secondly, while it would be possible to model the vehicle's motion in the vertical plane, the majority of AUV missions involve the vehicle firstly travelling to, and then maintaining a certain depth before performing the search patterns presented in Figure 1.2 in the horizontal plane. Therefore, it was deemed an acceptable simplification to only take into consideration motion in the horizontal plane when designing the formation control algorithms contained within this thesis.

2.4.3 Communication System

Communication in the underwater domain is characterised by low bandwidth (Redfield 2013), high latency (Diamant & Lampe 2011) and unreliability (Brignone et al. 2009). These traits are a result of having to use acoustic means of communication due to the aforementioned rapid attenuation of radio waves through water. As a result, the majority of modern AUVs communicate very little, if at all, while submerged and instead, make use of periodic surfacing periods where access to radio communication can be re-established (Petillo 2015; Paull et al. 2014). However, such techniques are not feasible in multi-vehicle deployments like the one proposed in this thesis where regular communication among the vehicles is essential.

In order to facilitate efficient communication among the vehicles, a suitable *channel access method* has to be implemented. Currently, the three most popular methods are Frequency Division Multiple Access (FDMA), Time Division Multiple Access (TDMA) and Code Division Multiple Access (CDMA) (Sozer et al. 2000) (Stojanovic & Beaujean 2016). While each of these methods have their own advantages and disadvantages, the work completed in this thesis will incorporate a realistic representation of the TDMA protocol. The justification for making this decision will be discussed in detail in Chapter 6. However, it is important to note that in doing so, it is impossible for each vehicle to have a continuously accurate representation of the positional data associated with the other members of the group. In fact, it is likely that the frequency at which the vehicles will receive a communication update from the same vehicle will be measured in minutes.

In summarising, Sections 2.4.1-3 have provided an overview of the various subsystems whose interaction is required to ensure AUVs operate as expected. Furthermore, the main assumptions and simplifications made with regard to these subsystems as applied to the work completed in this thesis have also been described. As discussed above in Section 2.4.2, in order to ensure that these subsystems can generate the required parallel line formation shown in Figure 1.3, a formation control algorithm has to be implemented within the

Autopilot of the vehicles *Guidance System.* The following section will now describe the various different types of formation control algorithm available and which of these have been tested within the AUV community.

2.5 Formation Control Algorithm

As alluded to in Section 2.4.2, the aim of the formation control algorithm designed in this thesis is to allow multiple vehicles to self-coordinate and generate the parallel line formation shown in Figure 1.3. As discussed in (Ren & Cao, 1010), modern formation control algorithms can be separated into two categories depending on the method used to generate the required formation

- Formation producing algorithms: The objective of the group is to achieve a prescribed desired formation shape.
- Formation tracking algorithms: Reference trajectories are created for the individual members of the group and the agents are controlled to track these reference trajectories.

As highlighted in (Oh et al. 2015), the topic of formation control as applied to the above two categories is one of the most actively studied areas of multi-agent systems with the following review papers; (Anderson, Baris Fidan, & Hendrickx, 2008;Chen, Member, Wang, & Member, 2005; Olfati-Saber, Fax, & Murray, 2007) detailing the exhaustive literature available on the topic. The purpose of the following four sections is to briefly describe the different methods used in the creation of these different formation control algorithms as well as their associated limitations.

2.5.1 Behavioural

Behaviour based formation control algorithms take inspiration from nature and imitate the mechanisms used by either schooling fish or flocking birds (Balch & Arkin 1998). These mechanisms operate by assigning a number of behaviours to each vehicle, e.g. cohesion, collision avoidance, move to goal, obstacle avoidance and so on. The resulting behaviour used by the vehicle will depend upon a number of parameters such as distance to neighbouring vehicles, distance to obstacles or prioritising mission objectives. Consequently, the resulting behaviour of each vehicle varies depending on its local operating conditions.

The work carried out in (Balch & Arkin 1998) demonstrates the effectiveness of the behavioural methodology to coordinate a group of four vehicles to move in certain

formations to a predetermined destination while avoiding static objects. The tests have been carried out on a number of wheeled robots within a laboratory environment where the local conditions would be more favourable than those experienced underwater. Nevertheless, communication latency was taken into consideration during the tests and was found to be as high as seven seconds in some instances. Further examples of the application of behavioural based formation control algorithms can be found in (Cruz et al., 2007) (Lawton et al., 2003).

2.5.2 Virtual Structures

Virtual structures is an example of a formation tracking algorithm and are used to coordinate multiple-vehicles to manoeuvre in particular formations by creating a rigid virtual structure within which, each vehicle is ascribed their own unique reference point. As the path of the virtual structure is predefined, the individual vehicles are programmed to follow the trajectory created by the motion of their reference point on the structure as it translates or rotates as shown below in Figure 2.8 (Lewis & Tan 1997).

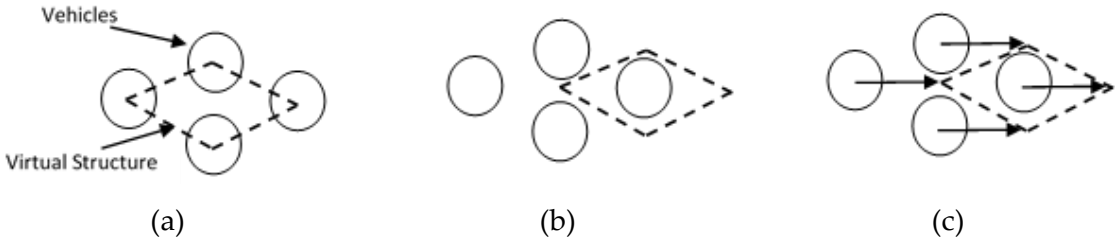


Figure 2.8- Process of virtual structure methodology - (a) Virtual Structure and vehicles are coincident, (b) Virtual Structure moves through predefined trajectory, (c) Individual Vehicles manoeuvre to stay with the Virtual Structure

While studies have shown the validity of the above method for coordinating multiple vehicles to maintain a specific formation (Li & Liu, 2008; Ramp et al., 2009), the successfulness of the methodology is dependent on prior knowledge of the interested area been known. This is a result of there being no ability to deviate from the predefined trajectory in the event of unexpected obstacles blocking the path of the individual vehicles. Therefore, as alluded to in Section 2.3.2, this inability of the vehicles to alter their trajectory severely limits the *autonomy* of the system.

2.5.3 Leader-Follower

As the name suggests, leader-follower coordination methodologies work on the premise that one vehicle within the group has a predefined trajectory and the remaining vehicles

maintain a specific distance and orientation to that leading vehicle (Cui et al. 2010). This method can be thought of as both a formation-producing algorithm and a formation tracking due to the fact that the leader follows a reference trajectory but remaining vehicles don't. Scalability of this method can also be achieved by a number of the following vehicles becoming leaders for other vehicles within the group. The simplicity of the scheme is one of its main advantages and sea trials described in (Edwards et al. 2004) have demonstrated its validity as a method to coordinate multiple vehicles.

However, this method relies on the successful operation of the leading vehicle, and while this is advantageous in certain aspects, such as the simplified communication protocols, it results in the overall system lacking redundancy (Edwards et al. 2004). On top of this, the aforementioned communication protocols are only simplified when a single leading vehicle is used due to the necessity of having to use communication scheduling when multiple leaders are introduced.

Although simplistic in nature, the reliance of this methodology on the leading vehicle is a particularly undesirable characteristic for underwater applications and one which cannot be easily rectified by the introduction of multiple leaders. In addition, although discussed in the literature, the effect of the inherent underwater communication delays on the success of the leader-follow methodology has so far not been studied (Cui et al. 2009). Furthermore, as with the *virtual structures* method, the autonomy of this method is also inherently limited by the fact that the trajectory of the *leader* is also predefined.

2.5.4 Artificial Potential Fields

Initially presented in the mid 1980's (Khatib. 1986), the Artificial Potential Field (AFP) method has since been used extensively in the robotics community for the purposes of path planning. The method operates by assigning an attractive virtual potential to destination points while associating a repulsive potential to obstacles. The merging of these two potential fields results in the vehicles being simultaneously capable of manoeuvring towards a destination point while at the same time, avoiding any obstacles within its path (Sfeir et al. 2011). While initially used exclusively for single vehicle path planning, the AFP method was extended to facilitate the coordination of multiple vehicles into various formations (Kim et al. 2006) (Ogren et al. 2004).

Included in this was the work completed by (Bennet, 2010) which demonstrated the ability of Bifurcating Potential Fields to coordinate multiple agents (vehicles) into a number of different formations through varying a number of algorithmic parameters. Included in these formations were the line and double line formations, a triangular formation, a ring

and double ring formation as well as a cluster. Furthermore, the ability of this method to not only generate a number of different formations but also simultaneously manoeuvre the vehicles to a predefined destination while avoiding obstacles was shown in (Bennet & McInnes, 2009). As well as the above, the ease at which this method could be scaled as well as its inherent reconfigurable nature were also demonstrated.

While APF methods have many advantages, including their ability to be mathematically verified (Bennet & McInnes, 2009), their successfully implementation is reliant on global information being available. This means that the formation control algorithm using an AFP method must be fully aware of the locations of all obstacles within the environment in which it is going to operate. However, as discussed in Chapter 1, this is not possible in the underwater environment and would therefore be an unrealistic assumption to make. For this reason, it was decided that the APF methodology could not be used within the work presented in this thesis.

2.5.5 Related Research – Simulation Based Studies

In Section 2.3.2.1, the research associated with formation control algorithms in the context of real-life deployments of multiple AUVs has been presented. In this section, the related research undertaken using mathematical modelling and simulation techniques will now be presented. This research is presented below in Table 2.5 and summarises the key features of each study in terms of the size of group simulated, the formation control strategy incorporated, the communication strategy used and the type of mathematical model used to simulate the dynamics of the vehicle.

What is immediately apparent from Table 2.5 is that the vast majority of the studies either did not take into consideration the delays associated with the underwater communication channel or the delays they incorporated were unrealistically small, i.e. 0.6s-5s. Furthermore, the data also illustrates that the number of vehicles simulated was also relatively small with the maximum group size only being equal to six. Nevertheless, the results presented within each study demonstrated that the chosen formation control algorithm operated as expected and the simulated vehicles were able to generate the required formation. However, the aforementioned unrealistic representation of the underwater communication channel, means that the real-life applicability of these studies cannot be verified.

Therefore, based on the data presented in Table 2.5 and the associated discussion, it's apparent that the simulation studies completed thus far do not reflect the current research and development trends in the AUV community in terms of the desire to deploy large groups of smaller and less expensive vehicles. Furthermore, the data presented in Table 2.5

also highlights that the simulation studies completed have not taken into consideration a realistic representation of the underwater communication channel. Based on this review, it is apparent that there is the need to undertake further simulations studies that not only investigate larger deployments of AUVs but also that the simulations completed represent a realistic representation of the nuances associated with the underwater channel.

Table 2.5 - Summary of simulation based research associated with formation control algorithms as applied to AUVs.

Number of Vehicles	Formation Control Strategy	Communication Delay (s)	Reference
4	Leader-Follower	0.6	(Millan et al. 2014)
3	Leader-Follower	2	(Soares et al. 2013)
3	Leader-Follower	0	(Soares et al. 2012)
6	Behavioural	0	(Yang & Zhang 2011)
3	Leader-Follower	0	(Kang et al. 2009)
4	Leader-Follower	0	(Wang et al. 2009)
3	Leader-Follower	Taken into consideration but not specified	(Burlutskiy et al. 2012)
3	Behavioural	5s	(Ghabcheloo et al. 2006)
2	Leader-Follower	0s	(Li et al. 2016)
3	Behaviour	0	(Jia & Li 2007)

2.6 Summary

This chapter has presented a review of the current state of the art associated with the development and deployment of AUVs. The brief summary of traditional propeller based AUVs presented in Section 2.2 illustrated their inherent range and endurance limitations.

As Section 2.3.1 illustrated, in an attempt to overcome these limitations, a number of projects created underwater vehicles with entirely different design philosophies. These projects resulted in the creation of two new categories of underwater vehicle, Autonomous Underwater Gliders (AUGs) and Hybrid Autonomous Underwater Vehicles (HAUVs). These vehicles, as a result of their different design philosophy, are able to cover thousands of kilometres over a period of several months during a single deployment and as a result, satisfy the requirements associated with long term, large scale oceanic mapping. Furthermore, as well as creating vehicles capable of improving the range and endurance associated with AUVs, Section 2.3.1.3 discussed the creation of a fourth category of underwater vehicle, the Biomimetic AUV. This particular type of vehicle was designed specifically to drastically improve the manoeuvrability associated with underwater vehicles. Section 2.3.1 concluded by illustrating that in terms of vehicle design, further improvements are likely only to be evolutionary and going forward, revolutionary advances in AUV operational capability are likely to come from increasing the autonomy associated with these vehicles.

This belief was further demonstrated in Section 2.3.2 where the literature review demonstrated that the current focus within the AUV community is to design smaller, cheaper AUVs with increased autonomy that are also able to operate as part of multi vehicle collaboration.

Thereafter, Sections 2.4 discussed the various subsystems and their associated functionality whose successful interaction will allow AUVs to be able to complete this collaborative autonomy. Furthermore, this section discussed the assumptions and simplifications made with regard to these subsystems as applied to the work completed in this thesis.

Section 2.5 presented the different types of formation control methodologies that can be used to achieve this collaborative autonomy as well as discussed the advantages and disadvantages associated with each. Finally, Section 2.5.1 presented a review of the different modelling and simulation studies that are associated with implementing formation control algorithms in the underwater environment. Importantly, this review illustrated that the majority of these studies did not take into consideration a realistic representation of the problematic underwater communication channel and therefore, the real life applicability of the results are debatable.

Chapter 3

RoboSalmon BAUV

3.1 Introduction

Due to the complications associated with operating in the underwater environment, an extensive campaign of algorithmic design and development is required to demonstrate whether or not it is possible to create a formation control algorithm capable of completing the mission profile outlined in Figure 1.3. In order to complete this process, the work completed in this thesis will use a pre-existing, validated mathematical model of the RoboSalmon BAUV that has been designed and built at the University of Glasgow (Mazlan & McGookin, 2012; McGookin & Watts, 2012; Watts & McGookin, 2013; Watts & McGookin, 2008).

The purpose of this chapter therefore is to present and describe this validated model and consider its applicability to be used to simulate several vehicles simultaneously. Before the mathematical model is presented though, this chapter will provide a brief overview of the RoboSalmon vehicle and the mechanisms used to allow it to imitate the swimming locomotion of the North Atlantic salmon.

To present the work described above, this chapter is structured in the following manner. Section 3.2 describes the RoboSalmon vehicle and its associated systems. Section 3.3 describes the accompanying validated mathematical model. Section 3.4 analyses it's suitability to simulate multiple vehicles simultaneously. Section 3.5 describes the modifications made to the model to reduce its complexity while Section 3.6 describes the results obtained from the necessary validation process undertaken. Finally, Section 3.7 provides a summary of the work complete

3.2 RoboSalmon BAUV

The RoboSalmon vehicle is based on the anatomy and physiology of the North Atlantic salmon (Mazlan 2015). This particular species of fish swims by means of Body and/or Caudal Fin (BCF) locomotion. As described in (Sfakiotakis et al. 1999), fish that use this particular type of locomotion do so by bending their bodies into a backward-travelling wave that extends to the caudal fin. In order to successfully replicate this motion, the RoboSalmon vehicle must also be able to generate this backword travelling wave. As shown below in Figure 3.1, in order to achieve this, the vehicle is separated into two sections: the *Body Section* and the *Tail Section*.

Figure 3.1 - Schematic diagram of RoboSalmon BAUV. (Figure has been removed due to copyright restrictions).

The *Body Section* is rigid and houses the sensors, controllers, processing units and power systems necessary to successfully operate the vehicle. The *Tail Section* meanwhile contains eight individually actuated joints whose combined motion allows the vehicle to imitate the undulatory motion of the North Atlantic salmon and therefore generate the aforementioned backward travelling wave (Mazlan 2015).

Furthermore, Figure 3.1 shows that the *Tail Section* of the vehicle accounts for approximately 50%-60% of the overall vehicle length. This has been designed specifically to ensure that the vehicle is an accurate representation of the anatomy of the North Atlantic salmon which as a result of belonging to the genus of fish called *subcarangiform*, uses approximately the same percentage of its body for propulsive purposes (McGookin & Watts 2012).

3.2.1 Actuated Tail Section

As discussed above and shown below in Figure 3.2, the *Tail Section* of the RoboSalmon vehicle consists of eight revolute joints each of which are driven by separate DC motors and connected by seven equally sized links.

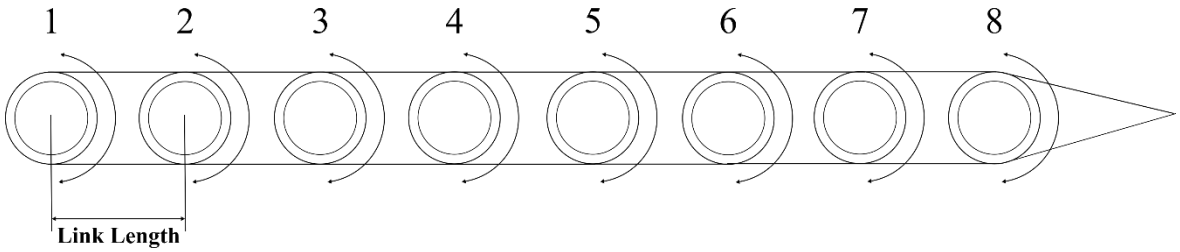


Figure 3.2 - Actuated tail section of RoboSalmon vehicle.

Figure 3.2 demonstrates that each revolute joint is free to rotate about its horizontal axis and as a result, it is possible to position the entire tail section into a number of different configurations. An example of such a configuration is shown below in Figure 3.3.

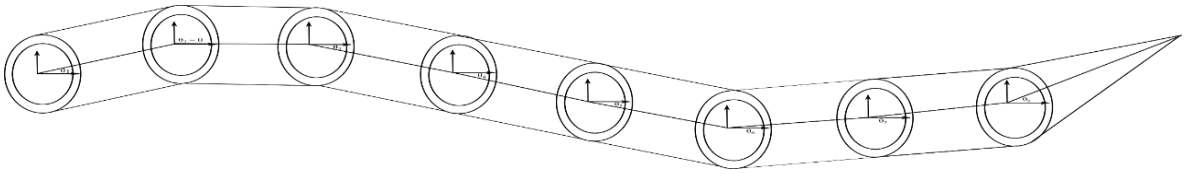


Figure 3.3- Actuated tail section replicating travelling wave approximation.

This ability to create different configurations is achieved by rotating each joint to a particular angle. This rotation results in the associated links moving laterally with the resulting combined motion of these eight joints resulting in a configuration similar to that shown above.

In order to ensure that each joint rotates to the required angular position, a control methodology has to be implemented. This is achieved by implementing a PID controller within each joint which. This controller then uses the difference between the joints actual angular position and its desired angular position to produce an input voltage to the DC motor. This voltage will then drive each individual motor to rotate towards its required angular position and hence allow the desired tail configuration to be achieved.

However, the process described above results in the *Tail Section* manoeuvring to only one of the eight configurations required to ensure the RoboSalmon vehicle can imitate one undulation of the North Atlantic salmon's swimming gait. Consequently, with the period of undulation set to one second(Mazlan 2015), this means that the desired joint angles will

change every 0.125s. As a result, the PID controller implemented (within each joint) must manoeuvre each joint to the required angle every 0.125s. After the *Tail Section* has driven the individual joints to consecutively create these eight configurations (within one second), the desired joint angles are reset to their initial values and the entire process is repeated. A flow diagram representing this process is presented below in Figure 3.4.

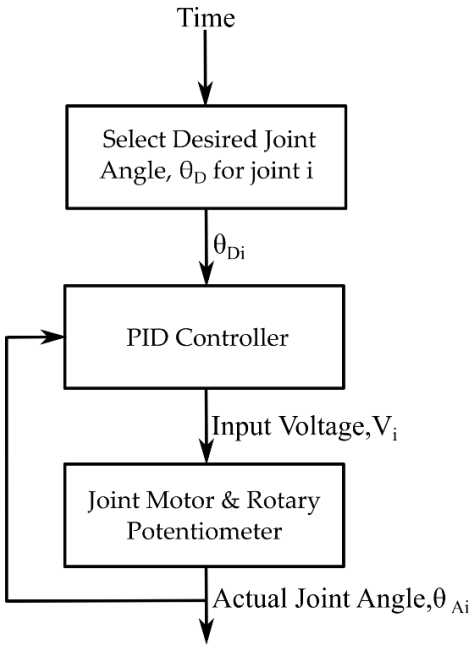


Figure 3.4- Flow diagram of tail section functionality for one joint.

As Figure 3.4 illustrates, the input to the PID controller is the desired angle for the motor associated with joint i, θ_{D_i}. Also shown is the fact that the desired joint angle selected as input to the PID controller is based on the parameter, Time. As alluded to previously, this parameter is included to ensure that every 0.125s, the desired joint angles for each motor are altered to the values required to ensure the vehicle replicates the swimming motion of the North Atlantic salmon as accurately as possible.

As highlighted in (Naddi,2015) and discussed previously in Chapter 2, although the RoboSalmon vehicle is able imitate the swimming gait of the North Atlantic salmon, the swimming speeds achieved by the vehicle are slower that those achieved by the North Atlantic salmon. The main reason for this discrepancy is due to the fact that the RoboSalmon vehicle contains only eight joints and as a result, it is impossible for it to exactly replicate the smooth swimming gait of the North Atlantic salmon. As well as being limited to only eight joints, (Naddi,2015) illustrated that actuator saturation also limits the motion of RoboSalmon's tail section. Actuator saturation in this instance, is associated with the physical limitations of the DC motors used and occurs when the PID controller produces an input voltage that is greater than the limitations of the motor. When this

occurs, the motors are no longer able to provide the required torque to rotate the joints to their desired positions.

In summarising, the above section has described the RoboSalmon vehicle and the functionality implemented within it to allow the swimming gait of the North Atlantic salmon to be imitated. The above section has also described the mechanical and electrical limitations of the vehicles design that result in it being unable to achieve the same swimming efficiency as the North Atlantic salmon.

The following section will now focus on presenting the functionality contained within the mathematical model of the vehicle. As discussed previously, this mathematical model has already been validated using experimental data obtained throughout the work completed by (Mazlan 2015).

3.3 Mathematical Model

Mathematical modelling is the process of using mathematical concepts and techniques to accurately describe the dynamics of a particular system. Today, due to the abundance of computer processing power available, mathematical modelling is used extensively in the engineering community through all phases of the design process to gain insight into a systems likely performance (Sargent, 2010). Mathematical models also provide a safe, inexpensive and effective platform to design, test and evaluate various control algorithms and theories before they are implemented onto the real system as software (Murray-Smith 2012b). For these reasons, the mathematical model of the RoboSalmon vehicle provides an ideal platform to test whether or not it is possible to design a formation control algorithm based on the behavioural mechanisms of fish that will allow the mission profile outlined in Figure 1.3 to be successfully undertaken.

However, as discussed extensively in (Murray-Smith, 2012b; Sargent, 2013), care must be taken to ensure that the mathematical model used is not only an accurate representation of the physical system being modelled but also that the model is fit for the purpose for which it has been designed for.

As the main aim of this thesis is to develop a formation control algorithm that is tolerant to the nuances of the underwater environment, it was initially envisioned that the vehicles could be represented using a simple point mass model. However, it was realised that in doing so, future investigations associated with operating vehicles as part of a formation would be limited. One such investigation envisioned was to analyse whether improvements in a vehicles range could be achieved by having the vehicles operate in close

proximity to one another. Such an investigation however, would require a mathematical representation of the vehicles drag which could not be obtained by using the aforementioned point mass model.

Consequently, the following section will now describe the pre-existing validated mathematical model of the RoboSalmon vehicle before analysing its applicability to be used to model multiple vehicles simultaneously as required for the work completed in this thesis.

3.3.1 RoboSalmon Mathematical Model Structure

As shown below in Figure 3.5, the mathematical model of the RoboSalmon vehicle can be separated into four main sections: *Tail Actuator Dynamics*, *Tail Kinematics*, *Vehicle Dynamics* and finally, the *Vehicle's Guidance System.* Although not strictly part of the mathematical model, the vehicle's *Guidance System* has been included to demonstrate the closed-loop nature of the RoboSalmon system.

As well as presenting the general structure of the mathematical model, Figure 3.5 also illustrates the rates at which these various sections are evaluated at. As shown, these range from 1000 Hz for the *Tail Actuator Dynamics* down to 4Hz for the *Guidance System*. By doing this, the mathematical model is able to evaluate the fast dynamics with a smaller time constant while the slower dynamics can be evaluated with a larger time constant. This methodology, known as multi-rate simulation, improves the computational efficiency of the model without sacrificing numerical accuracy (Gear & Wells 1984; Arnold 2007).

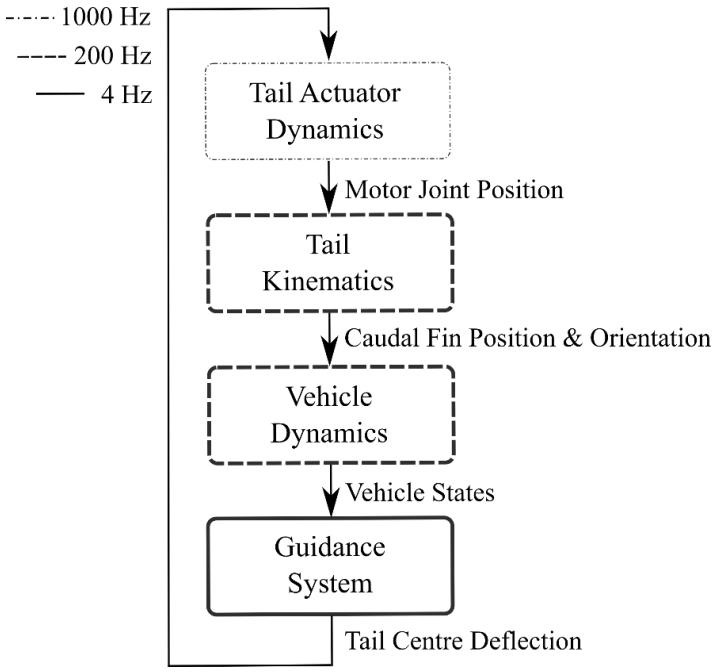


Figure 3.5 - Mathematical model structure & guidance system.

Before the mathematical model can be evaluated and analysed, an appropriate methodology has to be implemented that will allow it to be easily evaluated using a commercially available software package. The particular method chosen is the same as that used by (Mazlan, 2015; Watts, 2009) and is referred to as state-space modelling(Rowell 2002).

3.3.2 State Space Models & Simulation

State space modelling or the state space representation as it is often referred to as, is a particularly convenient way to represent a mathematical model in order to allow it to be easily evaluated using a commercially available software package such as MATLAB. The method operates by representing the dynamics of the system through a number of coupled first order differential equations which can be represented using the equation shown below (Rowell 2002):

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{3.1}$$

Here, $\mathbf{x}$ and $\dot{\mathbf{x}}$ are the state and state derivative vectors with the state vector, $\mathbf{x}$ representing the outputs of the mathematical model. $\mathbf{A}$ and $\mathbf{B}$ are known as the process and input matrices while $\mathbf{u}$ is the input vector. Depending on the order of the system being modelled, the size of the aforementioned vectors and matrices will vary. For example, for an n^{th} order system with r inputs, $\mathbf{x}$ and $\dot{\mathbf{x}}$ will both be column vectors of length, n, $\mathbf{A}$ will be a n-by-n square matrix, $\mathbf{B}$ will be an n-by-r matrix and $\mathbf{u}$ will be a column vector of length r.

Due to state-space modelling being able to evaluate n^{th} order systems, it can be used to evaluate complex, multiple input, multiple output systems (MIMO Systems) and as a result, is used frequently in the modelling and simulation of marine vehicles (Murray-Smith 2012a; McGookin 1997; Mazlan & McGookin 2012; Fossen 2011).

In order to use Equation (3.1) and the state space representation to evaluate a particular mathematical model, a number of procedural steps must be adhered to. These steps are shown below in Figure 3.6.

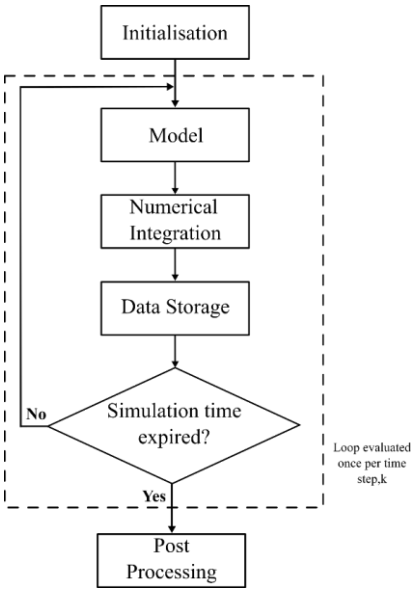


Figure 3.6- Flowchart of state space modelling technique.

Firstly, at the beginning of each simulation within the *Initialisation* section, the column vectors, **x** and **u** are populated with their initial conditions while the matrices **A** and **B** are populated with their constant values. Then, using Equation (3.1) in the *Model* section, the state derivative vector, $\dot{\mathbf{x}}$ is evaluated. This vector is then passed to the *Numerical Integration* section where it is numerically integrated to obtain new values for each of the states contained within the vector, **x**. These states are then stored in separate output vectors within the *Data Storage* section to provide a time history of the values calculated for each state. Thereafter, if the simulation time has not expired, Figure 3.6 illustrates that Equation (3.1) is evaluated once again except this time with the updated values for the state vector **x** calculated previously in the *Numerical Integration* section. These updated values are then used to obtain new values for each of the state derivatives which are then integrated to obtain a new state vector **x**. As Figure 3.6 demonstrates, this entire process is repeated until the end of the simulation. Once the simulation time has expired, the data contained within the *Data Storage* section will contain a complete time history of the values calculated for each state. These time histories can then be used to analyse the dynamics of the system being modelled

3.3.3 Simulation Variables & Reference Frames

In order to generate a suitable mathematical model of RoboSalmon, the kinematics of the vehicle must be defined. The first step in this process is to define the frames of references for the vehicle's velocity, position and orientation in three dimensional space (Fossen 1994). To achieve this, two reference frames are implemented: the *body-fixed* reference frame and the *Earth-fixed* reference frame. As the name suggests, the *body-fixed* reference frame is

attached to, and moves with the vehicle as it manoeuvres through space. The *Earth-fixed* reference frame is thought of as the inertial frame of reference and is therefore fixed to a point in space (Stevens & Lewis 2003). These reference frames and the associated notation are presented below in Figure 3.7.

As the motion of the tail and the subsequent forces generated are confined to the vehicle, these are evaluated with respect to the *body-fixed* frame of reference. For the purposes of guidance, navigation and control, it is more appropriate to know the position and orientation of the vehicle in the *Earth-fixed* reference frame (Cai et al. 2011). This means that the velocities and accelerations calculated with respect to the *body-fixed* reference frame must be translated into their equivalent parameters in the *Earth-fixed* reference frame using Euler's rotational theorem (Etkin & Reid 1996).

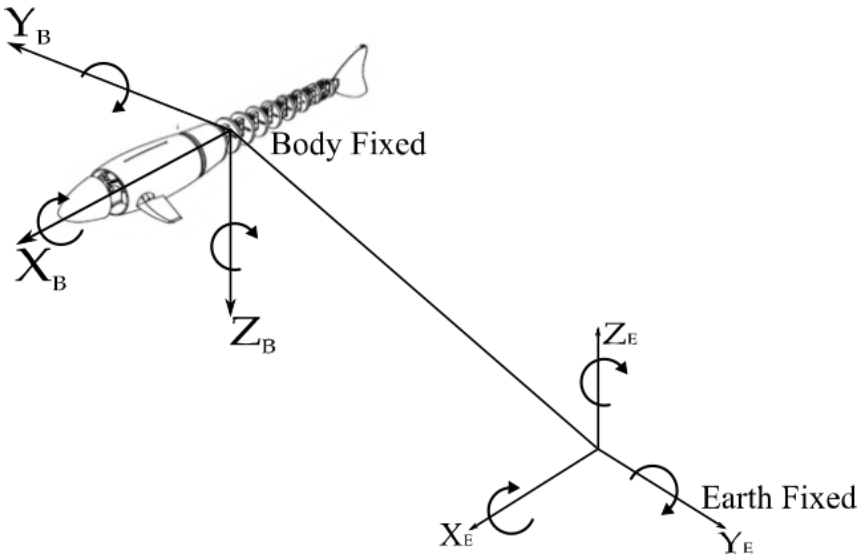


Figure 3.7- Body & Earth fixed reference frames used.

Furthermore, Figure 3.7 demonstrates that within the *body-fixed* reference frame, the RoboSalmon vehicle has six degrees of freedom - one rotational and one translational for each of the three dimensions shown. Associated with each of these degrees of freedom will be a force and translational velocity as well as a moment and rotational velocity. These forces, moments and velocities are used to describe the dynamics of the vehicle and the standard notation used to describe them, created by the Society of Naval Architects and Marine Engineers (SNAME) (Fossen 1994) is presented below in Table 3.1.

Table 3.1- SNAME notation used to define motion of marine vehicles

Degree of Freedom	Motion Description	Name	Forces & Moments	Linear and Angular Velocities	Position and Euler Angles
1	Motion in x-direction	Surge	X (N)	*u (m/s)*	x (m)
2	Motion in y-direction	Sway	Y (N)	*v (m/s)*	y (m)
3	Motion in z-direction	Heave	Z (N)	*w (m/s)*	z (m)
4	Rotation about x-axis	Roll	K (Nm)	*p (rad/s)*	Φ (rad)
5	Rotation about y-axis	Pitch	M (Nm)	*q (rad/s)*	Θ (rad)
6	Rotation about z-axis	Yaw	N (Nm)	*r (rad/s)*	Ψ (rad)

In summarising, the above two sections have not only described the state space representation used to evaluate the mathematical model but also the reference frames implemented to accurately evaluate not only the forces, accelerations and velocities of the vehicle but also its position and orientation in space. Building on this, the following sections will now describe the functionality contained within the various subsections of the mathematical model shown previously in Figure 3.5.

3.3.4 Tail Actuator Dynamics

As discussed in Section 3.2.1, the tail section of the RoboSalmon vehicle is comprised of eight revolute joints with each containing a separate DC motor and PID controller. To accurately model the motion of these joints, the functionality contained within this section of the model not only has to contain a mathematical representation of the DC motors but also an identical version of the PID controller implemented within the vehicle. As shown below in Figure 3.8, in order to achieve this, the functionality implemented is separated in two subsystems within a closed-loop feedback control structure.

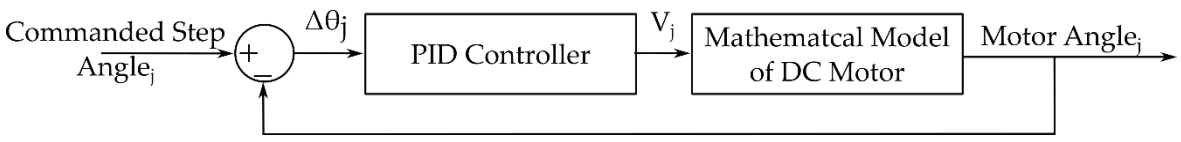


Figure 3.8- Structure of functionality within *Tail Actuator Dynamics* subsystem.

Figure 3.8 demonstrates that the input to the *Tail Actuator Dynamics* section of the model is a commanded step angle for each motor, j. Once again, this angle has been specifically chosen to ensure that the *Tail Section* of the vehicle is able to imitate the swimming gait of the North Atlantic salmon as accurately as possible. This angle, as shown above, is then compared with the actual angular position of motor j to produce an error, $\Delta\theta_j$ which forms the input to the control system. As discussed above, this control system is a standard PID controller which can be represented using the following equation:

$$V_j = K_p \Delta\theta_j + K_i \int \Delta\theta_j \, dt + K_d \frac{d\Delta\theta_j}{dt} \tag{3.2}$$

Here, $\Delta\theta_j$ represents the aforementioned error between the required and actual motor angle, V_j is the required voltage for motor j to achieve its desired angular position and K_p, K_i and K_d are the proportional, integral and derivative gains of the control system.

Up to this point, the functionality (Equation (3.2)) and processes discussed are identical to those implemented within the control architecture of RoboSalmon's *Tail Section*. In order to model the motion of each revolute joint, a mathematical representation of the dynamics of the DC motors has to be created. To achieve this, the electrical and mechanical properties of the DC motors are modelled using the equations shown below which combine Netwon's and Kirkhoff's laws and have been taken from (Franklin, G., Powell, J., Emama-Naeini 1991):

$$\mathrm{L}\frac{\mathrm{di_j}}{\mathrm{dt}} + \mathrm{Ri_j} = \mathrm{V_j} - \mathrm{K}_e\dot{\theta}_\mathrm{j} \tag{3.3}$$

$$\mathrm{J}\ddot{\theta}_\mathrm{j} + \mathrm{b}\dot{\theta}_\mathrm{j} = \mathrm{K_t i_j} \tag{3.4}$$

Here, L is the motor inductance, i_j is the current for motor j, R is the resistance, V_j is the voltage applied to motor j (evaluated using Equation (3.2)), $\dot{\theta}_j$ and $\ddot{\theta}_j$ are the motor angular velocities and accelerations, respectively, J is the moment of inertia, b is a damping constant, $K_e$is a motor constant, and K_t is the torque coefficient.

Based on the discussion presented in Section 3.3.2, in order to evaluate the angular positions of each motor using the equations presented above, they have to be rearranged into their equivalent state space representation. This is shown below in Equation (3.5).

$$\dot{\mathbf{x}} = \begin{bmatrix} -\frac{R}{L} & 0 & -\frac{K_e}{L} \\ 0 & 0 & 1 \\ \frac{K_t}{J} & 0 & -\frac{b}{J} \end{bmatrix} \mathbf{x} + \begin{bmatrix} \frac{1}{L} \\ 0 \\ 0 \end{bmatrix} \mathbf{u} \tag{3.5}$$

Where the state vector **x** in Equation (3.5) is defined as:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} i_j \\ \theta_j \\ \dot{\theta}_j \end{bmatrix} \tag{3.6}$$

Equation (3.5) demonstrates that the system of equations representing the electrical and mechanical properties of the DC motor can be reduced into a state space representation consisting of a number of coupled first order differential equations. As a result, the state derivatives given in Equation (3.5) can now be numerically integrated to obtain values for the angular position of each motor, θ_j based on the input voltage V_j supplied to the motor via the PID controller. Finally, it is important to note that as discussed in Section 3.2.1, the Commanded Step Angle for each motor, j changes every 0.125s in order to ensure the vehicle replicates the swimming gait of the North Atlantic salmon.

3.3.5 Tail Kinematics

Although the functionality described in the previous section allows the angular position of the individual joints to be estimated, the diagram shown below in Figure 3.9 demonstrates these angular positions are evaluated with respect to individual references frames associated with each revolute joint.

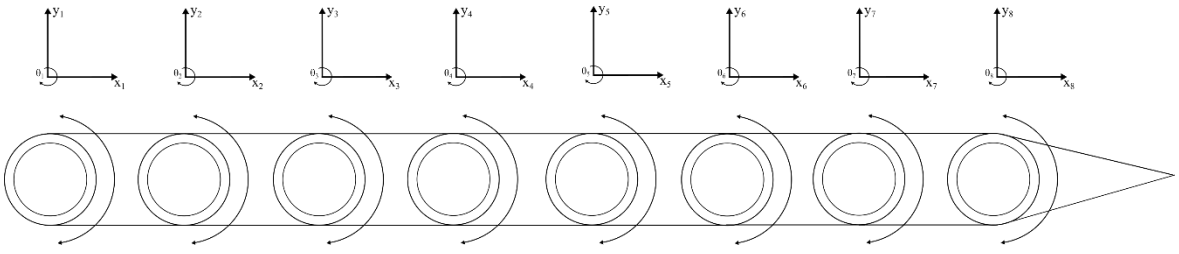


Figure 3.9 - Individual reference frames of tail section.

Therefore, a methodology has to be implemented that will translate these angular positions into their corresponding displacements in the *body-fixed* axis. To achieve this, a simplified representation of the Denavit-Hartenberg (Niku 2001) convention is implemented. As the *Tail Section* of the RoboSalmon vehicle can be thought of as a robotic manipulator, a simplified representation of the Denavit-Hartenberg convention provides a valid method by which the lateral displacement of the individual joints, as well as the caudal fin, in the

body-fixed axis, can be evaluated (Niku 2001). The simplified representation can be used due to the assumption that there is no link offset or twist associated with each revolute joint. After applying these simplifications, the D-H convention operates by applying the transformation matrix shown below to each of the links shown above in Figure 3.9.

$$ {}^{j-1}_{j}T = \begin{bmatrix} c\theta_j & -s\theta_j & 0 & a_{j-1} \\ s\theta_j & c\theta_j & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.7} $$

Here, j represents the revolute joint (motor) for which the transformation matrix between itself and the previous joint is being evaluated for, *c* and *s* represent the cosine and sine of the angle, θ_j (Equation (3.6)) and *a* represents the link length. To evaluate the position of each joint in the *body-fixed* axis, the matrix presented above is populated with the appropriate values and used in conjunction with the following equation:

$$ \mathbf{T}_{\mathrm{j}} = \prod_{j=1}^{j} {}^{j-1}_{j}\mathbf{T} \tag{3.8} $$

This multiplication of matrices produces a 4x4 square matrix of the following form:

$$ T_j = \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.9} $$

Where the elements r, represent the relative orientation of the two joints and the P elements represent the longitudinal, lateral, and vertical position of the jth joint with respect to the *body-fixed* axis.

Therefore, in order to evaluate the position of the caudal fin in the *body-fixed* axis, Equation (3.8) is extended to include all eight matrices representing the transformation between neighbouring reference frames as demonstrated below in Equation (3.10).

$$ \mathbf{T}_{9\,(\text{Caudal Fin})} = {}^{0}_{1}\mathbf{T}\,{}^{1}_{2}\mathbf{T}\,{}^{2}_{3}\mathbf{T}\,{}^{3}_{4}\mathbf{T}\,{}^{4}_{5}\mathbf{T}\,{}^{5}_{6}\mathbf{T}\,{}^{6}_{7}\mathbf{T}\,{}^{7}_{8}\mathbf{T}\,{}^{8}_{9}\mathbf{T} \tag{3.10} $$

In summarising, the above two sections have described the functionality contained within the mathematical model that allows the motion of the *Tail Section* of the RoboSalmon vehicle to be modelled. This functionality included elements that have been implemented within the physical RoboSalmon vehicle such as the PID controller and the associated gains as well as modelling elements, such as the mathematical representation of the electrical and

mechanical dynamics of the DC motors. Furthermore, the functionality implemented within the model to transform the angular positions of each revolute joint into its lateral and longitudinal position in the *body-fixed* axis has also been described. As a result, the motion of the caudal fin can now be estimated which, as shown in Figure 3.5, forms the input to the next main subsystem of the mathematical model, the *Vehicles Dynamics.*

3.3.6 Vehicle Dynamics

Regardless of the particular type of vehicle being modelled, the mathematical models used to evaluate their dynamics will likely be derived from Newton's 2nd Law. Using this approach, the dynamics of any marine vehicle can be represented using Equation (3.11) shown below.

$$\mathbf{M}\dot{\boldsymbol{v}} + \boldsymbol{C}(\boldsymbol{v})\boldsymbol{v} + \boldsymbol{D}(\boldsymbol{v})\boldsymbol{v} + \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} \tag{3.11}$$

Where normally, **M** would represent the rigid body mass and inertia properties of the system being modelled. However, as explained in (Fossen,1994) for marine vehicles, an additional component has to be included in this term to account for the additional inertia added to the system as a result of it displacing a certain volume of the local fluid medium it operates in. This additional term also has to be added to the Coriolis and Centripetal matrix, **C.** Therefore, for the purposes of describing the dynamics of the RoboSalmon vehicle, the matrices **M** and **C** can be thought of as consisting of two parts: rigid body inertia and added mass inertia.

While the components, **M** and **C** are created as a product of the derivation of the standard 6 DoF rigid body equations of motion, the remaining vectors $\boldsymbol{D}$,$\boldsymbol{g}$ and $\boldsymbol{\tau}$ represent the external forces and moments acting on the vehicle. These include the hydrodynamic damping forces, $\boldsymbol{D}$ such as the vehicles drag, the restoring forces, $\boldsymbol{g}$ due to the balance between the gravitational and buoyancy forces acting on the vehicle and finally, the vehicles inputs forces and moments, $\boldsymbol{\tau}$ created from its propulsive and control surface subsystems.

The vectors $\dot{\boldsymbol{v}}$ and $\boldsymbol{v}$ represent the accelerations and velocities of the vehicle as calculated in the *body-fixed* axis while $\boldsymbol{\eta}$ represents the position and orientation of the vehicle in the *Earth-fixed* axis.

As with the equations used to model the dynamics of the DC motors, Equation (3.11) has to be rearranged into its equivalent state-space representation before it can be numerically integrated. This state-space form is shown below in Equation (3.12).

$$\dot{\mathbf{v}} = \mathbf{M}^{-1}\left(-(\mathbf{C}(\boldsymbol{v}) + \mathbf{D}(\boldsymbol{v}) + \mathbf{g}(\boldsymbol{\eta})\boldsymbol{v}^{-1})\right)\boldsymbol{v} + \mathbf{M}^{-1}\boldsymbol{\tau} \quad (3.12)$$

The mathematical functionality and equations contained within the matrices **M, C, D, g** and $\boldsymbol{v}$ are all detailed extensively in both (Mazlan, 2015; Watts, 2009) as well as Appendix A of this Thesis.

In order to demonstrate where the outputs from the *Tail Kinematics* section of the model are used within the *Vehicle Dynamics* section, the following section will now describe the functionality contained within the vector, $\boldsymbol{\tau}$ which, as mentioned above, represents the external forces and moments acting on the vehicle.

3.3.6.1 Description of External Forces & Moments

The RoboSalmon vehicle generates its propulsive thrust by using its *Tail Section* to imitate the swimming gait of the North Atlantic salmon and thus, produce the same propulsive backward travelling wave that exists in Body and/or Caudal Fin locomotion (Sfakiotakus et al; 1999). Consequently, a suitable mathematical methodology has to be implemented that will produce an estimate for the thrust created from this motion.

Due to its intricate nature, there exists very few methods to accurately estimate the thrust generated (Sfakiotakus et al; 1999) by fish locomotion. One such method that has been used and is particularly suited to estimating the thrust generated from *subcarangiform* locomotion is known as Lighthill's *Large Amplitude Elongated Body Theory* (Lighthill 1971). As described by (Ellerby 2010), this method estimates the thrust by modelling the transfer of momentum between the fish and the surrounding fluid using the equation shown below.

$$T = mwk\ sin\,\theta_{CF} + \frac{1}{2}mw^2\ cos\,\theta_{CF} \quad (3.13)$$

At this point, it should be highlighted that w and k represent the perpendicular and tangential velocities of the caudal fin tip and θ_{CF} represents the angle between the caudal fin tip and the x-axis. This is because Lighthill's theory assumes that the thrust generated is concentrated at the trailing edge of the tail (Tytell 2004), i.e. the caudal fin tip. Therefore, it is within the above equation that the position and orientation of the caudal fin as calculated using Equation (3.11) is used within the *Vehicle Dynamics* section of the model.

Once the thrust force has been evaluated using Equation (3.13), it has to be separated into its x and y components. To achieve this, the following equations are utilised:

$$X_T = T\ cos\,\theta_{TC}$$

$$Y_T = T\,sin\,\theta_{TC}$$
$$N_T = Y_T d_{c.g.} \tag{3.14}$$

Here, X_T and Y_T represent the thrust forces acting in the longitudinal and lateral directions respectively, θ_{TC} represents the angle between the centreline of the tail section and the x-axis, $d_{c.g.}$ is the longitudinal distance between the caudal fin and the position of the vehicle's centre of gravity and finally, N_T is the resulting yawing moment created due to the lateral component of the thrust force and the moment arm, $d_{c.g.}$.

3.3.7 Vehicle Kinematics

In order to translate the *body-fixed* velocities calculated by numerically integrating Equation (3.12) into their *Earth-fixed* equivalents, a transformation is required. As discussed in Fossen (1994), this transformation is made up of a series of angular rotations that must be completed in a particular manner as outlined in *Euler's Theorem on Rotation.*

The transformations can be separated into their translational and rotational components and represented mathematically using the following equations.

$$\dot{\boldsymbol{\eta}}_{\mathbf{1}} = \mathbf{J_1}(\boldsymbol{\eta}_{\mathbf{2}})\mathbf{v_1}$$
$$\dot{\boldsymbol{\eta}}_{\mathbf{2}} = \mathbf{J_2}(\boldsymbol{\eta}_{\mathbf{2}})\mathbf{v_2} \tag{3.15}$$

Here

$$\boldsymbol{\eta}_{\mathbf{1}} = [x_e, y_e, z_e]'; \quad \boldsymbol{\eta}_{\mathbf{2}} = [\phi, \theta, \psi]';$$
$$\mathbf{v_1} = [u, v, w]'; \quad \mathbf{v_2} = [p, q, r]'; \tag{3.16}$$

In Equation (3.15) $\mathbf{J}_1$ and $\mathbf{J}_2$ represent the Euler transformation matrices for the linear and rotational velocities respectively and are shown below in Equations (3.17) and (3.18).

$$\mathbf{J_1}(\boldsymbol{\eta}_{\mathbf{2}}) = \begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi c\phi s\theta \\ s\psi c\theta & c\psi c\phi + s\phi s\theta s\psi & -c\psi s\phi + s\theta s\psi c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \tag{3.17}$$

$$\mathbf{J_2}(\boldsymbol{\eta}_{\mathbf{2}}) = \begin{bmatrix} 1 & s\theta t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & \frac{s\phi}{c\theta} & \frac{c\phi}{c\theta} \end{bmatrix} \tag{3.18}$$

By rearranging Equation (3.17), it is now possible to represent the above transformation in the desired state-space form:

$$\begin{bmatrix}\dot{\boldsymbol{\eta}}_1\\ \dot{\boldsymbol{\eta}}_2\end{bmatrix} = \begin{bmatrix}\mathbf{J_1}(\boldsymbol{\eta_2}) & \mathbf{0_{3x3}}\\ \mathbf{0_{3x3}} & \mathbf{J_2}(\boldsymbol{\eta_2})\end{bmatrix}\begin{bmatrix}\mathbf{v}_1\\ \mathbf{v}_2\end{bmatrix} \tag{3.19}$$

As with Equation (3.12), these velocities are numerically integrated to obtain not only the position of the vehicle in the *Earth-fixed* reference frame but also its orientation.

Finally, by combining Equation (3.19) with that of Equation (3.12), it is now possible to express the entire RoboSalmon mathematical model in the required state-space representation as shown below in Equation (3.20).

$$\begin{bmatrix}\dot{\mathbf{v}}\\ \dot{\boldsymbol{\eta}}\end{bmatrix} = \begin{bmatrix}\mathbf{M}^{-1}\left(-(\mathbf{C}(\mathbf{v}) + \mathbf{D}(\mathbf{v}) + \mathbf{g}(\boldsymbol{\eta})\mathbf{v}^{-1})\right)\boldsymbol{v}\\ \mathbf{J}(\boldsymbol{\eta})\end{bmatrix}\begin{bmatrix}\mathbf{v}\\ \boldsymbol{\eta}\end{bmatrix} + \begin{bmatrix}\mathbf{M}^{-1}\\ 0\end{bmatrix}\boldsymbol{\tau} \tag{3.20}$$

In summarising, Section 3.3 has provided an overview of the structure and functionality contained within the validated, high-fidelity mathematical model of the RoboSalmon vehicle. The following section will now analyse the run time performance of this model with respect to its ability to simultaneously simulate multiple vehicles.

3.4 Mathematical Model Performance

The performance of a mathematical model is often measured with respect to its ability to accurately represent the dynamics of the system being modelled (Murray-Smith, 2012). It is generally accepted that in order to improve the model's accuracy, the amount of mathematical detail and complexity implemented within it has to increase (Zeigler, Kim, Praehofer, 2000). Unsurprisingly, this has the undesirable effect of increasing the time taken to evaluate the model, i.e. its execution time. However, (Sargent, 1999) suggests that instead of simply creating an overly complex representation of the system, a model should be developed for a specific purpose and its validity measured with respect to that purpose. This is further emphasised by both (Murray-Smith, 2012) and (Brooks & Tobias 1996) who state a successful mathematical model often involves a trade-off between the level of detail included, the corresponding speed of solution (execution time) and the model's accuracy. As shown in (Mazlan 2015), while the ability of the model described in Section 3.3 to accurately model the RoboSalmon vehicle has already been demonstrated, its applicability to be used to simulate multiple vehicles simultaneously in a time efficient manner has yet to be established.

To achieve this, the mathematical model described in Section 3.3 has been altered to allow it to model multiple vehicles simultaneously. The subsequent relationship between the

number of vehicles modelled and the associated execution time of the model is presented below in Figure 3.10.

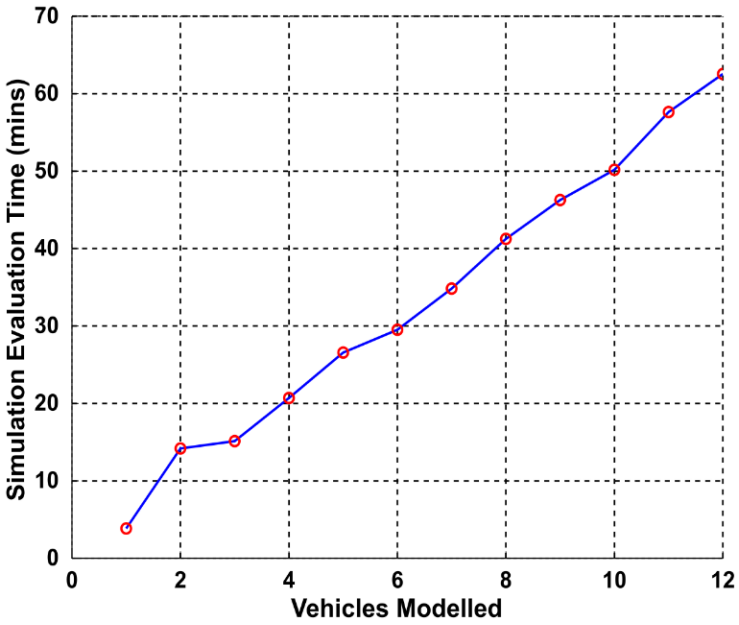


Figure 3.10 - Relationship between the number of vehicles modelled & simulation execution time

While it is unsurprising that the magnitude of the execution time increases as the number of vehicles modelled increases, Figure 3.10 demonstrates that on average, for each additional vehicle incorporated, the simulation time increases by approximately five minutes. This results in an execution time of over an hour for a scenario lasting only 10 minutes and including 12 vehicles. Therefore, given that the work completed in this thesis aims to investigate scenarios involving this number of vehicles, the mathematical model, in its current guise, isn't suitable for the subsequent work that will be undertaken in this research.

As a result, modifications will have to be made to the model in order to produce an improvement in its execution time. Before these alterations are made though, it is important to analyse the model in greater detail to understand which of its subsections are particularly inefficient. To achieve this, shown below in Figure 3.11 is the execution time for these various subsections as a percentage of the total execution time of the model.

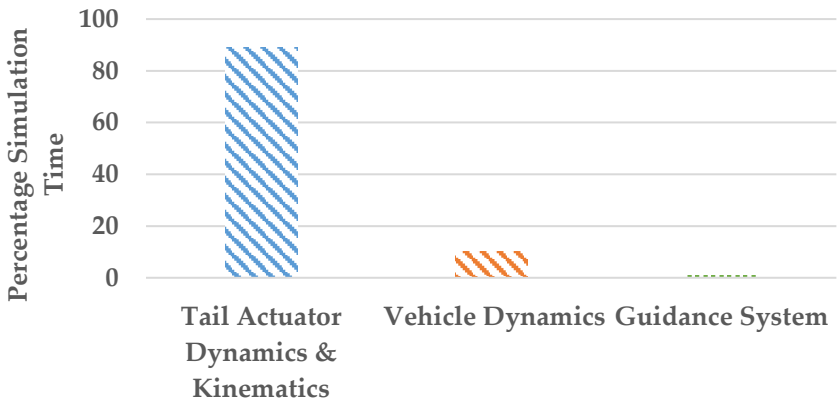


Figure 3.11 – Percentage of simulation time used to evaluate functionality contained within various subsystems.

The results presented above clearly demonstrate that the vast majority of the models execution time is attributed to evaluating functionality within the *Tail Actuator Dynamics & Kinematics* sections of the model. The reasons for this being the case are due to a number of factors. The first of which is due to the fact that the functionality contained within this section of the model is evaluated at a rate of 1000 Hz. Secondly, the requirement to use the 4th Order Runge-Kutta algorithm to numerically integrate Equation (3.5) results in these equations being evaluated four times every one thousandth of a second. Finally, the fact that the vehicles *Tail Section* is comprised of eight motor joints means that the functionality described above is evaluated 32 times every 0.001s. This means that overall, the functionality contained within this section of the model is evaluated 32,000 times per second.

In order to substantially improve the execution time of the model, it is apparent that the functionality contained within this section of the model will have to be either modified or replaced. As the remaining sections of the model, i.e. the *Vehicle Dynamics* will remain unchanged, the techniques implemented to replace this functionality will have to produce the exact same outputs as the original model and as shown in Figure 3.5, these outputs are the position and orientation of the caudal fin.

3.4.1 Model Categorisation

As discussed in (Sjöberg et al. 1995) and presented below in Table 3.2, mathematical models can be categorised into three groups depending on the level of prior knowledge used in their construction. In this work, prior knowledge represents the implementation of established physical relationships such as Newton's 2nd Law within the mathematical model.

Table 3.2- Categorisation of Mathematical Model Structures.

Model Category	Category Description
White-Box	The mathematical model is perfectly known; it has been possible to construct it entirely from prior knowledge and physical insight
Grey-Box	Some physical insight is available, but several parameters remain to be determined from observed data.
Black-Box	No physical insight is available or used, but the chosen model structure belongs to families that are known to have good flexibility and have been 'successful in the past'

Regardless, based on the descriptions presented below, the RoboSalmon model presented thus far can be categorised as a white-box model due to the fact that it's been derived entirely from first principles using Newton's 2nd Law of Motion for the vehicle dynamics and Kirchoff's law for the dynamics associated with the DC motors. As explained in the description section of Table 3.2, these types of models provide the user with the greatest physical insight into the system being modelled but in many cases, can be overly complex.

Black box models meanwhile provides the user with zero physical insight and instead operate by using measured experimental data to create a mathematical structure that is able to map a systems response entirely from its inputs (Ljung 2001). This process of creating the relevant mathematical structure based on a system's inputs and outputs is known as System Identification (Bai & Giri 2010; Ljung 1999). As opposed to the mathematical modelling procedures used in the creation of white-box models, black-box modelling can be thought of as more of a curve-fitting procedure. Nevertheless, when implemented correctly, this procedure has been shown to be capable of accurately modelling the dynamics of many complex systems by incorporating simple mathematical basis functions (Nielsen & Madsen 2006; Jin et al. 2001).

Grey-box modelling incorporates aspects of both white and black box modelling techniques to combine the individual strengths of the two methods with the aim of producing the most accurate and efficient representation of a systems dynamics (Bohlin 2006).

Due to the fact that they are created using simple mathematical basis functions (Hauth 2008), it is proposed that the functionality contained within the *Tail Actuator Dynamics &*

Kinematics subsections be replaced with a black-box model representation. In doing so, it is anticipated that the execution time of the model can be improved while maintaining the validity of the original validated model.

3.5 Reducing Model Complexity

The aim of this section is to ascertain whether or not the system identifications techniques discussed above can be implemented to create a black-box model to replace the processor intensive functionality contained within *Tail Actuator Dynamics & Kinematics* subsection of the RoboSalmon mathematical model. To begin, shown below in Figure 3.12 is the proposed changes to the structure of the mathematical model.

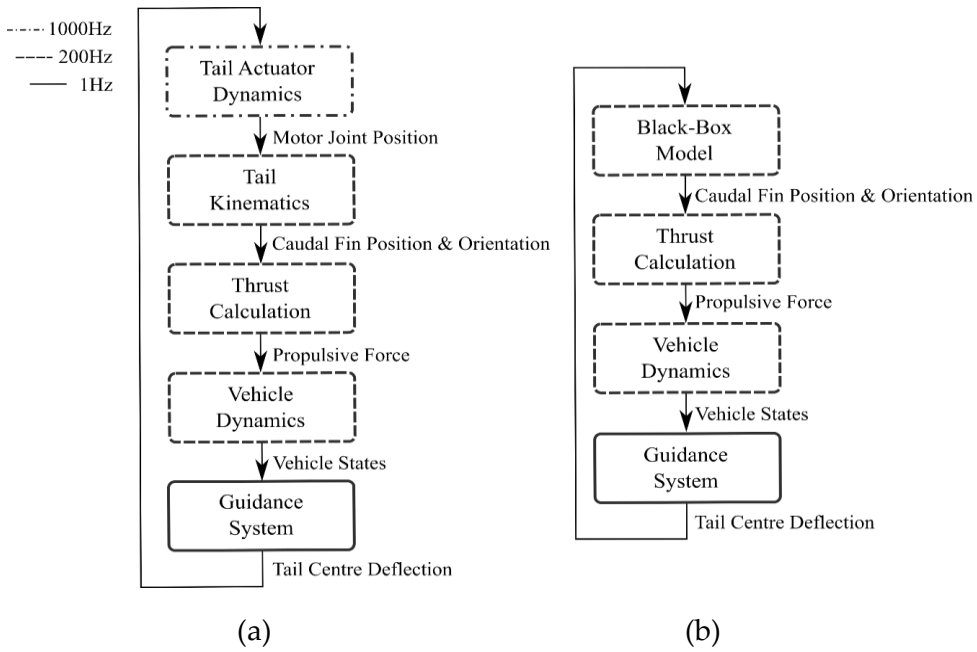


Figure 3.12- Comparison of (a) original mathematical model structure (b) proposed model structure.

On inspection of Figure 3.12, the input to the proposed black box model will be the required tail centre deflection angle, δ_T of the vehicle (its rudder angle) while its outputs, as shown above, will be the position and orientation of the caudal fin. Before a decision can be made as to whether or not system identification techniques can be implemented to replace this functionality, the relationship between these inputs and outputs needs to be analysed.

To achieve this, open loop simulations of the original, validated model where completed that varied the value for δ_T (Input) in 30º increments throughout its entire operation range. The associated position and orientation (Outputs) of the caudal fin were then recorded and the mapping between these inputs and outputs are presented below in Figure 3.13.

A number of observations can be made with regard to the motion of the caudal fin as δ_T is altered. The first and perhaps most obvious is the cyclical nature of the caudal fin's response. However, given the undulatory nature of fish locomotion, this is not at all surprising. More importantly though, Figure 3.13 demonstrates that when averaged over a single tail beat (one second interval), the position of the caudal fin, i.e. how much it deviates from the vehicle's centreline, is proportional to δ_T. Again, this is unsurprising, as a non-zero δ_T will move the position of the caudal fin laterally to either the right or left hand side of the centreline. Intuitively, the greater the deflection angle, the greater the lateral distance between the caudal fin and the vehicles centreline.

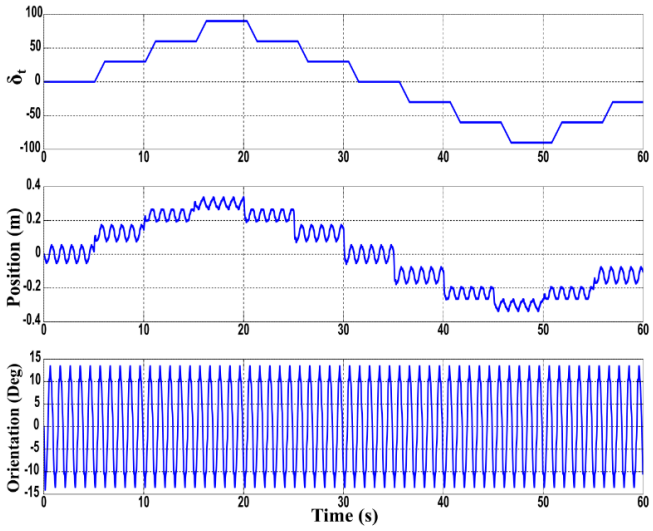


Figure 3.13 - Evolution of caudal fin position for various δ_T values throughout the vehicle's operation range.

Upon further analyses of Figure 3.13, it is apparent that in order to create a black-box model that is capable of representing the evolution of the caudal fin's position throughout its entire operational window, an additional input parameter on top of δ_T will have to be used. This is due to the fact that only through a combination of two parameters can each of the transient and steady-state motions presented above be uniquely identified. This is demonstrated below in Figure 3.14 where these two input parameters are compared with the corresponding output (caudal fin position) of the system.

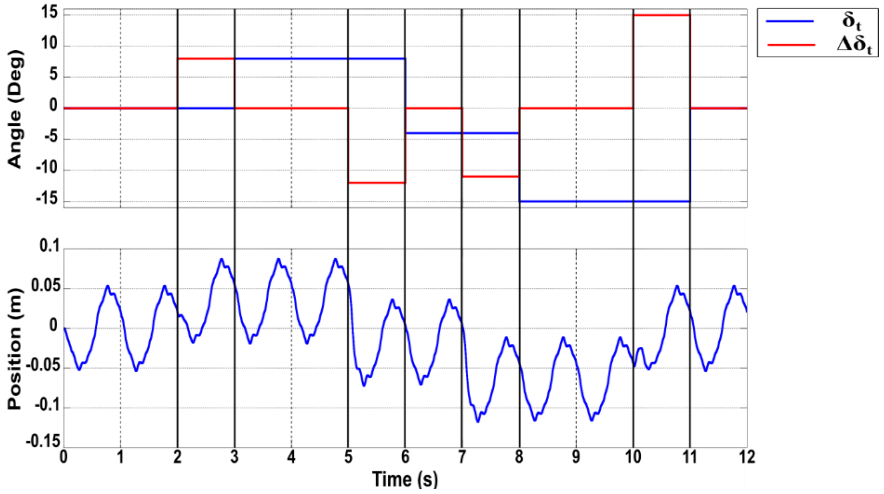


Figure 3.14- Relationship between system inputs (top) and corresponding system response (bottom).

As well as δ_T, Figure 3.14 demonstrates the second input parameter, $\Delta\delta_t$ which represents the change in δ_T from one tail beat period (one second) to the next. Figure 3.14 demonstrates that for each unique combination of these input parameters, there is a similarly unique output from the *Tail Actuator Dynamics & Kinematics* functionality in the form of the associated evolution of the caudal fin's position throughout that particular tail beat period.

Based on the results presented above, the structure of the black-box model used to replicate the functionality contained within the *Tail Actuator Dynamics & Kinematics* subsection will take the form of that shown below in Figure 3.15.

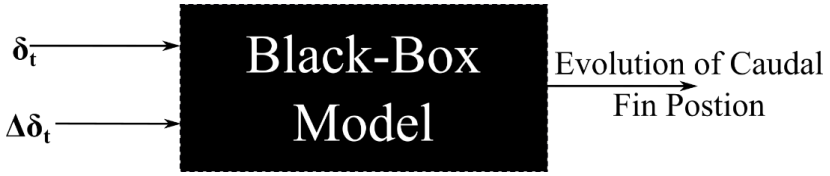


Figure 3.15- Black-box model structure to replace functionality within *Tail Actuator Dynamics* & *Kinematics* subsystems for caudal fin position.

Finally, as shown above in Figure 3.12 (b) and discussed previously, the position of the caudal fin as calculated using the proposed black-box model shown in Figure 3.15 is immediately used within the thrust calculation section of the model. Since this functionality is evaluated at 200 Hz, it is a requirement of the proposed black-box model to produce an updated value for the caudal fin's position every 0.005s. As a result, the output from the black-box model presented above will contain 200 elements with each element representing the position of the caudal fin at that particular point in the tail-beat period. This means that the evaluation of the resulting black-box model occurs once every second and a simple

indexing operation can be used to ensure the correct position is passed to the thrust estimation section of the model in the interim period.

As the bottom graph of Figure 3.13 illustrates, the input/output relationship for the orientation of the caudal fin is much simpler and is unaffected by changes in δ_T. As a result, a simpler, second black-box model can be used to represent this relationship as shown below in Figure 3.16.

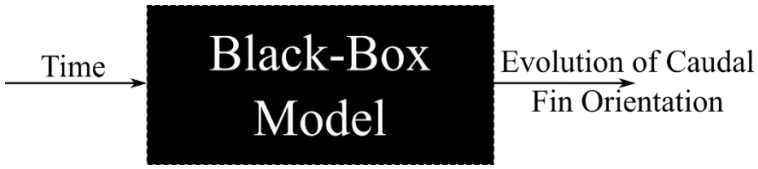


Figure 3.16- Black-box model Structure to replace functionality within *Tail Actuator Dynamics & Kinematics subsystems* for caudal fin orientation.

The black-box model proposed to replicate the evolution of the caudal fin's orientation contains just a single input, Time. This is due to the fact that as shown in Figure 3.13, time is the only parameter that the orientation of the caudal fin varies with and as a result, it has to be used as input to the black-box model. With the *tail-beat frequency* of the vehicle being equal to one second and the subsequent functionality within the model being evaluated at 200Hz, this input parameter will vary between zero and one in 0.005 increments. As a result, unlike the black-box model presented in Figure 3.15, this model is evaluated at 200Hz.

3.5.1 System Identification Process

As discussed previously, the process used to create black-box models is known as System Identification and occurs over three stages: *Data Acquisition, Model Selection & Evaluation* and *Model Validation* (Ljung 1999). The following sections will now discuss the steps taken within each of these stages in an attempt to create a black box model capable of replacing the functionality contained within the *Tail Actuator Dynamics & Kinematics* subsection of the model.

3.5.1.1 Data Acquisition

Black-box models are created exclusively from the input/output data obtained from the system being modelled. This data acquisition occurs during specially designed experiments where the user knowingly selects specific inputs that ensures the data obtained represents the system's response throughout its entire operational window. For the work completed in this thesis, the relevant data is obtained from the original validated model presented in Section 3.3. In order to obtain the relevant data, a number of open-loop simulations where

performed using the full nonlinear validated model. As discussed above, these simulations where designed specifically to ensure that the response of the system (i.e. the motion of the caudal fin) was obtained for δ_T (system input) values throughout the vehicle's entire operation range, i.e. -90º ≤ δ_T ≤ 90º in 5º intervals. As well as obtaining the steady-state responses of the system, the transient response of the system– both positive and negative - from each steady state value of δ_T were also obtained.

Once theses simulations where performed, the data obtained was arranged into a three dimensional matrix (row, column, page) of size 200 x 37 x 37. Within this matrix, the column and page dimensions are used to represent the unique combination of system inputs (δ_T & $\Delta\delta_t$) shown in Figure 3.15 at the aforementioned 5º intervals while the row entries represent the corresponding position of the caudal fin every 0.005s.

3.5.1.2 Model Selection

Once the relevant data has been acquired from the system, the next phase of the process is the evaluation of the mathematical model or the mathematical basis functions. However, before this can occur, a particular mathematical model structure has to be selected. As described in (Ljung, 1987), this is the most important choice to be made in the System Identification procedure and although numerous different model structures exist, a generalised form can be described as the weighted sum of a given basis function;

$$y_M(k) = \sum_{j=1}^{M} \alpha_j G_j(\mathbf{x}(k)) \tag{3.21}$$

Where y_M is the model output, α_j represents the individual weights and $G_j(\mathbf{x}(k))$ represents the evaluation of the chosen basis function for the input parameters represented by the vector, $\mathbf{x}(k)$.

Analysing Equation (3.21), it is apparent that the main decision to be made is what particular basis function to use. As described in (Horváth & Sergey 2003), there are many different functions to choose from including simple polynomials functions, Fourier Series approximations as well as Taylor's Series expansions. However, the work completed in this thesis uses a similar but slightly more complicated version of the generalised form presented above known as an artificial neural network (ANN). An ANN can be classified as an interconnected graph of simple mathematical functions whose operation can loosely be described as mimicking the function of the brain (Gurney, 1997). Like the brain, ANNs have the ability to 'learn' from the data they process and under certain user-defined conditions are capable of evaluating any given function (Cybenko 1989; Norgaard 1997;

Sjöberg et al. 1995). It is for this reason that ANNs have been implemented in this work to estimate the unknown mapping between the systems inputs (δ_T and $\Delta\delta_T$) and the corresponding position of the caudal fin.

3.5.1.3 Artificial Neural Networks (ANNs)

Shown below in Figure 3.17 is the standard structure of an ANN with an *Input* layer, a *Hidden* layer and an *Output* layer.

Unsurprisingly, the *Input* layer of a neural network contains its inputs which as discussed previously, for the work completed in this thesis, are the two parameters, δ_T and $\Delta\delta_T$. The *Hidden* layer contains a number of user-defined artificial neurons. Finally, similarly to the *Input* layer, the size of the *Output* layer is dictated by the number of outputs required from the network. As discussed previously, due to the 200Hz update rate of the *Vehicle Dynamics* section of the model (Figure 3.12 (b)), this value needs to be equal to 200. While the size of the *Input* and *Output* layers are dictated by the networks inputs and outputs, the number of artificial neurons contained within the *Hidden Layer* is user-defined. Based on the work completed in (Blum 1992) (Berry & Linoff 1997) (Heaton 2009) (Masters 1993), it was decided that in order to ensure the network would be able to successfully produce an accurate representation of the caudal fin's motion, 25 neurons would be included within this *Hidden* layer.

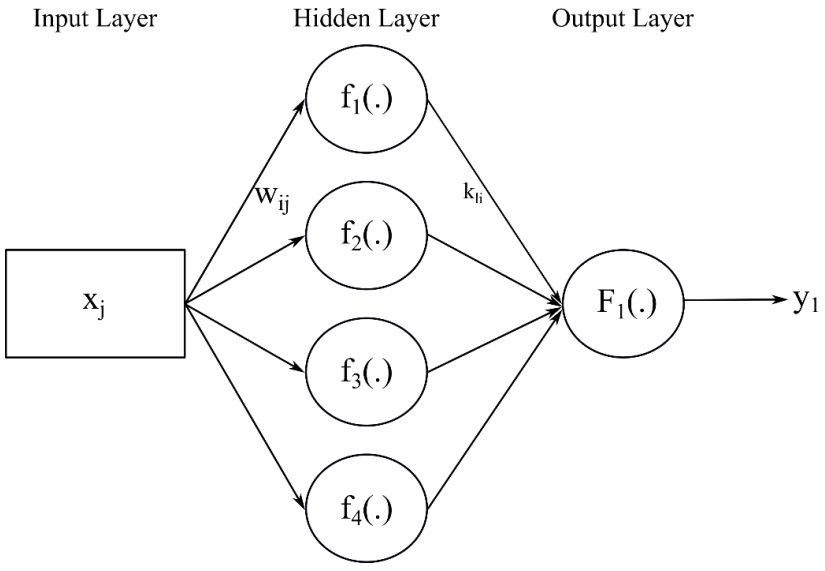


Figure 3.17- Structure of standard neural network.

Once the various layers discussed above have been defined, the output from the ANN used in this work can be expressed mathematically using Equation (3.22) shown below.

$$y_l(w,k,x) = F_l\left(\sum_{i=1}^{q} k_{li} f_i\left(\sum_{j=1}^{m} w_{ij}x_j\right)\right) \tag{3.22}$$

Here, y_l represents the l^{th} output from the network, w_{ij} represents the strength of the interconnecting weights between the input x_j and the i^{th} neuron in the hidden layer, m represents the number of inputs to the network, f_i is the activation function contained within the i^{th} neuron of the hidden layer, k_{li} represents the interconnecting weight between the i^{th} neuron in the hidden layer and l^{th} output neuron, q is the number of neurons in the hidden layer and finally, F_l is the activation function for the l^{th} output neuron.

While the size of the various layers of the ANN have been defined, it is apparent that the particular activation functions represented by F_l and f_i in Equation (3.21) have yet to be chosen. As shown below in Equation (3.23), it was decided that an approximation of the *hyperbolic tangent* function would be incorporated within the neurons of the *Hidden* and *Output* layers.

$$y_M = \frac{2}{\left(1 + e^{-2\left(\sum_{j=1}^{j=3} \alpha_j x_j\right)}\right)} - 1 \tag{3.23}$$

Again, x_j represents the system inputs, α_j represents the numerical strength (i.e. the weight) of the individual connections between the various layers of the network and finally, y_M represents the output of the neuron.

The reason for chosen the hyperbolic tangent function is to ensure that when averaged, the outputs obtained from the neurons in each layer are equal to approximately zero. This, as highlighted in (Le Cun et al. 1998) ensures the results obtained from the neurons are not biased in a particular numerical direction and as a result, the training process of the neural network is quickened.

This training process operates by implementing an algorithm that continuously alters the values for the various interconnecting weights (w and k in Equation (3.22)) until a suitable validation criteria has been satisfied and thus, the outputs from the ANN are deemed to produce an accurate representation of the system being mapped. While there are many different types of training algorithm available, the majority of projects use a variation of the popular optimisation technique known as gradient descent (Qian 1999). Included in these, is the backpropagation algorithm used in this work. A full derivation of the backpropagation method is presented in Appendix B.

Unsurprisingly, in order to allow this training process to be completed, not only does a suitable training algorithm need to be selected but also data must be supplied to the network that represents the inputs and associated outputs of the system for which the network is attempting to map. This data is referred to as the *training data* of the network and for the work undertaken in this research, is equal to the matrix of data obtained and described in Section 3.5.1.1.

In order to ensure the training process produces an ANN that is an accurate representation of the system being mapped, a validation methodology has to be incorporated within the training algorithm. To ensure that the ANN is able to accurately model the entire operational range of the vehicle and not just the values contained within the aforementioned *training data*, a process known as cross validation is undertaken.

This process involves using a *validation* data set to measure the ability of the network to predict the systems response to unseen data. As a result, it provides a superior metric by which to determine the ability of the network to model the systems entire operational range and guarantees that the network created doesn't contain a lack of *generalisation* (Krogh 1995).

While the particular validation methodology used within this work has been defined, a suitable validation metric still has to be set. The particular technique implemented in this work is known as the Theil's Inequality Coefficient (TIC) (Kat & Els 2012) and is evaluated using the equation shown below:

$$\varepsilon = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y - y_m)^2}}{\sqrt{\frac{1}{n}\sum_{i=1}^{200} y^2} + \sqrt{\frac{1}{n}\sum_{i=1}^{200} y_m{}^2}} \tag{3.24}$$

Where, ε represents the validation error, y represents the desired response from the network, y_m represents the networks actual response and n represents the number data points to be compare. In this case, the value of n is equal to the number of network outputs, 200.

Using the Theil's Inequality Coefficient as the validation metric is particularly useful as it normalises the difference between the two sets of data. This means that the value of ε will always be between zero and one. A value close to zero signifies that the two sets of data are similar while a value close to one signifies that the two sets of data are significantly different.

Using this metric, the training algorithm discussed above was evaluated and the weight values, i.e. w and k in Equation (3.22) associated with the minimum value for ε throughout this training process calculated. Subsequently, for the training process undertaken in this work, shown below in Figure 3.18, is the evolution of the validation error, ε as the training algorithm progresses through a number of *epochs*. A single *epoch* is completed once the training algorithm has completed a single training routine using the *training data.*

The results presented below demonstrate the expected behaviour with a relatively large initial error before the validation error begins to decrease as the training routine evolves. This reduction continues until approximately the 2500th epoch after which, the validation error begins to slowly increase and continuous to do so until the end of the training routine. Based on these results, the weights which will be used in the validated network are those calculated immediately before the validation error begins to increase, i.e. the 2455th epoch.

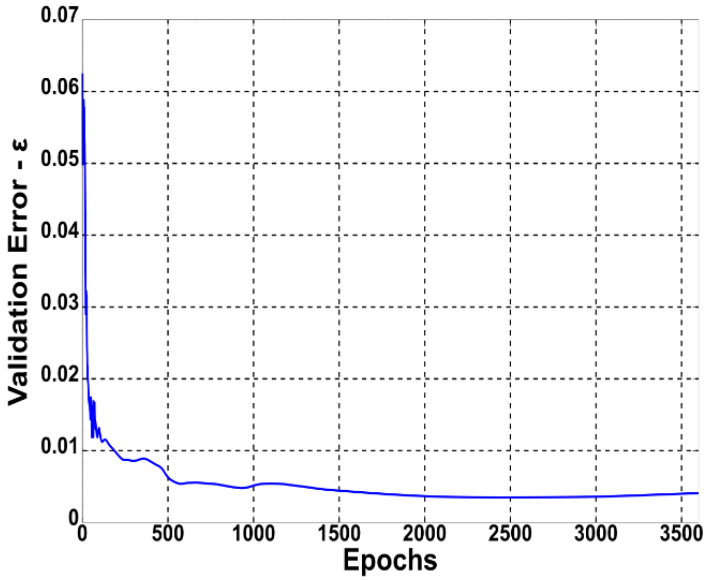


Figure 3.18- Evolution of validation error.

In summarising, the work presented above has described the methods used to create an ANN capable of modelling the motion of the RoboSalmon's caudal fin throughout its entire operational window. Furthermore, the cross-validation process undertaken and presented above in Figure 3.18 suggests that the ANN created is capable of producing an accurate representation of the caudal fin's motion. The next step in the validation process is to insert this black-box model in place of the original functionality contained within the *Tail Actuator Dynamics & Kinematics* subsection and validate the entire models response against the original validated mathematical model.

However, before this process occurs, another much simpler method of replacing the functionality within the *Tail Actuator Dynamics & Kinematics* subsection is presented. This method has been used widely and involves replacing the functionality described in Section 3.3.4 with a number of look up tables.

3.5.2 Look Up Tables

While look-up tables have been in existence for centuries, their purpose – to replace the evaluation of complex mathematical functions with tables containing the corresponding function values - has remained largely unchanged.

In order for a look-up table to be constructed, the combination of variables that produces a unique solution to the functionality being replaced must be identified. For the work completed in this thesis these parameters are δ_T, and $\Delta\delta_T$. In order to identify the correct point in the tail beat cycle, the parameter, t also has to be incorporated. By incorporating these three parameters, the LUT can be considered a three dimensional structure where every possible combination of the above three variables produces a unique solution (and position in space) as shown below in Figure 3.19.

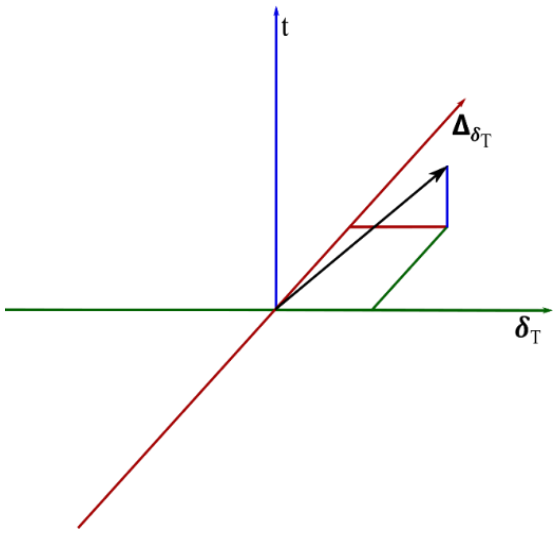


Figure 3.19 - Diagrammatic representation of 3-Dimensional look up table.

It's apparent from Figure 3.19 that the arrow is pointing towards a particular point in space and as a result, the value obtained will represent the position of the caudal fin for that unique combination of δ_T, $\Delta\delta_T$ and t. Therefore, during a single tail beat cycle, the values on the δ_T and $\Delta\delta_T$ axis will remain unchanged while the value for t will vary. Again, because of the requirement to provide data to the *Vehicle Dynamics* subsystem at a rate of 200Hz, the vertical axis in Figure 3.19 (and the equivalent field in the look up table) will include 200 values.

In order to populate the look up table with the relevant values, the high-fidelity model described in Sections 3.3 was evaluated for all possible integer combinations of δ_T and $\Delta\delta_T$ and the resulting motion of the caudal fin indexed within a look-up table. Once the table had been fully constructed, a simple indexing operation is used to retrieve the correct data based on the values of δ_T, $\Delta\delta_T$ and, t.

While the construction of the ANN described in Section 3.5.2.3 required a validation process, the incorporation of the look up table model doesn't. This is due to the fact that the values contained within these tables have been taking directly from the original validated mathematical model. Therefore, as long as the indexing operation has been completed correctly, the look up tables, by default, will provide an accurate representation of the relevant functionality.

3.6 Model Validation

The aim of this section is to demonstrate whether or not the black-box models presented in the previous two sections once incorporated into the full mathematical model, allow the dynamics of the RoboSalmon vehicle to be accurately modelled. To achieve this, a number of different scenarios have been simulated using these reduced fidelity models and the results compared with those from the original, high-fidelity model. As discussed previously, the particular parameters compared will be those presented in Table 3.1, i.e. the surge, sway & yaw velocities of the vehicle as well as the position of the vehicle in *Earth-fixed* axis and finally, the vehicle's heading angle. As with the validation process completed in Section 3.5, the scenarios completed have to ensure the entire operational range of the vehicle is tested, i.e. $-90^o \leq \delta_T \leq 90^o$. To achieve this a combination of open and closed loop manoeuvres are used.

3.6.1 Open Loop Manoeuvres

For the open loop manoeuvres, the well-known zig-zag and turning circle manoeuvres are used. These manoeuvres help evaluate particular characteristics of marine vehicles based on known rudder inputs. The turning circle manoeuvre, for example, is used to determine the manoeuvrability characteristics of a vehicle (Faltinsen 2006) while the zig-zag manoeuvre evaluates the response time of the vehicle in yaw (Fossen,2011). These particular manoeuvres were used as they can test the validity of the models throughout their entire operational range, i.e. $-90^o \leq \delta_T \leq 90^o$. This is shown below in Table 3.3 where the values used for δ_T across the two manoeuvres are presented.

Table 3.3- δ_T values used during various open loop simulations.

	ZigZag Manoeuvre	Turning Circle Manoeuvre
δ_T	±(10,20,30,40)°	±(50, 60, 70, 80, 90)°

The zigzag manoeuvre operates by setting the deflection angle of the control input, δ_T in this instance, to a specific value until the vehicle manoeuvres to a particular heading angle, ψ. At this point, the input is changed to -δ_T and remains there until the vehicle's heading angle manoeuvres to, $-\psi$. Then, the control input is changed back once again to +δ_T and the above process is repeated. Using this manoeuvre, the tail centre deflection angle will switch between ±δ_T. As a result, for the four manoeuvres presented in Table 3.3, the values of δ_T tested will be -40º≤δ_T≤40º in 10º intervals.

For the turning circle manoeuvre, the value of δ_T is kept at a constant value until the vehicle's heading angle rotates through a full 360º. As a result, to test each of the δ_T values presented in Table 3.3, ten simulations have to be completed.

In order to complete the validation process, an identical simulation has been undertaken for each of the manoeuvres presented in Tables 3.3 using the original high-fidelity model, the reduced fidelity model containing the look up table (RFMLUT) and the reduced fidelity model containing the ANN (RFMANN). As shown below in Table 3.4, the results obtained from the zig-zag manoeuvres from these reduced fidelity models have been compared with those obtained from the high-fidelity model using the Theil's Inequality Coefficient (TIC) presented previously in Equation (3.24).

Table 3.4- TIC values for zig zag manoeuvres with Look up Table Model.

δ_T	Surge	Sway	Yaw	X-Pos	Y-Pos	Heading	Avg
±10	$1.27\text{x}10^{-6}$	$1.49\text{x}10^{-6}$	$6.4\text{ x}10^{-6}$	$4.82\text{x}10^{-7}$	$1.36\text{x}10^{-5}$	$5.64\text{x}10^{-6}$	$4.82\text{x}10^{-6}$
±20	$1.57\text{x}10^{-6}$	$1.84\text{x}10^{-6}$	$5.5\text{ x}10^{-6}$	$7.71\text{x}10^{-7}$	$1.04\text{x}10^{-5}$	$5.62\text{x}10^{-6}$	$4.28\text{x}10^{-6}$
±30	$1.70\text{x}10^{-6}$	$1.90\text{x}10^{-6}$	$4.97\text{x}10^{-6}$	$8.80\text{x}10^{-7}$	$7.90\text{x}10^{-6}$	$5.02\text{x}10^{-6}$	$3.73\text{x}10^{-6}$
±40	$1.81\text{x}10^{-6}$	$1.90\text{x}10^{-6}$	$4.83\text{x}10^{-6}$	$1.04\text{x}10^{-6}$	$8.18\text{x}10^{-6}$	$5.14\text{x}10^{-6}$	$3.82\text{x}10^{-6}$

Table 3.5 - TIC values for zig zag manoeuvres with Neural Network Model.

δ_T	Surge	Sway	Yaw	X-Pos	Y-Pos	Heading	Avg
±10	$3.1\text{x}10^{-3}$	$8.9\text{x}10^{-3}$	$7.5\text{x}10^{-3}$	$5.29\text{x}10^{-4}$	$4.5\text{x}10^{-3}$	$2.50\text{x}10^{-3}$	$4.4\text{x}10^{-3}$
±20	$4.5\text{x}10^{-3}$	$1.2\text{x}10^{-2}$	$9.2\text{x}10^{-3}$	$8.70\text{x}10^{-4}$	$4.6\text{x}10^{-3}$	$2.70\text{x}10^{-3}$	$5.6\text{x}10^{-3}$
±30	$4.9\text{x}10^{-3}$	$1.1\text{x}10^{-2}$	$1.0\text{x}10^{-2}$	$8.45\text{x}10^{-4}$	$1.5\text{x}10^{-3}$	$2.50\text{x}10^{-3}$	$5.2\text{x}10^{-3}$
±40	$5.4\text{x}10^{-3}$	$1.1\text{x}10^{-2}$	$1.2\text{x}10^{-2}$	$8.72\text{x}10^{-4}$	$5.9\text{x}10^{-3}$	$5.00\text{x}10^{-3}$	$6.6\text{x}10^{-3}$

Overall, with the average coefficient value being of the order 10^{-6} and 10^{-3} for the RFMLUT and RFMANN models respectively, the results suggest that both models are capable of accurately replicating the dynamics of the RoboSalmon vehicle when $-40^{o} \leq \delta_T \leq 40^{o}$. While both small, the results presented above demonstrate a significant difference in the coefficient values obtained from the RFMLUT and RFMANN models with RFMLUT model, on average, producing a TIC value that is three orders of magnitude less than the RFMANN model. While this may seem a significant difference, the fact that the TIC values for both models are as small as they are suggests that any difference in the results obtained form the two models is likely to be negligible. Regardless, its apparent from these results that when the tail centre deflection angle is in the range $-40^{o} \leq \delta_T \leq 40^{o}$, the reduced fidelity models are capable of providing an accurate representation of the vehicle's dynamics.

While the results presented above demonstrate the validity of the reduced fidelity models when δ_T is between $\pm 40^{o}$, it will only be once the results produced from the turning circle manoeuvres are obtained, that it will be possible to determine the overall validity of the model. Subsequently, shown below in Tables 3.6 & 3.7 are the TIC values obtained from these manoeuvres. Specifically, Table 3.6 represents the coefficients obtained when the outputs from original, high-fidelity model are compared with those from the RFMLUT model while Table 3.7 represents the same results but for the comparison involving the RFMANN model.

Table 3.6 - TIC values for turning circle manoeuvre with Look up Table Model.

δ_T	**Surge**	**Sway**	**Yaw**	**X-Pos**	**Y-Pos**	**Heading**	**Avg**
50	1.36×10^{-6}	8.01×10^{-7}	2.48×10^{-6}	1.62×10^{-6}	9.26×10^{-7}	4.02×10^{-7}	1.26×10^{-6}
60	1.58×10^{-6}	7.74×10^{-7}	2.43×10^{-6}	1.96×10^{-6}	1.02×10^{-6}	4.26×10^{-7}	1.37×10^{-6}
70	2.18×10^{-6}	8.72×10^{-7}	2.76×10^{-6}	1.74×10^{-6}	1.04×10^{-6}	5.84×10^{-7}	1.53×10^{-6}
80	2.76×10^{-6}	8.52×10^{-7}	2.71×10^{-6}	2.76×10^{-6}	1.08×10^{-6}	5.89×10^{-7}	1.79×10^{-6}
90	4.11×10^{-6}	8.29×10^{-7}	2.65×10^{-6}	4.92×10^{-6}	1.18×10^{-6}	5.80×10^{-7}	2.38×10^{-6}
-50	1.6×10^{-6}	2.06×10^{-6}	1.39×10^{-6}	1.32×10^{-6}	2.16×10^{-7}	9.22×10^{-8}	1.11×10^{-6}
-60	1.82×10^{-6}	1.99×10^{-6}	1.40×10^{-6}	1.53×10^{-6}	2.65×10^{-7}	1.02×10^{-7}	1.19×10^{-6}
-70	2.13×10^{-6}	1.95×10^{-6}	1.42×10^{-6}	1.75×10^{-6}	3.2×10^{-7}	1.08×10^{-7}	1.28×10^{-6}
-80	2.59×10^{-6}	1.91×10^{-6}	1.43×10^{-6}	1.88×10^{-6}	3.67×10^{-7}	1.07×10^{-7}	1.38×10^{-6}
-90	3.44×10^{-6}	1.86×10^{-6}	1.42×10^{-6}	1.78×10^{-6}	4.0×10^{-7}	9.84×10^{-8}	1.5×10^{-6}

Analysing the results presented above and below, it is noticeable that they are very similar to those presented in Tables 3.5 & 3.6. This similarity is apparent when the orders of magnitude of the coefficients are taken into consideration where the results demonstrate that on average, they are of the order 10^{-6} and 10^{-3} for the RFMLUT and RFMANN models

respectively. These values are identical to those presented in the previous section and once again demonstrate the similarity in the results obtained from the reduced fidelity models and the original, validated model. These results, when combined with those presented in Tables 3.5 & 3.6, suggest that the reduced fidelity models are capable of producing an accurate representation of the RoboSalmon vehicle throughout its entire operation range, i.e. $-90^{o} \leq \delta_T \leq 90^{o}$.

Table 3.7- TIC values for turning circle manoeuvre with ANN Model.

δ_T	Surge	Sway	Yaw	X-Pos	Y-Pos	Heading	Avg
50	$2.3x10^{-3}$	$2.4x10^{-3}$	$3.7x10^{-3}$	$3.1x10^{-3}$	$1.6x10^{-3}$	$6.7x10^{-4}$	$2.3x10^{-3}$
60	$2.8x10^{-3}$	$2.5x10^{-3}$	$4.0x10^{-3}$	$4.1x10^{-3}$	$1.9x10^{-3}$	$7.5x10^{-4}$	$2.7x10^{-3}$
70	$4.0x10^{-3}$	$2.8x10^{-3}$	$4.5x10^{-3}$	$4.1x10^{-3}$	$2.3x10^{-3}$	$1.2x10^{-3}$	$3.2x10^{-3}$
80	$5.1x10^{-3}$	$2.8x10^{-3}$	$4.4x10^{-3}$	$7.0x10^{-3}$	$3.0x10^{-3}$	$1.3x10^{-3}$	$3.9x10^{-3}$
90	$7.6x10^{-3}$	$2.8x10^{-3}$	$5.2x10^{-3}$	$1.3x10^{-2}$	$4.9x10^{-3}$	$1.0x10^{-3}$	$5.7x10^{-3}$
-50	$1.6x10^{-3}$	$2.3x10^{-3}$	$2.5x10^{-3}$	$2.4x10^{-3}$	$3.3x10^{-4}$	$1.1x10^{-4}$	$1.5x10^{-3}$
-60	$1.8x10^{-3}$	$2.2x10^{-3}$	$2.4x10^{-3}$	$2.7x10^{-3}$	$2.43x10^{-4}$	$4.3x10^{-5}$	$1.6x10^{-3}$
-70	$2.1x10^{-3}$	$2.1x10^{-3}$	$2.5x10^{-3}$	$3.1x10^{-3}$	$3.29x10^{-4}$	$6.0x10^{-5}$	$1.7x10^{-3}$
-80	$2.5x10^{-3}$	$2.1x10^{-3}$	$2.4x10^{-3}$	$3.1x10^{-3}$	$5.3x10^{-4}$	$1.5x10^{-4}$	$1.8x10^{-3}$
-90	$3.4x10^{-3}$	$2.1x10^{-3}$	$2.8x10^{-3}$	$2.8x10^{-3}$	$5.0x10^{-4}$	$3.6x10^{-5}$	$1.9x10^{-3}$

However, it is crucial to note that as illustrated in Table 3.3, the values used for δ_T throughout these validation tests increased in 10^{o} intervals and as a result, the validity of the reduced fidelity models cannot be guaranteed by the results shown above in Table 3.4-7. In order to demonstrate further the validity of the reduced fidelity models, a closed loop manoeuvre needs to be undertaken. Incorporating a closed loop manoeuvre will not only further assist in indicating the validity of the reduced fidelity models but also the ability of the RoboSalmon vehicle to autonomoulsy navigate to a number of predefined waypoints.

3.6.2 Closed Loop Manoeuvre

As opposed to an open loop manoeuvre where the values used for, δ_T will be predefined, a closed loop manoeuvre involves the vehicle's *Guidance System* calculating the values for δ_T based on the vehicles current position and its desired location. Incorporating such a closed loop manoeuvre within the validation process ensures the reduced fidelity models are validated with respect to a realistic scenario likely to be undertaken by the RoboSalmon vehicle. However, before this final validation process is undertaken, the following section will present the guidance control law used in the RoboSalmon vehicle to calculate the appropriate value for δ_T that ensures the vehicle will manoeuvre in the desired direction.

3.6.2.1 RoboSalmon's Guidance Heuristic

Shown below in Figure 3.20 is a representation of the *Guidance System* incorporated within the RoboSalmon vehicle that allows it to autonomously navigate to any desired location.

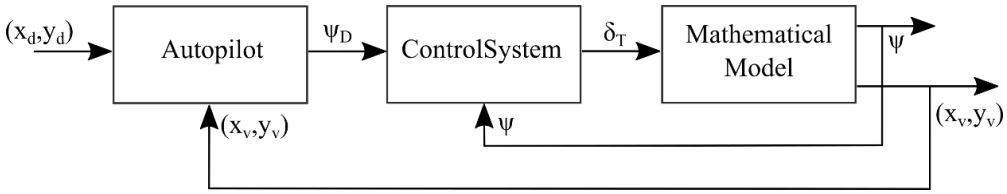


Figure 3.20- Representation of RoboSalmon's guidance system structure.

Here x_d and y_d represent the coordinates of the waypoint the vehicle is required to manoeuvre towards, ψ_D is the associated desired heading angle of the vehicle which ensures the vehicle continues to manoeuvre towards that required waypoint, δ_T is the required control surface deflection angle and x_v, y_v and ψ represent the current position and orientation of the vehicle in the *Earth*-fixed reference frame.

As the diagram demonstrates, for the work completed in this thesis, these parameters (x_v, y_v and ψ) are obtained from the vehicles mathematical model. In reality though, they would be calculated using the vehicles navigation system. Regardless of how they are calculated, Figure 3.20 demonstrates that the *Autopilot* subsequently uses them in conjunction with the vehicle's desired location, (x_d, y_d) to calculate the desired heading angle, ψ_D that will ensure the vehicle manoeuvres in the correct direction. This process is shown diagrammatically below in Figure 3.21.

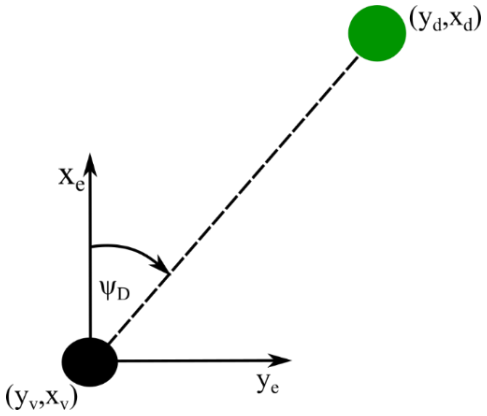


Figure 3.21- Geometry of waypoint guidance methodology.

As discussed in (Mazlan 2015), to calculate this desired heading angle, ψ_D Equation (3.25) is incorporated within *Autopilot* section of the vehicles *Guidance Systems.*

$$tan(\psi_D) = \left(\frac{x_d - x_v}{y_d - y_v}\right) \tag{3.25}$$

Equation (3.25) represents a standard Line-of-Sight calculation (Healey & Lienard 1993; Healey, Marco 1992). Normally, the value for ψ_D would be calculated by simply evaluating the inverse tangent of Equation (3.25). In this work though, the built-in Matlab function, atan2 is used instead (Mathworks 2006a). The reason for adopting this particular function is to ensure that the particular quadrant occupied by the desired waypoint can be correctly identified. As a result, the evaluated desired heading angle lies in the region, $-\pi$ to $+\pi$ as shown below in Figure 3.22.

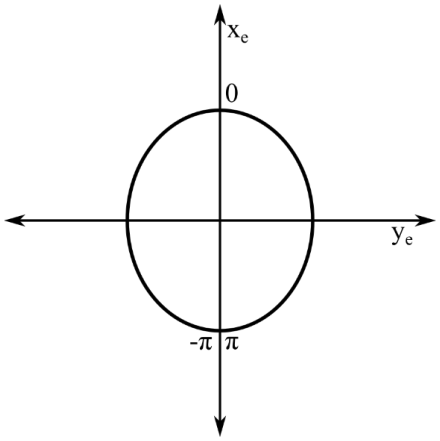


Figure 3.22- Diagrammatic representation of the four-quadrant inverse tangent function.

While using the atan2 function allows the correct quadrant to be identified, problems arise when two consecutives values for ψ_D are evaluated at opposite sides of the discontinuity, i.e. 176.2º and then -179.6º.

If left unaltered, this discontinuity will result in the *Guidance System* instructing the vehicle to turn 355.8º counter-clockwise, when intuitively, the vehicle should only have to turn 4.2º clockwise. To overcome this problem, a mapping $\psi_D: \langle -\pi, \pi \rangle \rightarrow \langle -\infty, \infty \rangle$ has to be constructed (Adkins & Yan 2006; Rafferty 2014). This mapping is achieved using the following equations:

$$\psi_D = \psi_{DPrevious} + sgn(\psi_{DPrevious})(2\pi - |\Delta\psi_D|) \tag{3.26}$$

Where

$$\Delta\psi_D = \psi_{DCurrent} - \psi_{DPrevious} \tag{3.27}$$

Applying the values of 176.2º and -179.6º to the above equations results in a desired heading angle of 180.4º. As desired, this value will result in the vehicle manoeuvring through the much smaller angle of 4.2º. In order to ensure every vehicle always manoeuvres through the smallest possible angle, Equations (3.26) & (3.27) are evaluated each time the *Autopilot* evaluates a new desired heading angle.

Importantly, as discussed in Chapter 2, the desired waypoints used in Equation (3.25) are predefined and will vary depending on the mission requirements. Furthermore, a proximity calculation is used to determine when the vehicle alters the values used for these waypoints. This proximity calculation evaluates the distance between the vehicle and the chosen waypoint and when its value is below a certain threshold, the *Autopilot* will automatically select the next waypoint in the predefined list. When this occurs, Equations (3.26)-(3.28) will evaluate an appropriate desired heading angle, ψ_D that will ensure the vehicle manoeuvres towards this newly selected waypoint. This process will then continue until the vehicle has manoeuvred within range of the final waypoint.

3.6.2.2 Control System

In the previous section, the functionality used in the vehicle's *Autopilot* to evaluate its desired heading angle, ψ_D was presented. The purpose of this section is to now describe the functionality contained within the *Control System.* As shown in Figure 3.20, this subsystem takes into consideration the aforementioned desired heading angle of the vehicle, ψ_D its current heading angle, ψ and produces the required tail deflection angle, δ_T that will ensure the vehicle manoeuvres in the correct direction.

To achieve this, a standard PI controller identical to that shown below in Equation (3.28) is used.

$$\delta_T = K_p \Delta\psi + K_i \int \Delta\psi \, dt \tag{3.28}$$

Here, K_p and K_i are the proportional and integral gains equal to 3.4x10-2 and 2x10-5 respectively and $\Delta\psi$ is difference between the desired heading angle and the current heading angle, ψ of the vehicle. It is important to note that the PI controller and associated gains presented in Equation (3.28) are identical to the controller used within the RoboSalmon vehicle.

3.6.2.3 Description of Closed Loop Manoeuvre

In the previous two sections, the functionality contained within the *Autopilot & Control* subsystems of the RoboSalmon vehicle that allow it to autonomously navigate to a number

of predefined waypoints was presented. The purpose of this section is to now describe the particular closed loop manoeuvre used as the final validation test for the RFMLUT and RFMANN models. As Figure 3.23 illustrates, the closed loop manoeuvre undertaken will involve the vehicle manoeuvring to four different waypoints sequentially.

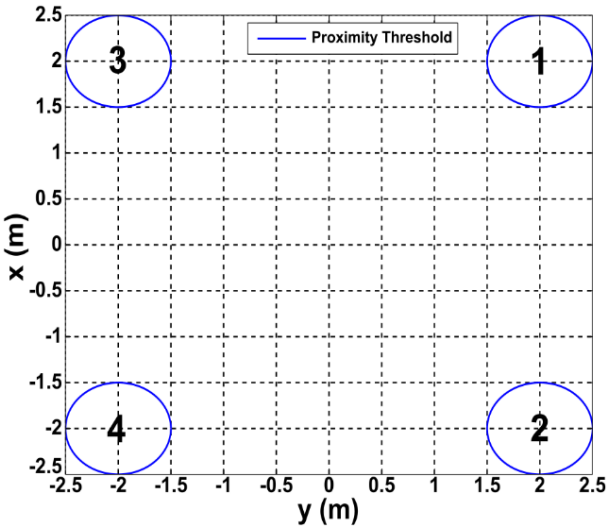


Figure 3.23 - Diagrammatic representation of waypoints used in closed loop validation test.

Figure 3.23 also illustrates the aforementioned proximity threshold which, as discussed previously, once the vehicle is within, will instruct the *Autopilot* to select the next waypoint in the predefined list. Furthermore, the particular sequence of waypoints shown in Figure 3.23 was chosen as not only will it further test the validity of the models but it will also demonstrate whether or not the mapping presented in Equations (3.26) and (3.27) operates as envisioned.

3.6.2.4 Results – Closed Loop Manoeuvre

Presented in this section is the comparison between the results obtained from the reduced fidelity models and the original validated model for the closed loop manoeuvre presented in Figure 3.23. To begin, as with the results presented for the open loop manoeuvres, shown below in Table 3.8 is the TIC values obtained when the results from the RFMLUT and RFMLUT models were compared with those from the original validated model.

Table 3.8 - TIC Values for reduced fidelity models for figure of eight manoeuvre.

Model	Surge	Sway	Yaw	X-Pos	Y-Pos	Heading	Avg
LUT	0.0035	0.0047	0.0063	0.0025	0.0011	1.53×10^{-4}	0.0030
ANN	0.0078	0.0118	0.0160	0.0092	0.0095	0.0013	0.0093

The results presented in Table 3.8 demonstrate that while the difference between the reduced fidelity models and the original model are still small, the values obtained are significantly greater than those obtained from the two open loop manoeuvres. The reason for this discrepancy is due to a number of factors. The first of which is due to the fact that during the open loop manoeuvres, the values used for δ_T where increased in 10^o intervals. Importantly, as shown in Tables 3.4, these intervals included data that had been used throughout the training process of the RFMANN model and as a result, the error produced should be small. Similarly, because the δ_T values tested throughout the open-loop manouvres where ineger values, the corressponding motion of the caudal fin as conatained within the look up tables would be identical to that produced from the original model.

Conversely, during the closed loop manouvres, δ_T is no longer guaranteed to be an integer value or increasing in the same 10^o intervals. As a result, at this point, the RFMANN is predicting the motion of the caudal fin having never processed data for this particular scenario before and as a result, the accuracy of the model is dependant on the generality of the network created. Similarily, with the RFMLUT model only containing the motion of the caudal fin for integer values of δ_T, there is bound to be a loss in accuracy when the RFMALUT has to interpolate between the integer values contained in the table.

Nevertheless, with an average coefficient value of 0.0093 and 0.0030 for the RFMANN and RFMLUT models respectively, the reduced fidelity models can still be classified as providing an accurate representation of the RoboSalmon's dynamics. This is demonstrated below in Figure 3.24 where the evolution of the various states as obtained from the three models are presented.

Visually, the results presented below appear almost identical with the differences between the three models difficult to ascertain. Figure 3.24(d) also demonstrates that the mapping presented in Equations (3.26) and (3.27) has operated as required and ensures the vehicle manoeuvres through the smallest possible angle when moving between the different waypoints.

Therefore, the results presented below in Figure 3.24 combined with those presented previously in Table 3.4-3.8 demonstrate the ability and validity of the two reduced fidelity models to accurately model the dynamics of the RoboSalmon vehicle.

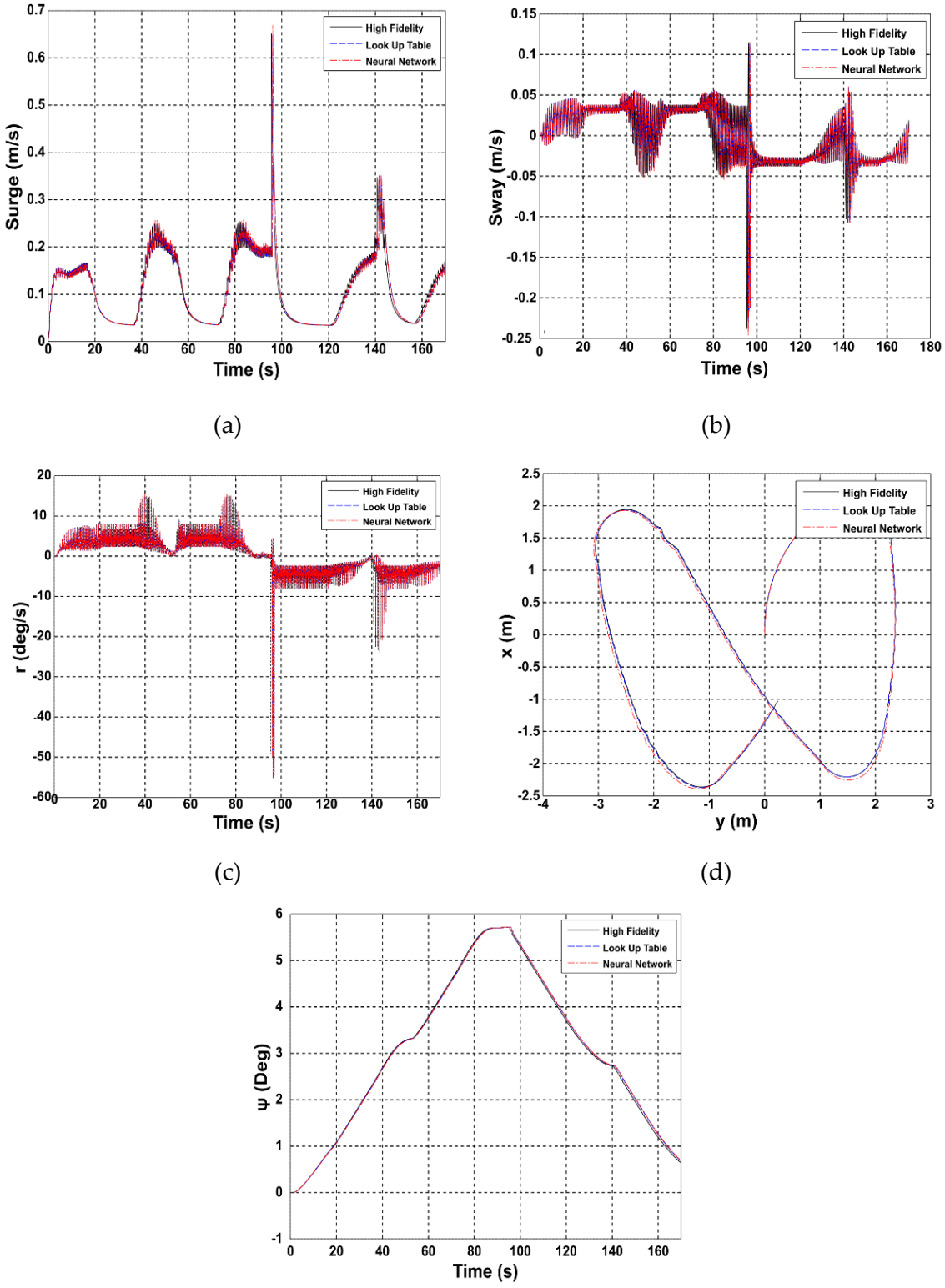


Figure 3.24- Comparison of results obtained from RFMLUT model, RFMNN model for (a) Surge Vehicle, (b) Sway Velocity, (c) Yaw Rate, (d) Trajectory & (e) Heading Angle for figure-of-eight manoeuvre.

However, as well as demonstrating the validity of these models, it is also necessary to demonstrate that as required, the two reduced fidelity models drastically reduce the

simulation execution time of the model. To demonstrate whether or not this is the case, shown below in Figure 3.25, is the relationship between the number of vehicles simulated and the corresponding simulation execution time of the models.

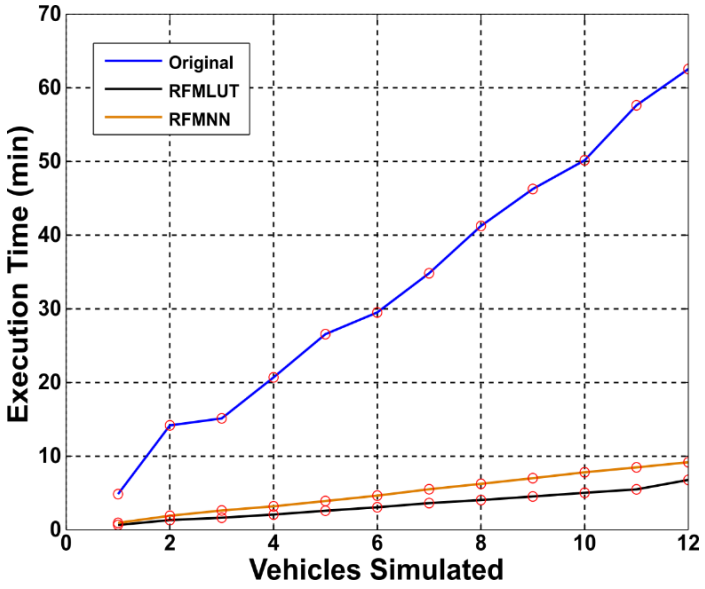


Figure 3.25- Relationship between the number of vehicles simulated and execution time for the various models.

Figure 3.25 clearly illustrates that as a result of incorporating the RFMANN and RFMLUT models, the simulation execution time of the original validated mathematical model has been reduced significantly. In the case of the RFMLUT model, the execution time has reduced by approximately 90% when twelve vehicles are modelled and for the RFMANN model, this value is slightly smaller at approximately 85%.

Therefore, combining these results with those of Tables 3.5-9 and Figure 3.28, it is apparent that the RFMANN and RFMLUT models not only maintain the accuracy of the original model but do so while drastically reducing the simulation execution time of the model.

In terms of selecting which one of these models is to be used for the remainder of the work completed in this thesis, the results presented above demonstrate that not only does RFMLUT model provide a more accurate representation of the vehicle's dynamics but also it does so in a more time efficient manner. As a result, it is for these reasons that the RFMLUT model will be used to simulate the dynamics of the RoboSalmon vehicle for the remainder of the work completed in this thesis.

However, it is important to note that the validation tests presented above have been obtained when the RoboSalmon's *Tail-Beat Amplitude* was set to 0.05m. The *Tail-Beat*

Amplitude represents the maximum lateral displacement of the vehicles caudal fin during a single undulation and can be used to alter the vehicles velocity. As the subsequent work completed in this thesis will involve the vehicle having to alter its velocity, it is necessary to also validate the model for different *Tail-Beat Amplitudes.* To achieve this, shown below in Table 3.9 are the TIC values obtained when the results obtained from the RFMLUT model were compared with those from the original model for the closed loop-manoeuvre presented in Figure 3.24 for two different *Tail-Beat Amplitude*s, i.e. 0.1m and 0.15m.

Table 3.9– TIC values for reduced fidelity models for figure of eight manoeuvre with different tail beat amplitudes.

Tail Beat Amplitude	Surge	Sway	Yaw	X-Pos	Y-Pos	Heading	Avg
0.1m	0.0071	0.064	0.040	0.0060	0.0180	0.0076	0.0024
0.15m	0.0020	0.035	0.046	0.0026	0.0112	0.0134	0.0184

As with the results presented in Table 3.8, the results presented above in Table 3.9 demonstrate that the RFMLUT model created is capable of accurately modelling the dynamics of the RoboSalmon vehicle when its *Tail-Beat Amplitude* is altered. Combining these results with those presented in Figure 3.25, it is apparent that the RFMLUT provides a valid model that can be used to design, develop and test the formation control algorithm in a time efficient manner.

3.7 Summary

To begin, this chapter, in Section 3.2 described the RoboSalmon vehicle and how the fully actuated *Tail Section* was designed to allow the vehicle to replicate the swimming gait of the North Atlantic salmon. Section 3.2 also described that due to the limited number of actuated joints as well as the physical limitations of the DC motors used, the *Tail Section* of the RoboSalmon vehicle is unable match the swimming performance of real North Atlantic salmon.

Thereafter, Section 3.3 provided a detailed description of the functionality contained within the original validated mathematical model of the RoboSalmon vehicle. Section 3.4 meanwhile illustrated that in its original guise, the mathematical model of RoboSalmon is ill-suited to be used to model multiple vehicles simultaneously due to its poor simulation execution time. The results presented in this section also demonstrated that this poor simulation execution time was caused primarily by the functionality contained within the *Tail Actuator Dynamics & Kinematics* section of model.

Section 3.5 described in detail the System Identification techniques implemented to replace this computationally expensive functionality with a combination of simpler mathematical basis functions and indexing operations.

In Section 3.6, a number of open and closed loop manoeuvres were completed to demonstrate that the two reduced fidelity models (RFMANN and RFMLUT) detailed in Section 3.5 were capable of accurately modelling the dynamics of the RoboSalmon vehicle. This was achieved by comparing the results obtained from the newly created reduced fidelity models with those obtained from the original validated model. The results demonstrate that although the two reduced fidelity models were capable of modelling the dynamics of the RoboSalmon vehicle accurately, the RFMLUT was more accurate.

Furthermore, Sections 3.6.2.1-2 described in detail the functionality contained within the *Autopilot* and *Control* subsystems that when combined, form the *Guidance System* of the RoboSalmon vehicle and as such, allow the vehicle to autonomously navigate to a number of predefined waypoints. Thereafter, Section 3.6.2.4 illustrated the ability of the reduced fidelity models to accurately model the dynamics of the RoboSalmon vehicle during a realistic closed loop manoeuvre. Section 3.6.2.4 also demonstrated the significant improvements in the simulation execution time obtained from the RFMANN and RFMLUT models when compared with the original validated model. As with the results comparing the accuracy of the two models, these results once again demonstrate that the RFMLUT model was slightly more time efficient than the RFMANN model. Based on these results, it was decided that for remainder of the work completed in this thesis, the RFMLUT would be used to model the RoboSalmon vehicle.

Chapter 4

Biomimetic Formation Control Algorithm

4.1 Introduction

As stated previously, the main aim of this thesis is to develop a formation control algorithm capable of successfully coordinating a group of BAUVs to complete the oceanic surveying missions outlined in Chapter 1 regardless of the nuances of the underwater environment. In the previous chapter, the mathematical model used in this work was presented and its validity proven. The focus of this chapter will now be the presentation of the formation control algorithm based on the behavioural mechanisms of fish partaking in schooling behavioural. Its implementation and the subsequent determination of its suitability to allow a group of BAUVs to complete the required oceanic surveying missions will be the main focus of this chapter.

To achieve this, the chapter is structured as follows. Section 4.2 provides a brief summary of the behavioural mechanisms used by fish when schooling and what requirements these place on the proposed algorithm. Section 4.3 describes the transformation of these behavioural mechanisms into a suitable formation control algorithm that can be implemented within the *Guidance System* of the RoboSalmon vehicle. Section 4.4 describes the simulation set up as well as the performance metrics and convergence criteria used to analyse the algorithms suitability to coordinate the vehicles as required. Section 4.5 analyses the results obtained from the various simulations completed and Section 4.6 presents a summary of the work completed in this chapter.

4.2 Behavioural Mechanisms of Schooling Behaviour

As discussed in detail in (Aoki, 1982; Huth & Wissel, 1991), in order for aquatic animals to take part in schooling behaviour they must exhibit three behaviours: *repulsion, orientation & attraction.* Furthermore, Aoki (Aoki, 1981) demonstrated that in order for these behaviours to produce a stable schooling mechanism, their implementation must be triggered by the proximity of each fish to its nearest neighbour(s). Therefore, as shown below in Figure 4.1, these behavioural mechanisms can be described by assigning a circular area of influence to each member of the group, within which, three concentric circles (behavioural zones) are contained. Depending on which of these three zones is occupied, each member of the school will manoeuvre in either a *repulsive*, *orientating* or *attractive* manner.

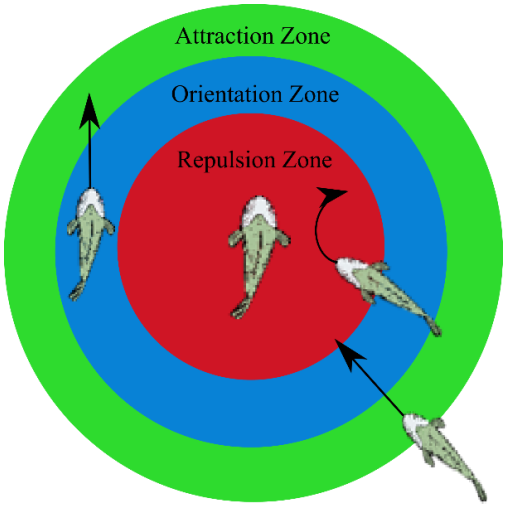


Figure 4.1 - Behavioural zones used to explain schooling behaviour.

As the diagram demonstrates, if a member of the school is exhibiting the *repulsive* behaviour, it will manoeuvre away from it nearest neighbour(s). Conversely, if an individual is using the *attractive* behaviour, it will move towards its nearest neighbour(s). Finally, individuals using the *orientating* behaviour will not only align their heading angle with that of their nearest neighbour(s) but also their velocity vector.

Inspecting Figure 4.1, it becomes apparent that in order for each member of the school to select the correct behaviour they must firstly be capable of determining the distance between themselves and their nearest neighbour(s). Thereafter, when using the *orientating* behaviour, the individuals must also be capable of determining the orientation and velocity of their nearest neighbour(s). In nature, these functions are completed through sensing organs such as vision and the lateral line (Partridge et al. 1980).

However, as discussed in Chapter 2, the only feasible way for each vehicle to have access to this data is through the transfer of data across the underwater communication channel. As well as ensuring every vehicle has access to the relevant data, Figure 4.1 also demonstrates that a number of additional tasks will need to be completed by the formation control algorithm before it is able to mimic the behavioural mechanisms of fish partaking in schooling behaviour. As shown below in Table 4.1, these tasks, when combined, form the algorithm's definition of requirements.

Table 4.1 – Definition of requirements for biomimetic formation control algorithm.

Requirement Topic	Description
1 - Communication	A suitable method to allow each vehicle to have access to up-to-date information on the position, orientation and velocity of neighbouring vehicles.
2 - Evaluation of Nearest Neighbour Distance	Implementation of a suitable equation that calculates the *absolute* distance between neighbouring vehicles.
3 – Select Correct Behaviour	Incorporation of a suitable decision making process that allows each vehicle to select the correct behaviour based on its proximity to its nearest neighbour.
4 - Implementation of Repulsive Behaviour	Implementation of a behavioural control law that ensures each vehicle manoeuvres in a repulsive manner and hence looks to *increase* the distance between itself and its nearest neighbour.
5 - Implementation of Orientating Behaviour	Implementation of a suitable control law that ensures each vehicle seeks to *align* its heading angle with that of its nearest neighbour(s).
6 - Implementation of Attractive Behaviour	Implementation of a suitable control law that ensures each vehicle seeks to manoeuvre towards its nearest neighbours.
7 - Implementation of Velocity Control	Implementation of a suitable velocity control law that depending on the behavioural control law being used, results in the vehicle either reducing, increasing or maintaining its velocity.

If the algorithm satisfies each of the requirements presented above, then the formation control algorithm will allow the BAUVs to mimic the behavioural mechanisms of fish when schooling. The following section will now describe in detail the functionality implemented within the *Autopilot* subsystem of the RoboSalmon vehicle in order to ensure that these requirements can be satisfied.

4.3 Biomimetic Formation Control Algorithm: Guidance Heuristic

Shown below in Figure 4.2 is the structure of the *Guidance System* used within the RoboSalmon vehicle.

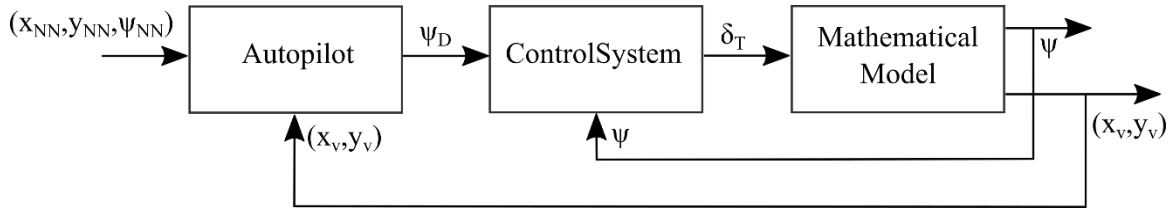


Figure 4.2 - Guidance system architecture.

The only difference between the diagram presented above in Figure 4.2 and that shown previously in Chapter 3 is that now, the inputs to the *Autopilot* subsystem are no longer the coordinates of the vehicles desired waypoints, i.e. x_d and y_d but instead, the position and orientation of the vehicles nearest neighbours, x_{NN}, y_{NN} and ψ_{NN}.

At this point, it is important to note that for work completed in this chapter, it is assumed that every vehicle has a continuously accurate representation of the positioning and orientation of every vehicle in the group, i.e. it is assumed that communication among the vehicles is instantaneous. The reason for implementing this assumption is to ensure that the formation control algorithm described and presented in this chapter initially operates as envisioned. Once this has been established, a realistic representation of the underwater communication channel will be incorporated in the subsequent chapters and its effect on the algorithm analysed.

Based on the structure of the *Guidance System* shown above, it is apparent that in order to successfully implement the behavioural mechanisms of fish as a suitable formation control algorithm, the *Autopilot* subsystem has to use the data available to it, i.e. x_v, y_v, x_{NN}, y_{NN} and ψ_{NN} to produce a desired heading angle, ψ_D that will manoeuvre the vehicle in either a *repulsive*, *orientating* or *attractive* manner. The functionality implemented to achieve this will be referred to as the *attractive*, *repulsive* and *orientating* control laws and are described in the following three sections.

4.3.1 Attractive Behaviour Control Law

Fish implementing the *attractive* behaviour manoeuvre towards their nearest neighbour(s) in order to reduce the *absolute* distance between themselves and their nearest neighbour(s).

As shown below in Equations (4.1) and (4.2), the control law implemented to achieve this *attractive* behaviour evaluates a waypoint which is equal to the average position of the vehicles nearest neighbour(s).

$$x_d = \frac{1}{I_{NN}} \sum_{NN=1}^{I_{NN}} x_{NN} \tag{4.1}$$

$$y_d = \frac{1}{I_{NN}} \sum_{NN=1}^{I_{NN}} y_{NN} \tag{4.2}$$

Where, as discussed above, x_{NN} and y_{NN} represent the position of the vehicles nearest neighbour(s) and I_{NN} is the number of neighbours taken into consideration within the control law. As shown below in Equation (4.3), the desired heading angle of the vehicle is then calculated using the same standard Line-of-Sight calculation presented previously in Section 3.6.2.1 of Chapter 3.

$$\tan(\psi_D) = \left(\frac{y_d - y_v}{x_d - x_v}\right) \tag{4.3}$$

Equations (4.1) - (4.3) illustrate that the *attractive* behavioural control law operates by implementing a standard Line-of-Sight calculation whereby the vehicles desired location is calculated to be the average position of its nearest neighbour(s). By calculating the vehicles desired location in this way, it ensures that regardless of the number if nearest neighbours taking into consideration, the vehicle is guaranteed to manoeuvre towards its nearest neighbours.

4.3.2 Repulsive Behaviour Control Law

The purpose of the *repulsive* behavioural control law is to ensure that neighbouring vehicles manoeuvre away from one another and as a result, the associated *absolute* distance between them increases. To achieve this, the following conditional control law is proposed.

$$\psi_D = \begin{cases} \left(\frac{NN_L}{NN_L + NN_R}\right)\frac{\pi}{2} & ,if\ NN_L > NN_R \\ -\left(\frac{NN_R}{NN_L + NN_R}\right)\frac{\pi}{2} & ,if\ NN_L < NN_R \\ 0 & ,if\ NN_L = NN_R \end{cases} \tag{4.4}$$

Here, NN_L and NN_R represent the number of nearest neighbours positioned to the left and right-hand side of the vehicle.

When using the *repulsive* behavioural control law, Equation (4.4) demonstrates that the vehicles are guided in the direction containing the least number of neighbouring vehicles. The inclusion of the ratios shown in Equation (4.4) will result in each vehicle manoeuvring with a heading angle based on its position relative to its nearest neighbours. This ensures that neighbouring vehicles will always manoeuvre away from one another when implementing the *repulsive* behavioural control law.

With the inclusion of the parameters, NN_L and NN_R in Equation (4.4), it is evident that each vehicle needs to be able to establish how many of its nearest neighbours are positioned to its right or left hand side. As shown below in Figure 4.3, this can be achieved by using the Line-of-Sight technique introduced earlier in Equation (4.3).

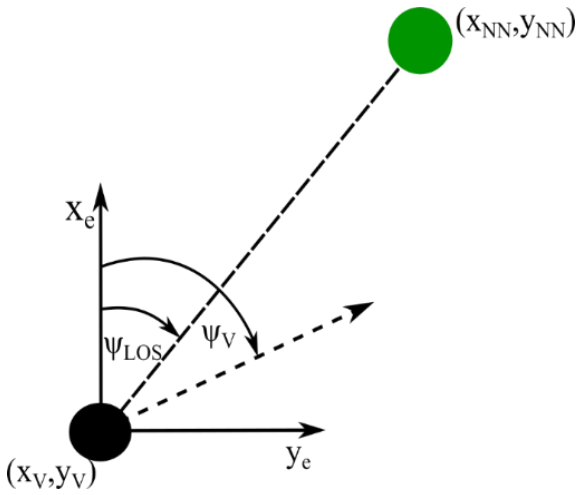


Figure 4.3 - Diagram demonstrating the effect that the vehicles heading angle has on determining the relative positioning of nearest neighbours.

Analysing Figure 4.3, it becomes apparent though that if only Equation (4.3) is used, the angle produced (ψ_{LOS} in Figure 4.3) will result in the neighbouring vehicle being categorised as being positioned to the vehicle's right hand side, i.e. $0^o \leq \psi_{LOS} \leq 180^o$. However, as Figure 4.3 illustrates, due to the heading angle of the vehicle, i.e., ψ_V, the vehicles nearest neighbour is actually positioned to its left. Therefore, in order to correctly identify the relative position of each vehicle's nearest neighbour, the following equation has to be implemented.

$$\psi_{R_{NN}} = \psi_{LOS} - \psi_V \tag{4.5}$$

As shown below in Figure 4.4, once Equation (4.5) has been evaluated, the angle $\psi_{RNN,}$ can be used to accurately determine how many of the vehicles nearest neighbour(s) are positioned to its right or left-hand side. This is achieved by using the conditional statements shown below.

$$NN_L = NN_L + 1 \quad ,if \quad -180^{\circ} \leq \psi_{R_{NN}} \leq 0^{\circ} \tag{4.6}$$

$$NN_R = NN_R + 1 \quad ,if \quad 0^{\circ} < \psi_{R_{NN}} < 180^{\circ} \tag{4.7}$$

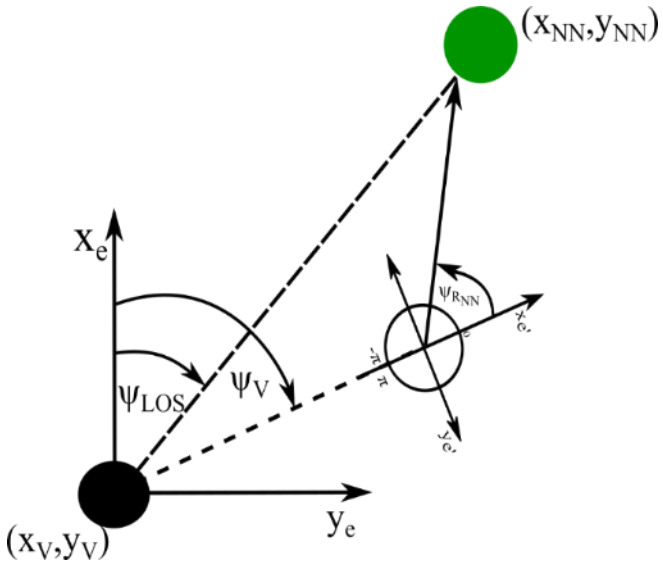


Figure 4.4 - Axis rotation required to determine relative positioning of nearest neighbour(s) (Exaggerated for clarity purposes).

4.3.3 Orientating Behaviour Control Law

The purpose of the *orientating* behavioural control law is to ensure that once within the orientation zone, each vehicle should seek to manoeuvre with the same heading angle as that of its nearest neighbour(s). To achieve this, the following control law is proposed.

$$\psi_D = \frac{1}{I_{NN}} \sum_{NN=1}^{I_{NN}} \psi_{NN} \tag{4.8}$$

Equation (4.8) demonstrates that when implementing the *orientating* behavioural control law, the desired heading angle of each vehicle is calculated to be the average heading angle of its nearest neighbour(s).

4.3.4 Velocity Control Law

As well as implementing the various heading control laws presented above, it is also necessary to incorporate an appropriate velocity control law. The purpose of the velocity control law will be to ensure that each vehicle converges to using the *orientating* behavioural control law as quickly as possible. To achieve this, the velocity control law implemented not only has to take into consideration the *absolute* distance between neighbouring vehicles but also their relative positioning as shown below in Figure 4.5.

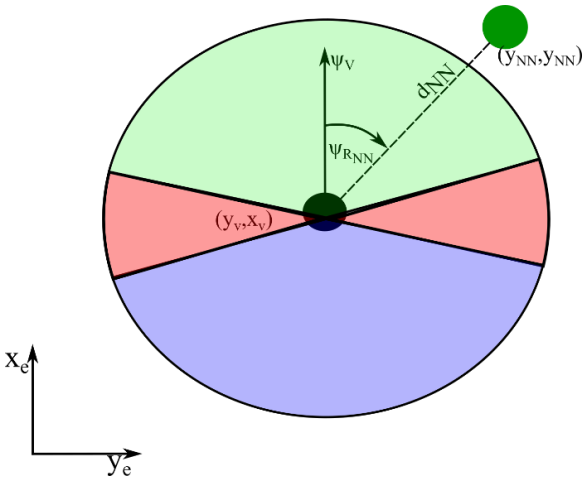


Figure 4.5 - Diagram representing the discretisation of the area surrounding each vehicle into three sections: Front (Green), Behind (Blue) & Beside (Red).

Figure 4.5 demonstrates that by defining angular limits for the parameter, ψ_{RNN}, the relative position of each vehicles nearest neighbour can be discretised into three sections: *front*, *behind* and *beside*. As shown below in Table 4.2, by combining the above categorisation with the *absolute* distances between neighbouring vehicles, i.e. d_{abs}, a suitable velocity control law can be created.

Table 4.2- Velocity Control Law implemented within formation control algorithm.

Nearest Neighbour Position	$d_{abs} > zoo_u$	$d_{abs} < zoo_l$	$zor \leq d_{abs} \leq zoa$
In Front	0.15	0.05	0.1
Behind	0.05	0.15	0.1
Beside	0.1	0.1	0.1

The table presented above demonstrates that depending on the relative positioning of neighbouring vehicles, ψ_{RNN} and the associated *absolute* distance between the two vehicles, d_{abs}, the velocity control law will instruct the vehicle to either slow-down, speed-up or remain at the nominal velocity. For example, if the autopilot calculates the angle ψ_{RNN} to be such that it defines a vehicles nearest neighbour to be ahead of it and the *absolute* distance between the vehicles, d_{abs} to be greater than the upper limits of the orientation zone (zoo_u in Table 4.2), the autopilot will change the tail beat amplitude of the vehicle to 0.15m (15cm). This will result in the vehicle accelerating in the direction of its nearest neighbour and thus, reducing the *absolute* distance between the two vehicles as required. Similarly, if the same

vehicle's nearest neighbour is positioned to the rear, the velocity control law will instruct the vehicle to slow-down.

4.3.5 Algorithm Structure

While the previous four sections presented the heading and velocity control laws implemented within the formation control algorithm, the purpose of this section is to illustrate the order in which the algorithm completes the various tasked summarised in Table 4.1. To achieve this, shown below in Figure 4.6 is the structure of the formation control algorithm detailing the stages at which certain parameters are evaluated at and the relevant decision making processes undertaken.

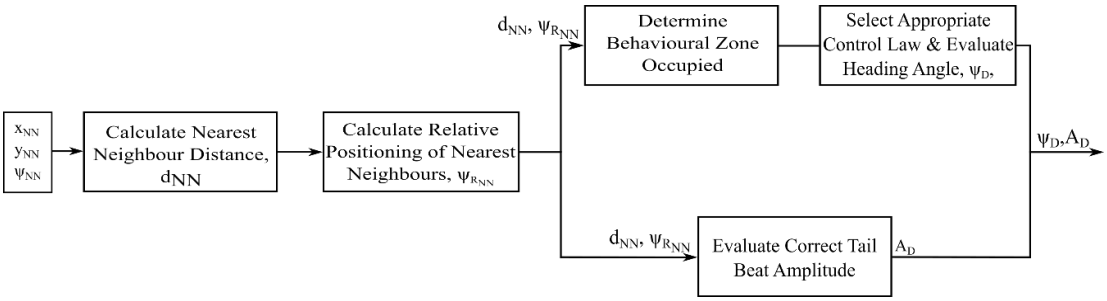


Figure 4.6 - Algorithm Structure.

The diagram demonstrates that the first two tasks completed by the algorithm is the evaluation of the *absolute* distance between each vehicle and its nearest neighbour(s), d_{abs}, as well as the associated relative positioning, $\psi_{R_{NN}}$. Thereafter, the algorithm uses these values to not only determine which of the three behavioural control laws to implement but also which *Tail-Beat Amplitude,* A_D to select in order to alter the vehicles velocity. Lastly, the algorithm implements the appropriate control law and calculates the desired heading angle of the vehicle, ψ_D.

The evaluation of the *absolute* nearest neighbour distance shown in Figure 4.6 is completed using the standard equation for the distance between two points in two dimensional space as shown below:

$$d_{NN} = \sqrt{(x_v - x_{NN})^2 + (y_v - y_{NN})^2} \tag{4.9}$$

The selection of which control law to implement is achieved using conditional statements similar to those shown above in Equations (4.6) and (4.7). This time though, these statements determine whether the value calculated for d_{abs} is above, below or within the orientation zone. It is also important to note that if multiple behavioural zones are

simultaneously occupied by various neighbouring vehicles, priority is given to the zone closest to the vehicle.

Finally, the calculations and processes presented above in Section 4.3.1-5 are all evaluated within the *Autopilot* subsystem of the BAUV and as such, they are evaluated at the same rate as the vehicle's *Guidance System*, i.e. at 4 Hz.

4.4 Simulation Setup and Performance Metrics

The heading and velocity control laws presented above have been designed specifically to allow a group of BAUVs to mimic the behavioural mechanisms of fish within school structures. In doing so, it is anticipated that the formation control algorithm will be able to coordinate the vehicles into a stable group structure that will also allow the vehicles to complete the oceanic surveying missions outlined in Figure 1.3. To demonstrate whether or not this is indeed the case, the algorithm has to be thoroughly tested and the effect of varying key algorithmic parameters analysed. The following section will now describe the simulation setup and algorithmic parameters varied to achieve this.

4.4.1 Simulation Setup

As discussed above, the purpose of the simulations completed in this study is to establish whether or not the heading and velocity control laws presented in Section 4.3 operate as envisioned. Furthermore, the effect of varying certain algorithmic parameters on the algorithms ability to coordinate the group as required also has to be established. To achieve this, two groups of simulations were undertaken. The first group was designed specifically to test the *repulsive* control law while the second group was designed to test *attractive* control law. Due to the design of the algorithm, the *orientating* and velocity control laws will inherently be tested throughout both groups of simulations. To create these two groups of simulations, the vehicle's initial positions were chosen specifically to ensure that at the start of the simulation, the majority of vehicles would have to incorporate either the *repulsive* or *attractive* l behavioural control laws.

As well as ensuring that the various control laws were thoroughly tested, it was also important to understand what effect varying the algorithmic parameter, I_{NN} has on the algorithms ability to form a stable group structure. As a result, for the two groups discussed above, this parameter has been varied from one through to *N-1*, where N represents the number of vehicles within the group. By doing this, all possible values with regard to this algorithmic parameter can be investigated. Subsequently, shown below in Table 4.3 is a

summary of the algorithmic parameters used throughout the course of the simulations completed within this study.

Table 4.3- Simulation Parameters.

Parameter	Group 1 (Repulsive)	Group 2 (Attractive)
Behavioural Zones Size (m)	[20,25,200]	[6,11,200]
Nearest Neighbours Utilised (I_{NN})	1,2,4,6,8,11	1,2,4,6,8,11

In order to thoroughly test the control laws created, for each combination of parameters presented in Table 4.3, 100 simulations were completed that varied the initial starting position of each vehicle. While each vehicle had different starting positions throughout the 100 simulations, the values chosen ensured that the vehicles would still initially implement either the *repulsive* or *attractive* control laws. Therefore, overall, 1200 simulations where completed throughout the course of this particular study.

As inferred by the values used for I_{NN}, the size of the group simulated in this study is equal to twelve. The reason for using this particular group size is due to two factors. Firstly, twelve vehicles is approximately double the size of any other realistic investigation into implementing a successful formation control algorithm in the underwater environment (Das et al. 2016). Secondly, while Chapter 3 demonstrates that the reduced fidelity model is significantly faster than the original high-fidelity model, its execution time still increases with the number of vehicles simulated. A group of twelve vehicles was therefore deemed to be a suitable compromise between these two factors.

4.4.2 Performance Metrics & Analysis Tools

In order to determine whether or not the algorithm has been successful, four performance metrics have been used; the standard deviation of the vehicles heading angle as well as the maximum, mean and minimum *absolute* nearest neighbour distances.

The reason for calculating the standard deviation is to demonstrate whether or not the individual vehicles have all been able to converge towards manoeuvring with the same heading angle. The maximum, mean and minimum *absolute* nearest neighbour distances meanwhile are evaluated to demonstrate whether or not the distance between each vehicle and its nearest neighbour has converged to a value within the confines of the orientation zone.

While the evaluation of the four parameters discussed above is relatively simple, the following sections describe how they have been calculated to represent the data obtained from the 100 simulations completed for each unique combination of the parameters presented in Table 4.3.

4.4.2.1 Evaluation of the Standard Deviation of Vehicle Heading Angle

Shown below is the equation used to evaluate the evolution of the standard deviation of the vehicles heading angle for the simulations completed in this study (Mathworks 2006b).

$$\sigma_{t,j} = \sqrt{\frac{\sum_{i=1}^{N}\left(\psi_{i,t,j} - \overline{\psi}_{t,j}\right)^2}{N-1}} \tag{4.10}$$

Here N represents the number of vehicles within each simulation (i.e. twelve), $\psi_{i,t,j}$ is the heading angle for vehicle i at time t in simulation j and $\overline{\psi}_{t,j}$ is the average heading angle of the group at time t in simulation j.

Once evaluated $\sigma_{t,j}$ represents the standard deviation of the vehicles heading angle at time, t in simulation, j. In order to provide a metric which represents the evolution of this parameter across the 100 simulations completed, the following equation is implemented.

$$\overline{\sigma}_t = \frac{1}{k}\sum_{j=1}^{100}\sigma_{t,j} \tag{4.11}$$

Here, k represents the number of simulations completed (i.e. 100) and $\sigma_{t,j}$ is the standard deviation of the vehicles heading angle at point t in simulation j calculated using Equation (4.10). As a result, $\overline{\sigma}_t$ represents the average standard deviation of the vehicle's heading angle at time t, as calculated across the 100 simulations completed for each value of I_{NN}.

In order for the formation control algorithm to be deemed to have generated a stable group structure, the converged value for this parameter should not exceed 1°. This criteria was selected as it represents the typical accuracy of a standard compass used in underwater vehicles (Paull et al. 2014).

4.4.2.2 Evaluation of Maximum, Mean & Minimum Nearest Neighbour Distance

As shown below in Equation (4.12), to calculate the distance between each vehicle and its nearest neighbour, the standard equation for measuring the distance between two points in 2-dimensional space is used.

$$d_{i,t,j} = \sqrt{\left(x_{i,t,j} - x_{NN_{t,j}}\right)^2 + \left(y_{i,t,j} - y_{NN_{t,j}}\right)^2} \tag{4.12}$$

Here $x_{i,t,j}$ and $y_{i,t,j}$ represent the x and y position of each vehicle, i at time t in simulation j. Meanwhile, $x_{NN_{t,j}}$ and $y_{NN_{t,j}}$ represent the position of vehicle i's nearest neighbour at time t in simulation j. As a result, $d_{i,t,j}$ represents the distance between each vehicle i and its nearest neighbour at time t in simulation j.

Therefore, $d_{i,t,j}$ can now be thought of as a multi-dimensional array that represents the distance between each vehicle and its nearest neighbour every 0.1s throughout the 100 simulations completed for each value of I_{NN}. In order to evaluate the average *absolute* nearest neighbour distance across the 100 simulations completed, the following equation is used.

$$\overline{d}_{abs_t} = \frac{1}{k}\sum_{j=1}^{k}\frac{1}{N}\sum_{i=1}^{N} d_{i,t,j} \tag{4.13}$$

Once again, N represents the number of vehicles within each simulation and k is the number of simulations completed (i.e. 100).

It's apparent from Equation (4.13) that the average nearest neighbour distance from within each simulation is evaluated first. This produces a two dimensional array where each column represents the evolution of the average nearest neighbour distance as obtained from simulation j. Afterwards, this value is then averaged across the 100 simulations completed to produce the mean nearest neighbour distance every t seconds throughout all the simulations completed.

The maximum and minimum distances are evaluated in an identical way to Equation (4.13) except instead of evaluating the mean value each time, the maximum and minimum values are calculated instead. As expected, the convergence criteria associated with the evaluation of the minimum, mean and maximum *absolute* nearest neighbour distances is that they all converge to a value that is within the confines of the orientation zone.

Combined with the evaluation of the average standard deviation described above in Equation (4.10), the data obtained using the methods described in Equation (4.10) – (4.13) will provide a definitive and concise method by which to analyse whether or not the formation control algorithm operates as designed.

4.5 Results

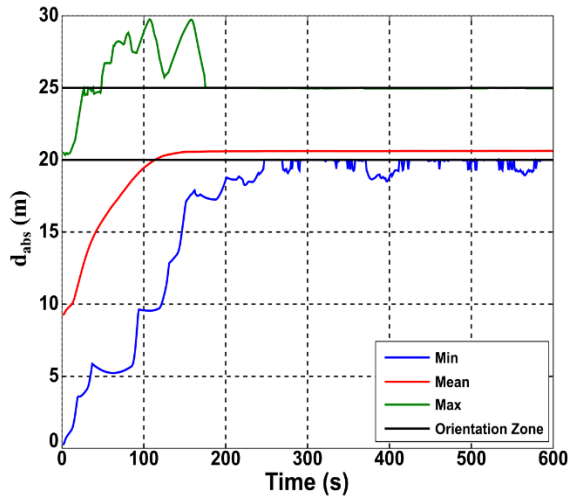
There are three main objectives related to the work completed in this chapter. The first is to determine whether or not the formation control algorithm presented in Section 4.3 operates as envisioned and is capable of coordinating the vehicles into a stable group structure. In this work, a stable group structure is defined as one in which the standard deviation of each groups heading angle is less than or equal to 1º and the value obtained for d_{abs} for each vehicle converges to within the confines of the orientation zone. The second objective is to ascertain what effect varying the algorithmic parameter, I_{NN} has the algorithms ability to generate this stable group structure. The final objective is to assess the ability of the algorithm to generate the parallel line formation required for the oceanic surveying missions as outlined in Chapter 1.

To demonstrate whether or not these objectives have been achieved, the results presented below are separated into three sections. The first section analyses the results obtained from the simulations where every vehicle is initially using the *repulsive* control law. The second section then presents and analyses the results obtained from the simulations where every vehicle is initially using the *attractive* control law. Finally, the last section demonstrates whether or not the algorithm has coordinated the vehicles into the required parallel line formation.

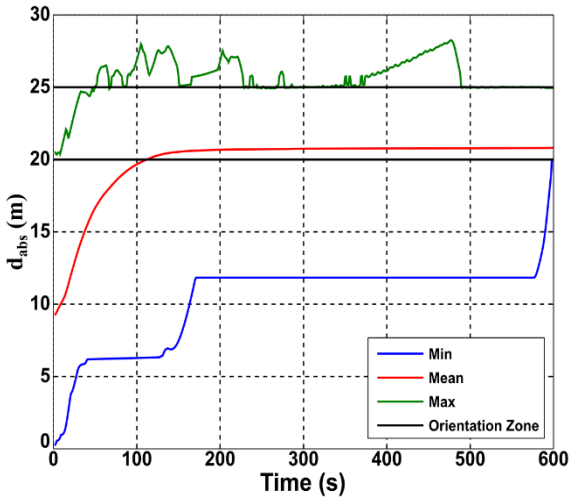
4.5.1 Analysis of Repulsive & Orientating Control Laws

As discussed above, the first objective of this work is to demonstrate whether or not the formation control algorithm has been successful in generating a stable group structure. To achieve this, shown below in Figure 4.7 is the evolution of the minimum, mean and maximum values obtained for the parameter, d_{abs} as I_{NN} is increased. As discussed above, because the minimum, mean and maximum values have been calculated, the results presented below represent the entire envelope of results obtained from the 100 simulations completed for each value of I_{NN}.

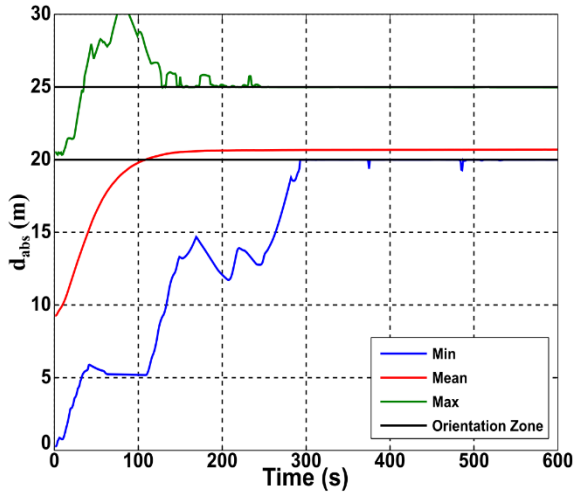
On initial inspection, the results presented below demonstrate that regardless of the value used for $I_{NN,}$ the evolution of the mean value for d_{abs} is as expected, with the value initially increasing before converging to a value that is within the confines of the orientation zone. Furthermore, the results also illustrate that the converged value for this parameter is always positioned close to the lower boundary of the orientation zone.



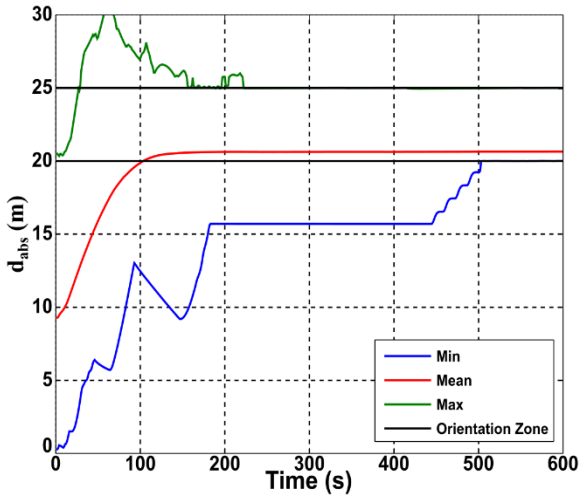
(a) $I_{NN} = 1$



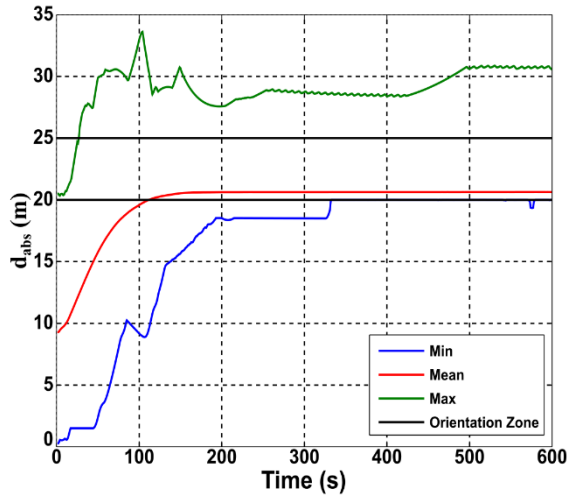
(b) $I_{NN} = 2$



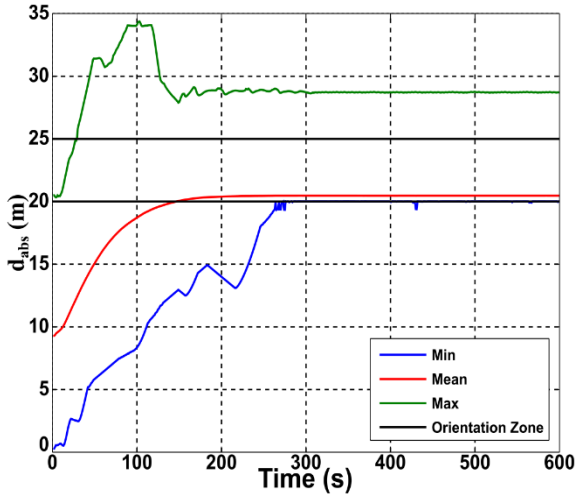
(c) $I_{NN} = 4$



(d) $I_{NN} = 6$



(e) $I_{NN} = 8$



(f) $I_{NN} = 11$

Figure 4.7- Evolution of the minimum, mean and maximum d_{abs} values as I_{NN} is increased.

Once again, this result is to be expected and is due to the fact that because the majority of the vehicles were initially implementing the *repulsive* control law, their d_{abs} value when switching to the *orientating* control law will be by default, at the lower boundary of the orientation zone.

Nevertheless, as with the results presented with the mean value for d_{abs}, these results also illustrate that the minimum d_{abs} value increases and then converges to value that is within the confines of the orientation zone. It is important to note that the evolutions of the minimum and mean values discussed above all initially start from a value that is within the confines of the repulsion zone and then gradually increase to converge to a value within the orientating zone. Therefore, these results (minimum and mean d_{abs} values) all suggest that the *repulsive* and *orientating* control laws operate as required regardless of the value used for I_{NN}.

Importantly though, the results presented in Figures 4.7 (e) and (f) demonstrate that the evolution of the overall maximum d_{abs} value has been unable to ensure every vehicle converges to a value that is within the confines of the orientation zone. Instead, the results clearly illustrate that a certain number of vehicles have diverged to a d_{abs} value that is within the confines of the attraction zone. Moreover, the result also demonstrate that these vehicles are subsequently unable to converge back within the confines of the orientation zone.

These results suggest that the *attractive* control law presented in Section 4.3.1 is unable to manoeuvre every vehicle as required when I_{NN} is equal to either eight or eleven. Due to the fact that this non-convergence is associated with the implementation of the *attractive* control law, these results will be discussed in more detail in Section 4.5.2 along with the results obtained from the second group of simulations discussed in Table 4.3. For the remainder of this section, the results will focus solely on the results obtained from simulations where only the *repulsive* and *orientating* control laws were implemented.

Consequently, while the results presented in Figure 4.7 suggest that the *repulsive* and *orientating* control laws are able to operate as required regardless of the value used for I_{NN}, these results don't take into consideration the evolution of the standard deviation of the vehicles heading.

Subsequently, shown below in Figure 4.8 is the evolution of the standard deviation of the vehicles heading angle as the value used for I_{NN} is increased. While the results presented below are initially as expected with the value increasing as the vehicles implement the *repulsive* control law, the results clearly demonstrate that the standard deviation value

doesn't always – as required - converge to a value equal to approximately zero, particularly when I_{NN} is equal to one or two.

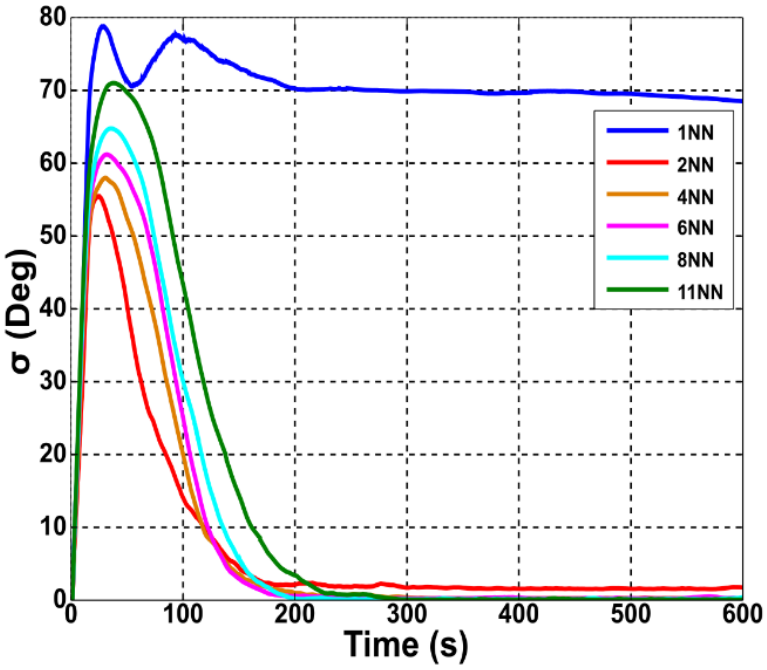
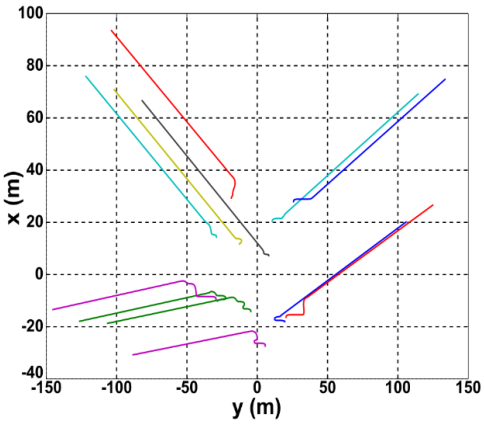


Figure 4.8 - - Evolution of σ as I_{NN} is increased.

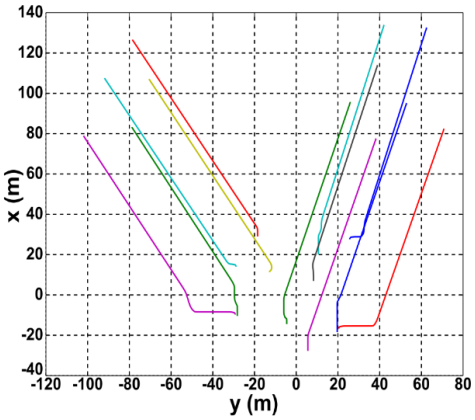
These results contradict those presented in Figure 4.7 (a) and (b) where the results clearly illustrate that when I_{NN} is equal to one or two, the minimum, mean and maximum d_{abs} values all converge to a value within the orientation zone. The only explanation that satisfies these contradictory results is that while the *repulsive* control law operates as expected *and* regardless of the value used for I_{NN}, the ability of the *orientating* control law improves as the value for I_{NN} is increased.

This is shown to be the case below in Figure 4.9 where the different trajectories obtained as I_{NN} is increased are presented. Before analysing the results presented below, it is important to emphasise once again that the only initial condition varied from Figure 4.9 (a) to (d) is the value used for I_{NN}.

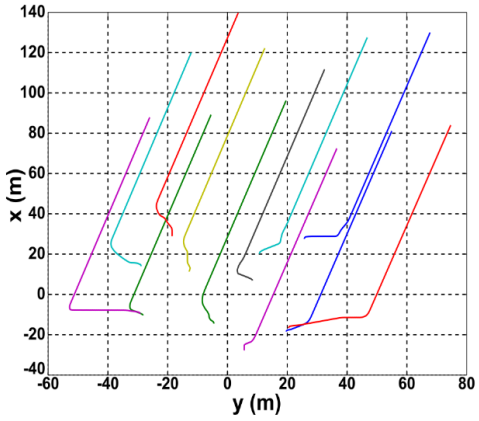
On inspection, the results presented below in Figures 4.9 (a) and (b) clearly demonstrate that instead of the entire group of BAUVs moving in the same direction when the *orientating* control law is implemented, a number of subgroups have formed, with each group moving in a different direction. This formation of multiple sub-groups explains why in Figure 4.8, the standard deviation of the vehicles heading angle isn't capable of converging below 2° when I_{NN} is equal to either one or two.



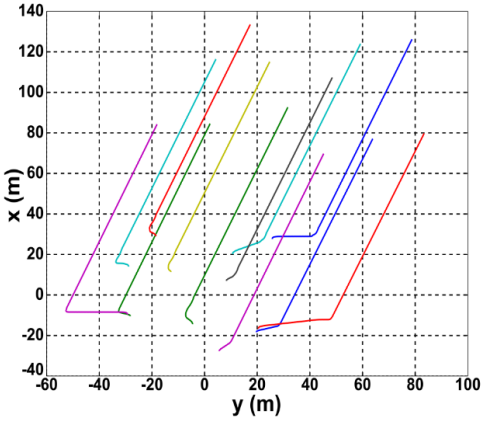
(a) - $I_{NN} = 1$



(b) - $I_{NN} = 2$



(c) - $I_{NN} = 4$



(d) - $I_{NN} = 11$

Figure 4.9- Variation in vehicle trajectories as I_{NN} is increased

Thereafter, the trajectories presented in Figures 4.9 (c) and (d) suggest that once the value for I_{NN} is equal to or greater than four, the *orientating* control law is able to ensure every vehicle converges to approximately the same heading angle and hence, a single stable group structure can be generated. These results suggest that in order for the *orientating* control law to operate as required, the value for the algorithmic parameter, I_{NN} needs to be set to *at least* four. To definitively demonstrate whether or not this is the case, the following section will use Graph Theory to determine the minimum criteria associated with this parameter that will ensure the formation of a single group structure.

4.5.1.1 Achieving Consensus using the Orientating Control Law

In order to implement Graph Theory to determine the smallest value of I_{NN} that ensures every vehicle will converge to the same heading angle, the *orientating* control law shown above in Equation (4.8) has to be modified to include a mathematical representation of the vehicles interaction topology. This modification is shown below in Equation (4,14).

$$\psi_{D_i} = \frac{1}{I_{NN}} \sum_{j=1}^{I_{NN}} A_{ij}[\psi_j] \quad \forall j = 1 \ldots, I_{NN} \tag{4.14}$$

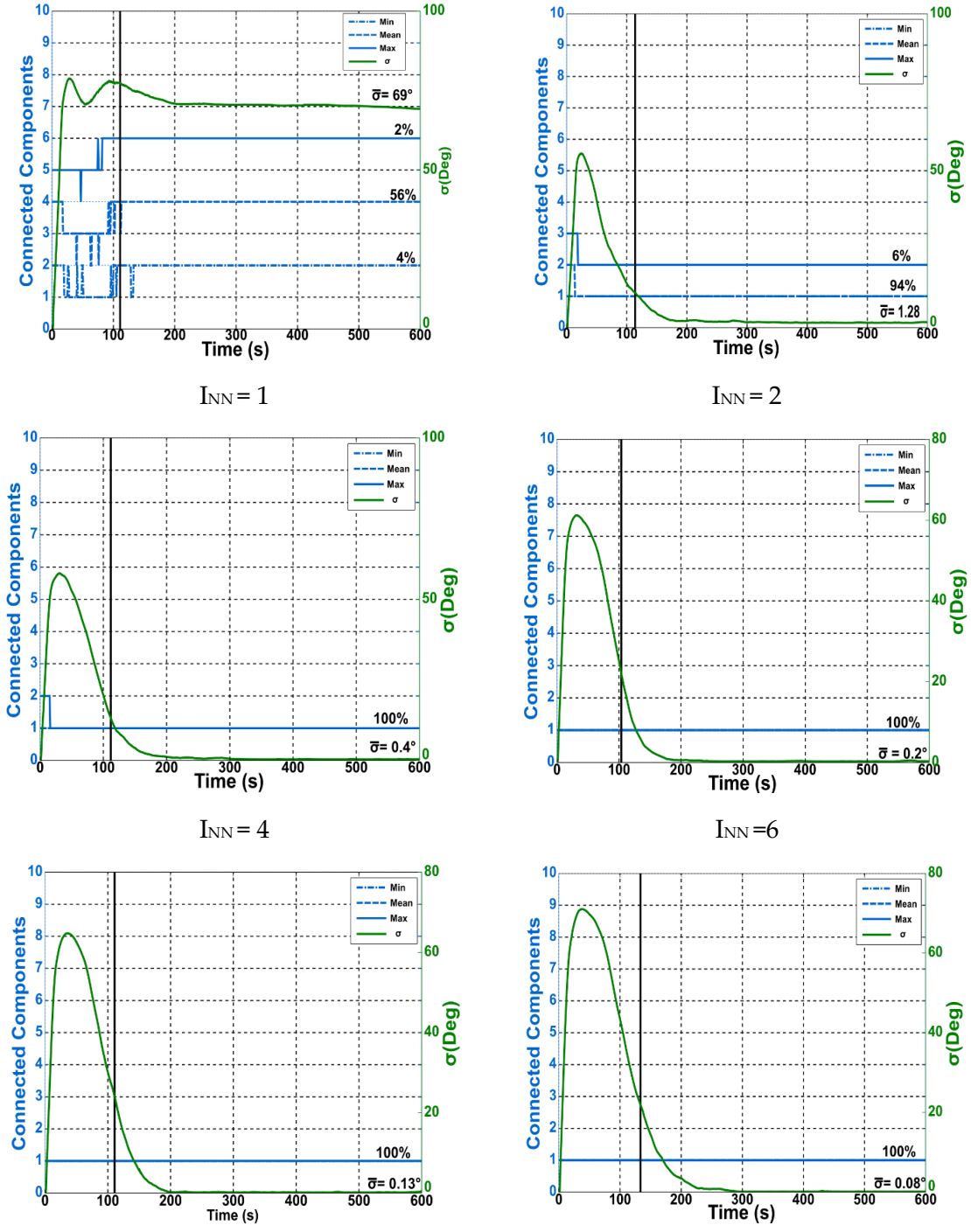
As with Equation (4.8), ψ_{D_i} represents the desired heading angle of vehicle i, ψ_j is the heading angle of vehicle i's j[th] nearest neighbour and A_{ij} is the adjacency matrix.

The adjacency matrix is a component of the Laplacian and is used to represent whether or not a link or connection exists between two nodes in the same graph. In this particular instance, the adjacency matrix represents whether or not neighbouring vehicles use one another's heading data when calculating their desired heading angle. One of the most useful features of the adjacency matrix is that it assists in providing a method by which to determine the particular type of graph being produced. This can be achieved by equating the number of times zero appears as an eigenvalue of the Laplacian Matrix as being equal to the number of connected components (subgroups) within that particular graph.

For the results presented in Figure 4.9, the above feature was implemented and the evolution of the number of connected components (subgroups) within each simulation evaluated. The results obtained are shown below in Figure 4.10 alongside the associated evolution of the average standard deviation of the vehicles heading angle. Also shown in Figure 4.10 is the percentage of simulations containing a particular number of connected components (subgroups), i.e. Figure 4.10(a) demonstrates that 56% of the simulations converged to an interaction topology containing four connected component, i.e. four subgroups.

The results presented below demonstrate a clear correlation between the value used for I_{NN}, the number of connected components in the underlying interaction graph and the converged standard deviation of the vehicles heading angle. For example, Figure 4.10 (a) demonstrates that when each vehicle only takes into consideration its single nearest neighbour, 56% of the simulations produced a graph topology containing four connected components with the resulting converged standard deviation being equal to approximately 69º. Conversely, Figure 4.10(d) demonstrates that when each vehicle takes into consideration its six nearest neighbours, the underlying graph for each group always contains a single connected component and the resulting converged average standard deviation reduces drastically to 0.2º. The reason for this variation is due to the fact that when Equation (4.8) is implemented, ψ_D converges to the average heading angle of the neighbouring vehicles it is interacting with. As a result, when there are multiple connected components (subgroups) as in Figure 4.10(a), each vehicle converges to a heading angle that

is equal to the average heading angle of that particular subgroup of connected vehicles. This is why in Figures 4.10 (a) and (b), multiple subgroups have been formed that all manoeuvre in different directions.



$I_{NN} = 1$ $I_{NN} = 2$

$I_{NN} = 4$ $I_{NN} = 6$

$I_{NN} = 8$ $I_{NN} = 11$

Figure 4.10- Evolution of the number of connected components and σ as I_{NN} is increased.

Based on the results presented above, it is apparent that in order to guarantee that the *orientating* behavioural control law operates as intended, the underlying graph representing

the interaction topology of the vehicles *must* always contain a *single* connected component. As the results shown in Figure 4.10 (d), (e) & (f) demonstrate, this means that I_{NN} has to be equal to *at least* six when the group consists of twelve vehicles.

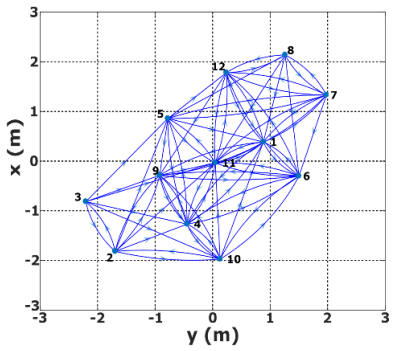
This result is not surprising and has been published several times in the literature associated with determining the minimum requirements for achieving consensus within multi-vehicle systems (Ren, Wei, Beard,R.W, Atkins 2007; Ren & Beard 2008; Jadbabaie et al. 2003; Ren & Beard 2005; Ren & Beard 2004).

However, while I_{NN} being equal to half the group size is the minimum requirement for achieving consensus, the results presented in Figure 4.10 demonstrate that improved performance can be achieved if I_{NN} is increased beyond this minimum value. This improved performance is illustrated by the fact that the average converged standard deviation value decreases from 0.2º to 0.08º when I_{NN} is increased from six to eleven. Furthermore, the results presented below in Table 4.4 illustrate that the associated time taken for the vehicles to agree on a desired heading angle once every vehicle is using the orientating control law decreases as I_{NN} is increased.

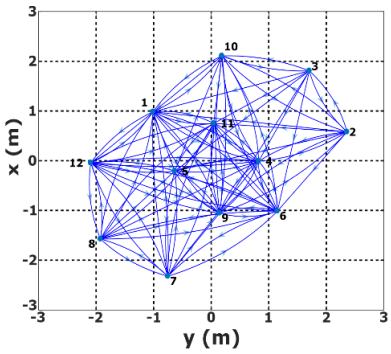
Table 4.4- Variation in consensus time as I_{NN} is increased when the *Orientating* control law is being used

I_{NN}	**Mean Consensus Time (s)**
6	20
8	19
11	16

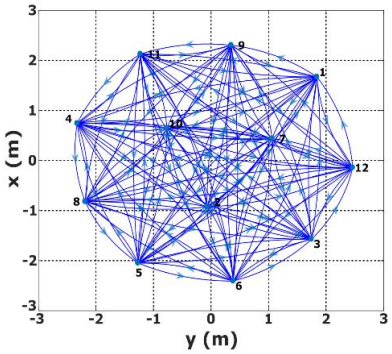
As the graphs presented below in Figure 4.11 demonstrate, this improved performance is due to the fact that as the value for I_{NN} is increased, the connectedness of the underlying interaction topology gradually increases. Therefore, with each vehicle having access to more neighbouring vehicle's data, the time taken for Equation (4.8) to converge to the same heading angle will reduce and hence, a stable group structure can be generated faster. Therefore, these results demonstrate that the *orientating* control law operates most efficiently when I_{NN} is equal to 11.



(a) Inn = 6



(b) Inn = 8



(c) Inn = 11

Figure 4.11- Variation in the underlying graph topology when I_{NN} is equal to (a) 6 (b) 8 and (c) 11.

This result however, is in direct contrast to the *repulsive* control law where the results presented below in Table 4.5 demonstrate that the time taken for the vehicles to converge to implementing the *orientating* control law increases as the value for I_{NN} is increased. The results illustrate that on average, the time taken for every vehicle to begin implementing the *orientating* control law increases by 36s when the value used for I_{NN} is increased from 6 to 11. The reason for this is due to the fact that as I_{NN} is increased, the denominator of Equation (4.4) increases. This results in the relative angle between neighbouring vehicle's trajectories reducing and as a result, the rate at which neighbouring vehicles manoeuvre away from one another also reducing.

Table 4.5 - Variation in time spent using *Repulsive* control law as I_{NN} is increased.

I_{NN}	**Time**
6	115
8	123
11	151

In summarising, the results presented above demonstrate that in order to ensure that the formation control algorithm is capable of generating a stable group structure, the number of nearest neighbours used by each vehicle, i.e. I_{NN} has to be equal to *at least* six. In doing so, this ensure that the underlying interaction topology of the vehicles is always connected. Importantly though, the results presented in Table 4.4 and Figure 4.10 demonstrate that in order for the *orientating* control law to operate as efficiently as possible, the value for I_{NN} should be set to 11. However, as Table 4.5 illustrates, setting I_{NN} to this value for both the *repulsive* and *orientating* control laws results in the performance of the *repulsive* control law becoming suboptimal. Therefore, based on these results, it is apparent that if the *repulsive* and *orientating* control laws are to operate as efficiently as possible, the values used for the algorithmic parameter, I_{NN} cannot be the same and needs to be equal to six for the *repulsive* control law and eleven for the *orientating* control law.

4.5.2 Analysis of Attractive & Orientating Behavioural Control Laws

In the previous section, the results obtained from the simulations implementing the *repulsive* and *orientating* control laws were presented and analysed. The focus of this section is to now analyse the results obtained when the *attractive* and *orientating* control laws were used. Subsequently, shown below in Figure 4.12 is a general overview of the results obtained from these simulations.

Based on the results presented in the previous section it is unsurprising that the results presented below demonstrate that as I_{NN} is increased, the percentage of simulations satisfying the convergence criteria associated with the standard deviation of the vehicles heading angle increases and reaches a maximum when I_{NN} is equal to eleven. Surprisingly though, the results also illustrate that even when I_{NN} is equal to 11, not all the simulations undertaken were able to ensure the parameter, σ_t converged to a value approximating zero.

Nevertheless, when the results associated with both convergence criteria are compared, it is apparent that the convergence performance of the algorithm is at its best when I_{NN} is set to eleven. Subsequently, the majority of the work presented in this section will focus on these results.

Combining the convergence percentages of both performance metrics when I_{NN} is equal to eleven, it becomes apparent that there are a number of instances where the standard deviation convergence criterion has been satisfied yet the *absolute* nearest neighbour distance criterion hasn't. While not immediately apparent, it has also been discovered that there exists four instances (simulations) where the formation control algorithm has been

unable to satisfy both criteria. Subsequently, the following sections will analyse the results obtained from these simulations in order to ascertain the reasons for this non-convergence.

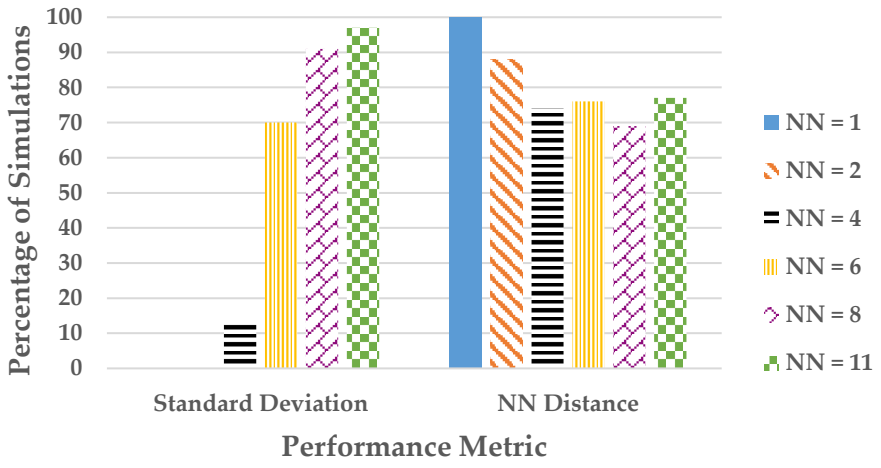


Figure 4.12 - Variation in the percentage of simulations satisfying the convergence criteria associated with σ and d_{abs} as I_{NN} is increased.

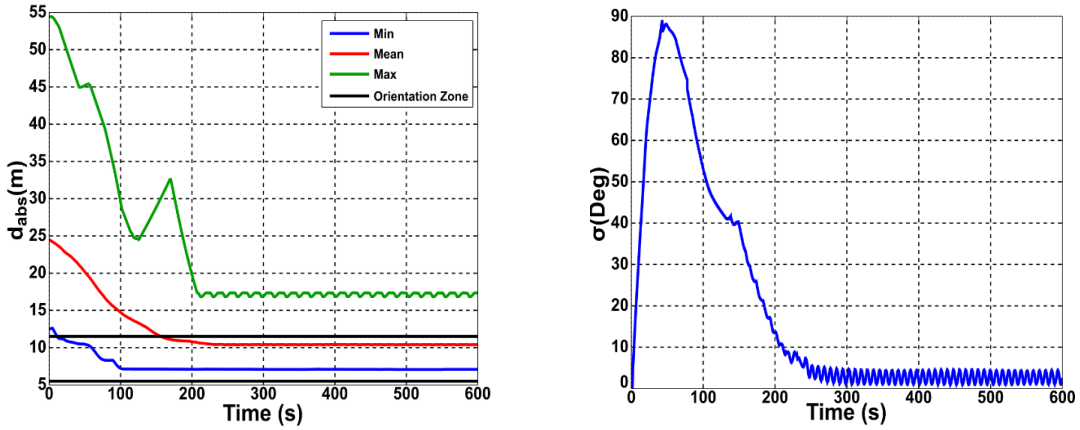
4.5.2.1 Analysis of Simulations Unable to Satisfy both Convergence Criteria

As discussed above, there exists four instances from the simulations undertaken where the *attractive* and *orientating* behavioural control laws were unable to satisfy the necessary convergence criteria. To understand why this is the case, these four simulations where identified and the results obtained analysed in greater detail. These results, as shown below in Figure 4.13 (a) represent the evolution of the minimum, maximum and mean *absolute* nearest neighbour distances as obtained from these four simulations. Similarly, Figure 4.13 (b) represents the evolution of the average standard deviation of the vehicles heading angle as obtained from these four simulations.

Due to the fact that both the minimum and mean *absolute* nearest neighbour distances have been able to converge to a value within the confines of the orientation zone, the results presented above suggest that it is only a small percentage of vehicles that have been unable to satisfy the necessary convergence criterions. Upon closer analysis, this was deemed to be the case and in actual fact, of the four simulations analysed, only a solitary vehicle from within each simulation was unable to converge as required.

However, the most intriguing facet of Figure 4.13 (a) is the convergence of the maximum *absolute* nearest neighbour distance to a value significantly greater than the upper threshold of the orientation zone. This convergence suggests that the *desired* heading angle produced from the *attractive* behavioural control law is no longer manoeuvring each of the four vehicles towards their respective nearest neighbours but conversely, in approximately the

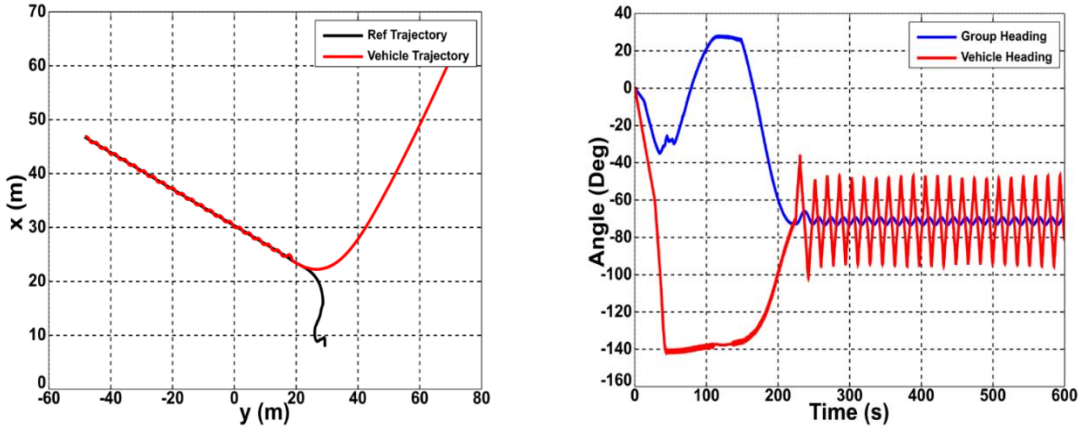
same direction. This is shown to be the case when the standard deviation of the vehicle's heading angle is taken into consideration in Figure 4.13 (b). The results clearly demonstrate that while σ reduces significantly, it no longer converges to a steady state value close to zero but instead, it oscillates around an average value of approximately 3º. Since it is already known that eleven out of the twelve vehicles are using the *orientating* behavioural control law, the individual vehicle still using the *attractive* behavioural control law must be causing this oscillation.



(a) (b)

Figure 4.13- Evolution of (a) minimum, mean & maximum values for d_{abs} and (b) σ.

To understand why these oscillations occur, shown below in Figure 4.14 (a) is a comparison of the reference trajectory calculated using the *attractive* behavioural control law (Equations (4.1) - (4.3)) and the *actual* trajectory of one of the four vehicles still using this *attractive* control law.



(a) (b)

Figure 4.14 - Comparison of (a) vehicle trajectory with reference trajectory and (b) vehicle heading angle with group heading angle.

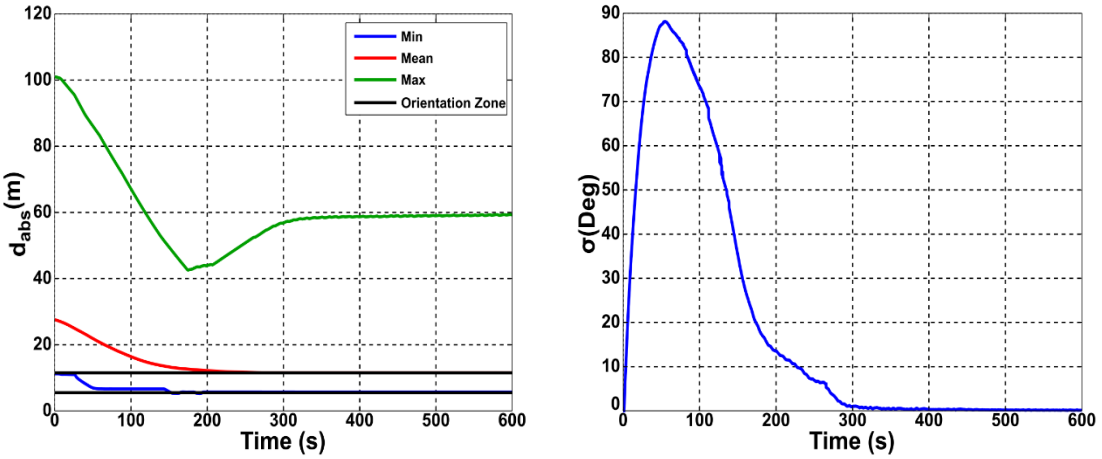
The trajectories presented above clearly demonstrate that the position of the vehicle quickly converges to approximately the same point in space as the average position of its nearest neighbours as calculated using Equations (4.1) & (4.2). Due to this close proximity, it is physically impossible for the vehicle to manoeuvre to the required position without overshooting. Nevertheless, the *attractive* control law continues to attempt to achieve this and as shown in Figure 4.14 (b), this results in the vehicle's heading angle continuously oscillating around the average heading angle of the group. Unsurprisingly, it is because of this large oscillation that the evolution of σ as shown in Figure 4.13 (b) is unable to converge close to zero.

The results presented in this section demonstrate that when the *attractive* control law is used, there is the possibility for the situation to arise whereby the *desired* and *actual* position of a vehicle are near coincident. When this occurs, Figure 4.14 (a), demonstrates the vehicle will continuously oscillate around the average heading angle of the group. While this behaviour appears in only 4% of the simulations tested, it is apparent that alterations will have to be made to the *attractive* behavioural control law in order to remove this behaviour and improve the performance of the algorithm.

4.5.2.2 Analysis of Simulations Unable to Satisfy Nearest Neighbour Distance Criterion

The results presented in this section will focus solely on the remaining nineteen simulations that *were* able to satisfy the convergence criterion for σ but *not* the vehicles nearest neighbour distance, d_{abs}. To achieve this, shown below in Figure 4.15 is the evolution of the minimum, mean and maximum nearest neighbour distances and the associated σ values as calculated (and averaged) from these nineteen simulations.

As expected, while Figure 4.15 (b) demonstrates that the vehicles have been able to converge toward a common heading angle, Figure 4.15 (a) demonstrates that a number of vehicles have been unable to successfully transition from using the *attractive* behavioural control law to the *orientating* one.



(a) (b)

Figure 4.15- Evolution of the (a) minimum, mean & maximum values for d_{abs} and (b) σ.

As in the previous section, in order to understand why this is the case, it is necessary to analyse the evolution of each vehicles trajectory with the associated reference trajectory calculated using the *attractive* behavioural control law. While the results presented above represent the average values across the nineteen simulations considered, the results shown below in Figure 4.16 are taken from individual simulations. Nevertheless, the results presented are representative of the trends discovered across the various simulations completed.

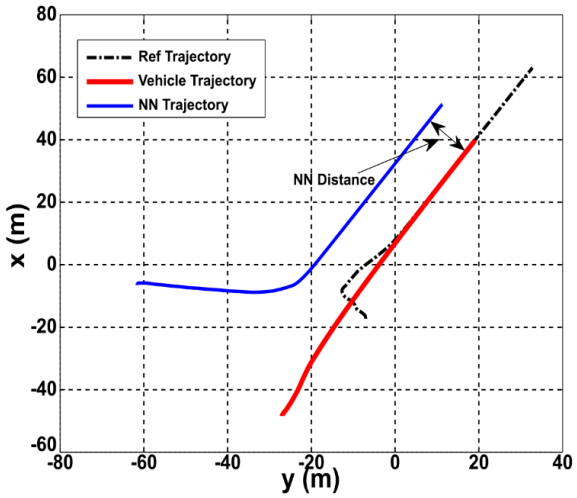


Figure 4.16 – Comparison of reference trajectory, vehicle trajectory & nearest neighbour trajectory when I_{NN} is equal to Eleven.

It is important to note that similarly to the previous section, the reference trajectory presented in Figure 4.16 is obtained from Equations (4.1) - (4.3) and is equal to the average position of the vehicles nearest neighbours in the x-y plane. However, unlike Figure 4.14 (a), the results demonstrate that the vehicles trajectory is no longer coincident with the

reference trajectory but instead, trails it by a significant distance. Furthermore, because every other vehicle within the group is using the *orientating* behavioural control law, this reference trajectory will be parallel with that of the remaining vehicles in the group. Therefore, because the vehicle is using the *attractive* control law, it too will converge to the same heading angle as the other vehicles in the group. It is for this reason that the evolution of σ as shown in Figure 4.15 (b) converges towards zero despite a vehicle continuing to use the *attractive* control law.

As the vehicle is now manoeuvring with a heading angle that is very similar to that of its nearest neighbour, the only way in which it can reduce d_{abs} further is by using the velocity control laws presented in Table 4.2. However, as shown in Figure 4.16, because of the relative positioning of the two vehicles, this will only be effective until the two vehicles are positioned alongside one another. Once this occurs, the vehicle will no longer be able of reduce its nearest neighbour distance any further. It is for this reason that for 19% of the simulations completed, the vehicles were able to satisfy the convergence criteria for σ but not $d_{abs.}$ It is also for these reasons that the results presented in Figures 4.7 (e) and (f) were unable to converge back to a d_{abs} value that is within the confines of the orientation zone.

In summarising, the previous two sections have presented the reasons why in 23% of the simulations completed, the *attractive* behavioural control law has been unable to simultaneously satisfy the necessary convergence criteria. As Figures 4.14 and 4.16 demonstrate, this non-convergence is due to a vehicle being positioned either directly on top of, or behind the position it evaluates to be the average position of its nearest neighbours from Equations (4.1) and (4.2). This demonstrates that the non-convergence is a result of selecting I_{NN} to be equal to eleven. However, crucially the results presented in Figure 4.12 demonstrate that improvements in the *attractive* control laws ability to satisfy the necessary criteria cannot be achieved by simply altering the value used for $I_{NN.}$

Therefore, these results suggest that the *attractive* control law, in its current guise, is unable to guarantee that a group of BAUVs will form a stable group structure. As a result, it is apparent that alterations will have to be made to this control law in order to ensure it can guarantee the formation of a stable group structure. However, before these alterations are discussed, the following section will discuss the efficiency of the formation control algorithm at generating the required parallel line formation outline in Chapter 1.

4.5.3 Mapping Efficiency of Formation Control Algorithm

While the previous two sections focussed on analysing the ability of the formation control algorithm to promote the formation of a stable group structure, this section considers the efficiency of the resulting group formation to complete the surveying missions outlined in Chapter 1. As shown below in Figure 4.17 in order to determine whether or not the required parallel line formation has been generated, it is necessary to evaluate the *lateral* distances between each vehicle and its two nearest neighbours.

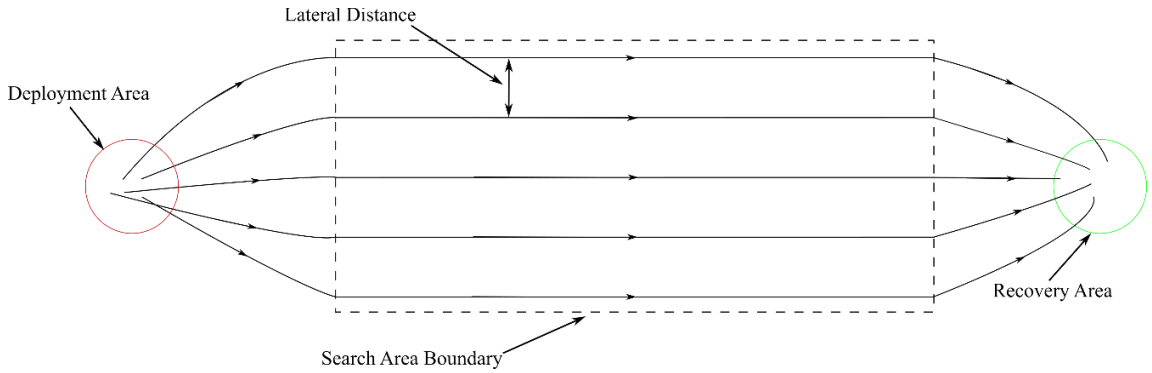


Figure 4.17- Overview of a Multi-Vehicle Oceanic Surveying Mission.

Analysing Figure 4.17, it is apparent that in order to generate the required formation, every vehicle should have a *lateral* distance to their closest nearest neighbour, d_1 that is within the confines of the orientation zone, i.e. $zoo_l \leq d_1 \leq zoo_u$. Thereafter, for the vehicles not positioned at the extreme left and right hand side of the group, i.e. N-2 vehicles, the *lateral* distance to their second nearest neighbour, d_2 should also be within the confines of the orientation zone, i.e. $zoo_l \leq d_2 \leq zoo_u$. For the two vehicles positioned at either side of the group, their d_2 values should be double the normal orientation zone limits, i.e. $2(zoo_l) \leq d_2 \leq 2(zoo_u)$. These convergence criteria are summarised below in Table 4.6.

Table 4.6 - Algorithm convergence criteria.

Parameter	Convergence Criteria
Lateral Distance to Nearest Neighbour – d_1	For N Vehicles: $zoo_l \leq d_1 \leq zoo_u$
Lateral Distance to Second Nearest Neighbour – d_2	For N-2 vehicles: $zoo_l \leq d_2 \leq zoo_u$ For 2 vehicles: $2(zoo_l) \leq d_2 \leq 2(zoo_u)$

In order to determine whether or not the criteria presented in Table 4.6 has been satisfied, shown below in Figure 4.18 is the evolution of the percentage of vehicles satisfying the criteria associated with the parameter d_1. The results presented below have been evaluated across all the simulations completed in this study when I_{NN} was set to 11. This means the percentage value shown below is the percentage of 2400 vehicles.

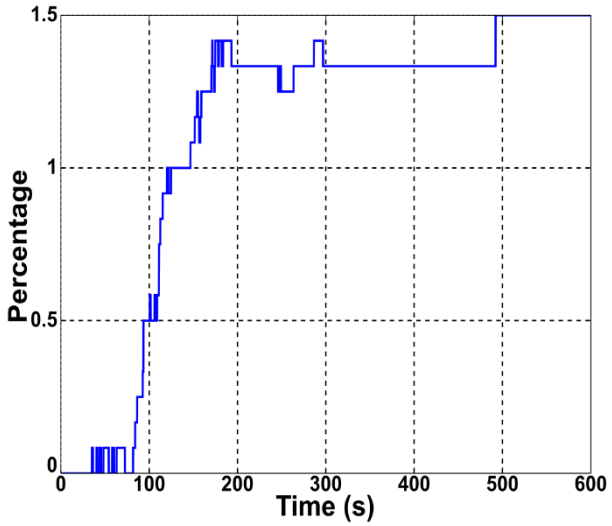


Figure 4.18- Evolution of the percentage of vehicles satisfying the criteria associated with d_1.

The results clearly demonstrate that of the 2400 vehicles simulated, only 1.5% where able to satisfy the convergence criteria associated with the algorithmic parameter, d_1. Based on these figures, it is apparent that very few if any of the simulations undertaken were able to satisfy the necessary criteria to ensure the required parallel line formation was generated. Therefore, as with the results presented in Section 4.5.2, it is apparent that alterations will have to be made to the formation control algorithm in order for it to not only operate more efficiently but also ensure that it will be able to successfully generate the required parallel line formation.

4.6 Summary

The results presented above in Section 4.5 have provided a detailed analysis of the results obtained from the simulations undertaken throughout this study. The aim of the following section is to now provide a summary of the pertinent results obtained from this analysis and also detail the necessary changes that need to be made to the algorithm in order to improve its performance.

In Section 4.5.1, the results demonstrated that regardless of the value used for I_{NN}, the *repulsive* behavioural control law is capable of manoeuvring each vehicle to a nearest neighbour distance that is within the confines of the orientation zone. In Section 4.5.1.1, the results demonstrate that in order for the *orientating* control law to operate as efficiently as possible, the underlying interaction graph should not only be connected but also *fully* connected. In general terms, this means that I_{NN} needs to be equal to N-1 and for the work completed in this thesis, this means that I_{NN} has to be equal to 11. While setting I_{NN} to eleven guarantees the *orientating* control law operates as efficiently as possible, the results presented in Section 4.5.1 also demonstrate that I_{NN} needs to be set to six to ensure the *repulsive* control law operates as efficiently as possible. Furthermore, Section 4.5.2 illustrates that the *attractive* control law is incapable of guaranteeing that the required convergence criteria will be satisfied for any value of I_{NN} Therefore, alterations need to be made to this control law to ensure it operates as expected regardless of the value used for I_{NN}. Finally, Section 4.5.3 demonstrated that the formation control algorithm, in its current guise, is unable to coordinate the vehicles into the required parallel line formation presented in Chapter 1 and as a result, it is unable to efficiently complete the required oceanic surveying operations. It is therefore apparent from the results presented in this chapter that creating a formation control algorithm that simply imitates the behavioural mechanisms of fish partaking in schooling behaviour cannot ensure the formation of a stable group structure. Furthermore, this algorithm has also been shown to be incapable of coordinating the vehicles into the required parallel line formation. To amend this, it is proposed that the following changes be made to the formation control algorithm:

- Instead of each vehicle using the *absolute* distance to its nearest neighbour when determining which of the three control laws to use, the algorithm is altered to use the *lateral* distance instead.
- The *attractive* behavioural control should be altered to ensure the scenarios presented in Sections 4.5.2.1-2 are unable to materialise in the future.

The following chapter will now describe the implementation of these alterations. Furthermore, the same performance metrics used within this chapter will also be used in order to analyse to what extent these alterations have improved the ability of the algorithm to complete the mission profile outlined in Chapter 2.

Chapter 5

Modified Formation Control Algorithm

5.1 Introduction

As the results presented in Chapter 4 demonstrate, the biomimetic formation control algorithm is unable, as required, to coordinate the vehicles into the required parallel line formation. Furthermore, the results also demonstrate that the algorithm cannot guarantee the convergence of every vehicle to within the confines of the orientation zone. The reason for this non-convergence is due to the conflicting requirements placed on the algorithmic parameter, I_{NN} by the *orientating* and *attractive* behavioural control laws.

In order to achieve consensus in the shortest period of time and therefore ensure every vehicle converges to approximately the same heading angle as quickly as possible, the *orientating* behavioural control law requires that the value for I_{NN} be set to the size of the group minus one, i.e. N-1. However, if this value is used within the *attractive* control law, the results presented in Section 4.5.2 demonstrate that there is a 23% chance that the algorithm will be unable to satisfy the necessary convergence criteria. Furthermore, as mentioned above, the results presented in Section 4.6.3 established unequivocally that the algorithm, in its current guise, is also unable to coordinate the vehicles into the required parallel line formation.

The aim of this chapter therefore is to present the modifications made to the algorithm in the anticipation that once incorporated, they will allow the algorithm to coordinate the vehicles into the required formation. In the process, these modifications will also aim to ensure that the *attractive* and *orientating* control laws operate in the desired manner regardless of the value used for I_{NN}. To demonstrate whether or not these modifications have been successful they will be thoroughly tested using the same two groups of simulations used in Chapter 4.

The chapter presents the work described above in the following manner. Section 5.2 describes the modifications made to the algorithm and the reasons for making them. Section 5.3 presents a summary of the simulation setup used in Chapter 4 and now within the work completed in this chapter. Section 5.4 presents and analyse the results obtained from these simulations. Finally, Section 5.5 presents a summary of the work completed in this chapter.

5.2 Formation Control Algorithm – Implementing Modifications

The main recommendation of Chapter 4 was to alter the formation control algorithm so that the *lateral* distance between neighbouring vehicles is used instead of the *absolute* distance when deciding which particular behavioural control law to implement. As discussed in the summary of Chapter 4, the purpose of this modification is to ensure that the required parallel line formation can be generated and hence, the vehicles will be capable of completing the oceanic surveying missions discussed in Chapter 1. Before these modifications are presented though, shown below in Table 5.1 is a summary of the algorithmic parameters used within the modified algorithm presented in this chapter.

Table 5.1 - Description of parameters used in modified algorithm.

Parameter	Description	Value
d_{abs}	Absolute distance to nearest neighbour	Any real value
d_1	Lateral distance to nearest neighbour	Any real value
$\psi_{R_{NN}}$	Relative angular position of vehicles nearest neighbour	$-180^o \leq \psi_{R_{NN}} \leq 180^o$
NN_L	Number of nearest neighbours to the vehicles left	$0 \leq NN_L \leq N-1$
NN_R	Number of nearest neighbours to the vehicles right	$0 \leq NN_R \leq N-1$

While the *lateral* distance, d_1 will be predominantly used to decide which behavioural control law to implement, Table 5.1 demonstrates that the *absolute* distance, d_{abs} is still used

within the modified algorithm. Unsurprisingly, the reason for maintaining this parameter is to ensure that neighbouring vehicles do not collide with one another.

As with the biomimetic formation control, $\psi_{R_{NN}}$ is used to determine the relative position of each vehicles nearest neighbour and the parameters NN_L and NN_R are implemented to determine the number of neighbouring vehicles positioned to each vehicles right and left hand side.

While the methods used to evaluate the parameters, d_{abs}, $\psi_{R_{NN}}$, NN_L and NN_R have already been presented in Chapter 4, Section 5.2.1 below will now describe the functionality implemented in order to evaluate the *lateral* distance, d_{lat} between each vehicle and its nearest neighbours.

5.2.1 Evaluation of Lateral Nearest Neighbour Distances

As shown below in Figure 5.1, in order to evaluate the *lateral* distances between neighbouring vehicles, it is necessary to assume that the vehicles are manoeuvring with parallel trajectories.

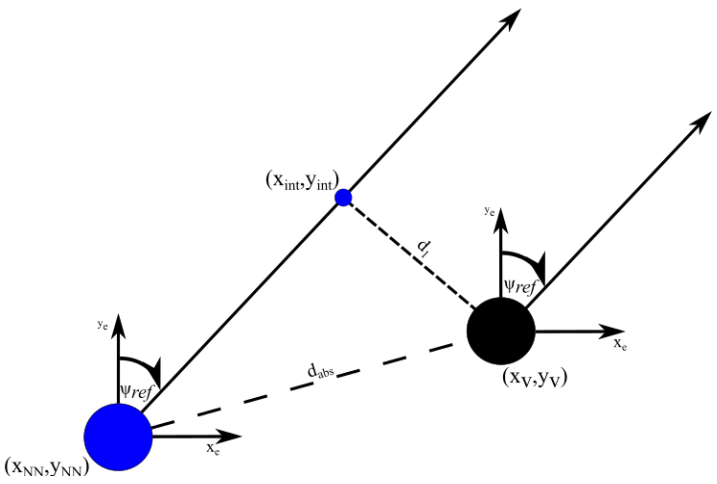


Figure 5.1- Diagram & Notation used to evaluate the lateral distance between neighbouring vehicles.

With the trajectories assumed to be straight lines, they can be represented using Equations (5.1) and (5.2) shown below.

$$y_v = m_v x_v + c_v \tag{5.1}$$

$$y_{NN} = m_{NN} x_{NN} + c_{NN} \tag{5.2}$$

Where x_{NN} and y_{NN} are the coordinates of each vehicle's nearest neighbour(s), x_v and y_v are the coordinates of the vehicle itself and finally, m_v and m_{NN} represent the gradient of the lines calculated using Equation (5.3) below.

$$m_v = m_{NN} = \tan(\psi_{ref}) \tag{5.3}$$

While not defined yet, ψ_{ref} is the angle used within the modified algorithm to enable the entire group to manoeuvre in a purposeful direction. Regardless, the equation of the line perpendicular to these two lines (dotted line labelled d_1 in Figure 5.1) can be defined using Equation (5.4) shown below.

$$y_{perp} = m_{perp}x_{perp} + c_{perp} \tag{5.4}$$

Where m_{perp} is equal to:

$$m_{perp} = \frac{-1}{tan(\psi_{ref})} \tag{5.5}$$

And c_{perp} is equal to:

$$c_{perp} = y_v - m_{perp}x_v \tag{5.6}$$

The coordinates of the intersection, x_{int} and y_{int} can now be evaluated using Equation (5.7) and (5.8) shown below.

$$x_{int} = -\left(\frac{c_{perp} - c_{NN}}{m_{perp} - m_{NN}}\right) \tag{5.7}$$

$$y_{int} = m_{perp}x_{int} + c_{perp} \tag{5.8}$$

Where c_{NN} is calculated using the following equation:

$$c_{NN} = y_{NN} - m_{NN}x_{NN} \tag{5.9}$$

The *lateral* distance between the two vehicles can now be evaluated using Equation (5.10) shown below:

$$d_{lat} = \sqrt{(x_v - x_{int})^2 + (y_v - y_{int})^2} \tag{5.10}$$

As with the evaluation of the *absolute* nearest neighbour distance, the above process is repeated until the *lateral* distance to every vehicle within the group is known.

5.2.2 Identifying Absolute Nearest Neighbour

As discussed above, the modified algorithm presented in this chapter uses the *lateral* distance between neighbouring vehicles when deciding which particular control law to implement. However, in order to ensure neighbouring vehicles do not collide with one another, the modified algorithm also monitors the *absolute* distance between neighbouring vehicles.

As with the original formation control algorithm presented in Chapter 4, it achieves this by using Equation (4.9). Once the distance to each vehicle is known (and stored within the vector, $\mathbf{D}_{abs}$), the algorithm sets the vehicles *absolute* nearest neighbour to be equal to the neighbouring vehicle associated with the minimum value of the vector, $\mathbf{D}_{abs}$.

5.2.3 Identifying Lateral Nearest Neighbour

While each vehicles *absolute* nearest neighbour is identified by evaluating the minimum value of $\mathbf{D}_{abs}$, the methodology implemented to evaluate each vehicles *lateral* nearest neighbour is slightly different and is dependent on not only the *lateral* distance between neighbouring vehicles but also their relative position within the group. For example, if a vehicle has more neighbouring vehicles positioned to its left-hand side, the algorithm identifies the vehicle's *lateral* nearest neighbour as the vehicle positioned immediately to its left-hand side. Similarly, for vehicles with more neighbouring vehicles positioned to their right-hand side, the opposite applies. As shown below in Figure 5.2, implementing this particular identification strategy results in the interaction graph associated with each vehicles *lateral* nearest neighbour being globally connected.

The direction of the arrows presented in Figure 5.2 indicate each vehicle's *lateral* nearest neighbour, i.e. Vehicle 1's nearest neighbour is Vehicle 2 and so on.

Figure 5.2 shows that because Vehicles 1-6 have more vehicles positioned to their left-hand side, their *lateral* nearest neighbour is the vehicle positioned immediately to their left. Conversely, vehicles 7-12 evaluate their *lateral* nearest neighbour to be vehicle positioned immediately to their right.

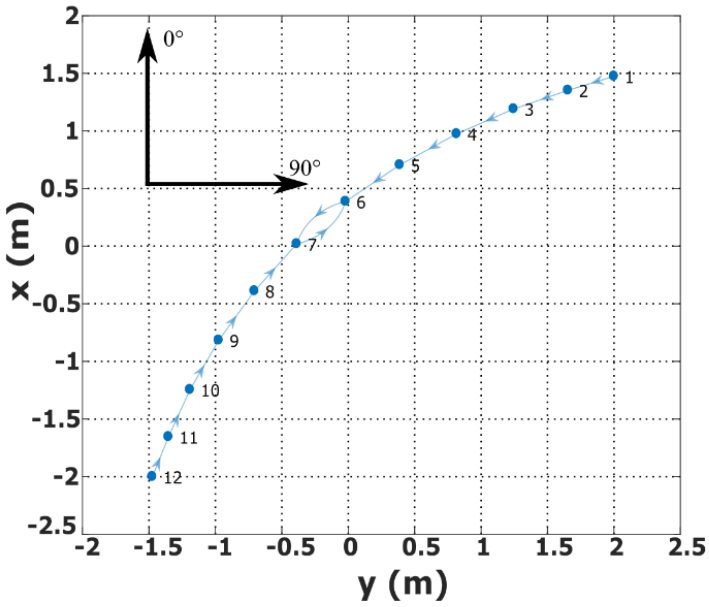


Figure 5.2– Group interaction topology associated with each vehicle's lateral nearest neighbour.

Importantly, Figure 5.2 also demonstrates that the number of vehicles positioned to either side of each vehicle varies depending on the heading angle of the group. For example, if every vehicle in Figure 5.2 were to manoeuvre with a heading angle of 0º, Vehicle 12 would be classified as being positioned to the extreme left of the group and conversely, Vehicle 1 would be considered to be positioned to the extreme right. However, if the group were to manoeuvre with a heading angle of 90º, the opposite would apply. Therefore, it is apparent that the intended heading angle of the group has to be taken into consideration when deciding how many vehicles are positioned to the right or left-hand side of the group.

5.2.3.1 Evaluating Relative Positioning Of Nearest Neighbours

As discussed in Section 4.3.2, in order to evaluate the relative positioning of each vehicle's nearest neighbour(s), the following equation has to be used:

$$\psi_{R_{NN}} = \psi_{LOS} - \psi_V \tag{5.11}$$

Where, as shown below in Figure 5.3, ψ_{LOS} represents the line-of-sight angle calculated using Equation (5.12) and ψ_V is the current heading angle of the vehicle itself.

$$tan(\psi_{LOS}) = \left(\frac{y_{NN} - y_v}{x_{NN} - x_v}\right) \tag{5.12}$$

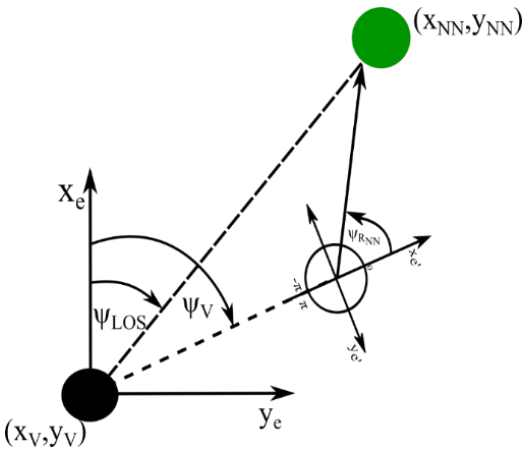


Figure 5.3 - Diagram demonstrating the effect that the vehicles heading angle has on determining the relative positioning of nearest neighbours.

Where, x_{NN} and y_{NN} represent the x and y coordinates of the vehicle's nearest neighbour and x_v and y_v represent the position of the vehicle itself.

However, for the modified algorithm, instead of using ψ_V , Equation (5.11) has been altered to take into consideration the aforementioned reference heading angle of the group, ψ_{ref}. This reference heading angle is calculated using Equation (5.13) below.

$$\psi_v = \psi_{ref} = tan^{-1}\left(\frac{y_{des} - \bar{x}_{NN}}{x_{des} - \bar{y}_{NN}}\right) \tag{5.13}$$

Where the parameters x_{des} and y_{des} represent the coordinates of the group's desired location (user defined goal destination) and $\bar{x}_{NN}$ and $\bar{y}_{NN}$ represent the coordinates of the groups average position calculated using the following equations:

$$\bar{x}_{NN} = \frac{1}{I_{NN}} \sum_{N=1}^{I_{NN}} x_{NN} \tag{5.14}$$

$$\bar{y}_{NN} = \frac{1}{I_{NN}} \sum_{NN=1}^{I_{NN}} y_{NN} \tag{5.15}$$

As discussed extensively in Chapter 4, in order for the vehicles to achieve consensus as quickly as possible, each vehicle is required to take into consideration every other vehicle within the group, i.e. I_{NN} = N-1.

Regardless, once ψ_V has been replaced with ψ_{ref} it will now be possible for each vehicle to accurately evaluate the number of vehicles positioned to its left or right-hand side relative

to the reference heading angle of the group. This is achieved by using the following conditional statement:

$$NN_L = NN_L + 1 \quad ,if \quad -180^{\circ} < \psi_{R_{NN}} < 0^{\circ} \tag{5.16}$$

$$NN_R = NN_R + 1 \quad ,if \quad 0^{\circ} < \psi_{R_{NN}} < 180^{\circ} \tag{5.17}$$

$$NN_L = NN_L + 1 \quad ,if\ \psi_{R_{NN}} = 0^{\circ} \tag{5.18}$$

$$NN_R = NN_R + 1 \quad ,if\ abs(\psi_{R_{NN}}) = 180^{\circ} \tag{5.19}$$

Crucially, the conditional statements presented above in Equation (5.16)-(5.19) guarantee that regardless of the relative positioning of neighbouring vehicles, the modified algorithm selects each vehicle's *lateral* nearest neighbour such that the interaction topology shown in Figure 5.2 is generated.

Once each vehicle's *lateral* and *absolute* nearest neighbour distances have been identified, the next phase of the algorithm is to use these values to determine which behavioural control law to implement. This decision making process and how it alters from the one shown in Chapter 4 are presented in the following section.

5.2.4 Decision Making Process

Shown below in Figure 5.4 is the decision making process incorporated within the modified formation control algorithm to ascertain which particular behavioural control law should be implemented.

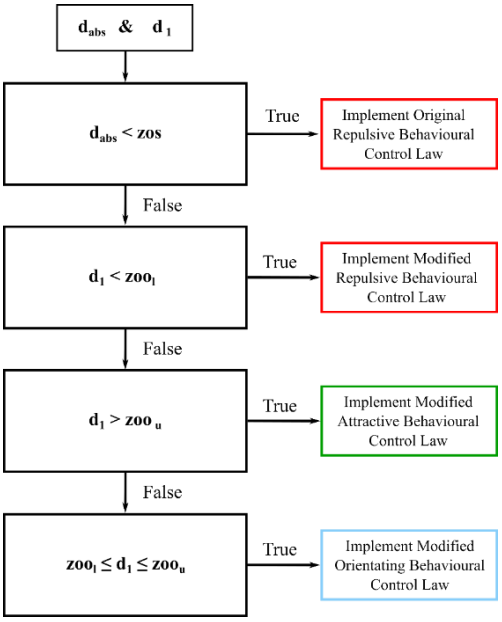


Figure 5.4 - Decision making structure used within the modified formation control algorithm.

As the flow diagram demonstrates, the modified formation control algorithm contains four conditional statements – one more than the algorithm presented in Chapter 4. The reason for including this additional conditional statement is to ensure that as well as coordinating the vehicles into the required parallel line formation, the modified algorithm is also able to ensure that neighbouring vehicles do not collide with one another.

Figure 5.4 demonstrates that the first conditional statement determines whether or not the value for d_{abs} is below a certain limit, zos. If it is, the vehicles are deemed to be too close together and the original *repulsive* behavioural control law is implemented. As the results presented in Chapter 4 demonstrate, this ensures neighbouring vehicles manoeuvre away from one another and the value for d_{abs} increases above the limit, zos. Thereafter, the flow diagram demonstrates that the modified formation control algorithm implements either the modified *repulsive*, *attractive* or *orientating* control law depending on whether the value for d_1 is below, above or within the confines of the orientation zone. The following three sections will now describe the modifications made to these three behavioural control laws.

5.2.5 Modified Repulsive Behavioural Control Law

As discussed above, the aim of the modified *repulsive* control law is to ensure that every vehicle converges to a d_1 value that is within the confines of the orientation zone. Subsequently, to achieve this, the following conditional statements have been incorporated:

$$\psi_D = \begin{cases} \psi_{ref} + \psi_{rep} & if\ NN_L \geq NN_R \\ \psi_{ref} - \psi_{rep} & if\ NN_L < NN_R \end{cases} \tag{5.20}$$

In the above equation, ψ_{rep} represents a user defined algorithmic parameter which can be used to alter how quickly neighbouring vehicles manoeuvre away from one another. As the aim of Equation (5.20) is to increase the *lateral* distance between neighbouring vehicles as quickly as possible this parameter has been set to 90º. As discussed below, when combined with the interaction topology shown in Figure 5.2, setting ψ_{rep} to 90º should result in the vehicles generating the required parallel line formation in the shortest possible time.

In order to understand why this is the case, it is necessary to consider the interaction graph shown in Figure 5.2. In doing so, it becomes apparent that when the conditional statements presented above are implemented, Vehicles 6 & 7 will manoeuvre in the opposite direction to one another. As a result, the *lateral* distance between these two vehicles will increase at the fastest possible rate until it is within the confines of the orientation zone. At which point, these two vehicles will manoeuvre with a heading angle equal to the aforementioned reference heading angle, ψ_{ref}.

Thereafter, Figure 5.2 demonstrates that because Vehicles 6 and 7 are now moving with a heading angle equal to ψ_{ref} , the *lateral* distance between Vehicles 5 & 6 and 8 & 7 will increase at the fastest possible rate until they too are within the confines of the orientation zone. This process of sequential convergence will continue until the *lateral* distances between Vehicles 12 & 11 and 1 & 2 are also within the confines of the orientation zone.

5.2.6 Modified Orientating Behavioural Control Law

As mentioned previously, the reference heading angle, ψ_{ref} has been included to ensure the group of vehicles are capable of manoeuvring to any desired location. Intuitively therefore, it is proposed that vehicles using the *orientating* behavioural control law should manoeuvre with a heading angle that is equal to this value. As discussed in Section 5.2.3.1, the *reference* heading angle is calculated using Equations (5.13)-(5.15).

5.2.7 Modified Attractive Behavioural Control Law

As Figure 5.4 demonstrates, if the value for d_1 is greater than the upper boundary of the orientation zone, the formation control algorithm will implement the modified *attractive* behavioural control law. As shown below in Equation (5.21), this modified control law also contains two conditional statements:

$$\psi_D = \begin{cases} \psi_{ref} - \psi_{att} & ,if\ NN_L \geq NN_R \\ \psi_{ref} + \psi_{att} & ,if\ NN_L < NN_R \end{cases} \tag{5.21}$$

As with Equation (5.20), the parameter, ψ_{att} is a user defined parameter which similarly to the modified *repulsive* control law determines how quickly neighbouring vehicles manoeuvre towards one another. However, unlike Equation (5.20), its value has been set to 45º. The reason for selecting this particular value is that it ensures neighbouring vehicles manoeuvre towards one another as quickly as possible while at the same time, ensuring the entire group continues to manoeuvre towards the group's desired location.

Nevertheless, Equation (5.21) demonstrates that when the modified *attractive* control law is implemented, vehicles positioned at opposite sides of the group will manoeuvre with converging trajectories until the value for d_1 (for neighbouring vehicles) is within the confines of the orientation zone. Again, because of the particular interaction topology implemented, the vehicles positioned at the centre of the group (Vehicles 6 & 7 in Figure 5.2) will converge first. Thereafter, the sequential convergence of neighbouring vehicles as discussed in Section 5.2.5 will occur until every vehicle is positioned as required.

5.2.8 Velocity Control Law

While it was necessary to alter the heading control laws of the formation control algorithm, it was decided that the velocity control laws presented in Section 4.4.4 would remain unchanged. This means that in the modified formation control algorithm, the heading control laws ensure each vehicle converges to the required *lateral* nearest neighbour distance while the velocity control law ensures each vehicle converges to the required *absolute* nearest neighbour distance.

5.2.9 Modified Algorithm Structure

As shown below in Figure 5.5, the only change made to the original algorithmic structure shown in Figure 4.6 is the evaluation of the *lateral* distance, $\mathbf{d}_{lat}$ between each vehicle and its various nearest neighbours as described in Section 5.2.1.

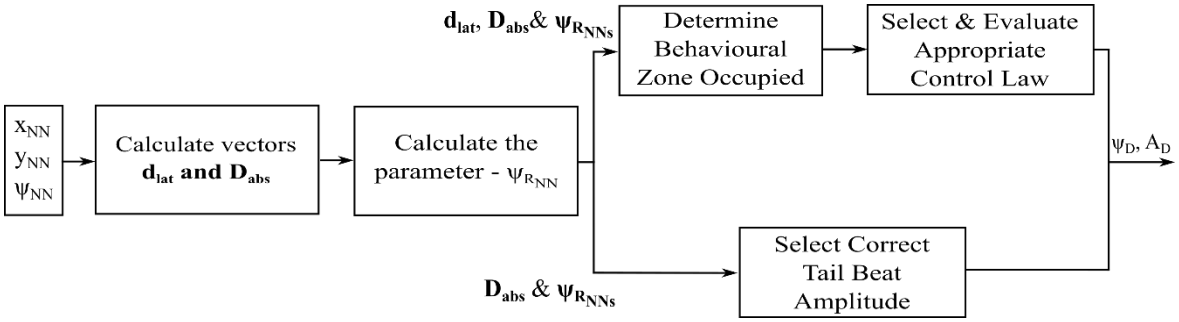


Figure 5.5 - Modified algorithm structure.

As Figure 5.5 illustrates, the input to the formation control algorithm has remained unchanged and is still the positional and heading data communicated by each vehicles nearest neighbours, i.e. x_{NN}, y_{NN} and ψ_{NN}. This data is then used to populate the vectors, $\mathbf{d}_{lat}$ and $\mathbf{D}_{abs}$. As discussed previously, these vectors contain the *lateral* and *absolute* distances between the vehicle and its nearest neighbours as calculated using Equations (4.9) and (5.10). After these distances are evaluated, the relative positioning of each vehicle is calculated and stored within the vector, $\boldsymbol{\psi}_{\mathbf{R_{NN}}}$. Thereafter, the vector, $\mathbf{d}_{lat}$ is used in conjunction with the decision making process of Figure 5.4 to determine which behavioural control law should be implemented. Similarly, the vectors $\mathbf{D}_{abs}$ and $\boldsymbol{\psi}_{\mathbf{R_{NN}}}$ are also used to determine whether the vehicle needs to increase, decrease or maintain its current velocity based on the velocity control law presented in Section 4.3.4. Once these processes have been completed, the formation control algorithm produces the desired heading angle of the vehicle, ψ_D and the necessary *Tail Beat Amplitude*, A_D that will manoeuvre the vehicle in the required direction.

5.3 Simulation Setup & Performance Metrics

The main aim of this chapter is to demonstrate that the modifications made to the algorithm operate as envisioned and as a result, it is now capable of successfully coordinating the vehicles into the required parallel line formation. In order to definitively demonstrate whether or not this is the case, it is important that the simulations and performance metrics used are similar to those in Chapter 4. Therefore, the following two sections provide a brief review of these and highlight any differences or improvements identified within the subsequent analysis.

5.3.1 Simulation Setup

Table 5.2 demonstrates that similarly to Chapter 4 the simulations undertaken in this chapter are separated into two groups. Once again, these groups are categorised depending on the initial control law used by the vehicles. At this point, it is important to note that although the *lateral* distances are now predominantly used to determine which behavioural control law to implement, the vast majority of the vehicles will still use either the *repulsive* or *attractive* laws at the start of each simulation.

Table 5.2 - Simulation Parameters.

Parameter	Group 1	Group 2
Initial Behaviour Utilised	**(Repulsive)**	**(Attractive)**
Behavioural Zones Size (m)	[20,25,200]	[6,11,200]
Nearest Neighbours Used (I_{NN})	11	11
Reference Heading Angle (ψ_{ref})	$\frac{\psi}{4}$	$\frac{\psi}{4}$

However, unlike Chapter 4, Table 5.2 demonstrates that the simulations completed in this chapter do not vary the parameter, I_{NN} and instead, its value is kept constant at 11. This is due to the fact that as shown in Figure 5.2, each vehicle needs to know its relative position within the group and as a result, needs to know the position of every other vehicle in the group.

Furthermore, the other variation between the simulations completed in this chapter and the previous one, is the inclusion of the parameter, ψ_{ref}. As Table 5.2 illustrates, a constant

value of 45º has been selected. The decision not to vary this parameter is to ensure that variations in its value cannot be attributed to the algorithm not operating as expected.

Finally, as with Chapter 4, for each of the two groups presented in Table 5.2, 100 separate simulations have been completed. Due to the fact that I_{NN} is no longer altered, the results presented below are compiled from 200 simulations.

5.3.2 Performance Metrics

In order to allow a direct comparison with the results obtained in the previous chapter, the performance metrics used in this chapter are the same as those presented in Chapter 4. As a result, Table 5.3 below provides a summary of these metrics and the criteria that must be satisfied in order to demonstrate that the algorithm has coordinated the vehicles into the required parallel line formation.

Table 5.3 - Description of performance metrics & convergence criteria.

Parameter	Description	Convergence Criteria
Standard Deviation of Vehicle Heading Angle - σ	Allows the variation in vehicle heading angle across each simulation (group) to be analysed.	Value for σ should converge to a value less than or equal to 1º
Lateral Distance to Nearest Neighbour – d_1	Determines whether or not the heading control laws have successfully coordinated each vehicle to the required *lateral* separation distance.	$zoo_l \leq d_1 \leq zoo_u$ for all vehicles.
Lateral Distance to Second Nearest Neighbour – d_2	Determines whether or not the heading control laws have successfully coordinated each vehicle to the required *lateral* separation distance for its second nearest neighbour.	For N-2 vehicles: $zool \leq d_2 \leq zoou$ For 2 Vehicles: $2(zool) \leq d_2 \leq 2(zoou)$
Absolute Nearest Neighbour Distance - d_{abs}	Provides a metric by which to determine if the velocity controller is operating as expected.	$zoo_l \leq d_{abs} \leq zoo_u$ for all vehicles.

As with the results presented in Chapter 4, the purpose of analysing the standard deviation of the vehicles heading angle, σ is to determine whether or not the algorithm has been successful in ensuring every vehicle (from within each simulation) converges to the same heading angle. Thereafter, as discussed in Section 4.5.3, the purpose of evaluating and analysing the *lateral* distance between each vehicle and its two nearest neighbours, i.e. d_1 and d_2 is to determine whether or not the vehicles have been able to coordinate themselves into the required parallel line formation. Finally, analysing the *absolute* distance between each vehicle and its nearest neighbour allows a determination of whether or not the velocity control law still operates as expected and as such, allows each vehicle to manoeuvre directly alongside their nearest neighbour.

If the first three criteria presented in Table 5.3 are satisfied, the modifications made to the algorithm can be deemed successful. This is due to the fact that if these three criteria are satisfied, not only does it guarantee that every vehicle has converged to the same heading angle but more importantly, it also demonstrate that the vehicles are moving in the required parallel line formation. Thereafter, if the final convergence criteria associated with the parameter, d_{abs} is satisfied, it demonstrates that the velocity control law still operates as envisioned and the neighbouring vehicles are positioned directly alongside one another.

5.4 Results

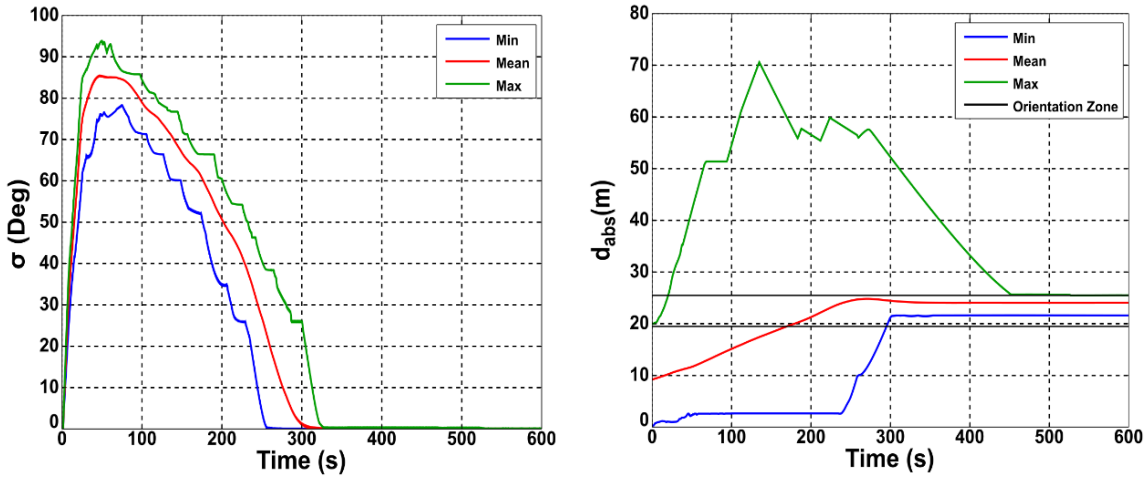
The aim of this section is to analyse the results obtained from the simulations using the modified algorithm presented in Section 5.2. The analysis of these results is to ascertain whether or not the modifications made have improved the algorithms ability to coordinate the vehicles into the required parallel line formation. To achieve this, Sections 5.4.1 and 5.4.2 will present the results obtained from the two groups of simulations presented and summarised above in Table 5.2.

5.4.1 Analysing the Modified Repulsive & Orientating Behavioural Control Laws.

As with the results presented in Chapter 4, the main method to determine whether or not the algorithm has promoted the formation of a stable group structure is to analyse not only the evolution of the standard deviation of the vehicle's heading angle, σ but also the *absolute* nearest neighbour distance, d_{abs}. Subsequently, shown below in Figure 5.6(a) and (b) is the evolution of these two parameters in terms of the minimum, mean and maximum values obtained throughout the simulations completed. The reason for including the minimum and maximum values is to present the entire envelope of results obtained throughout the

simulations completed in order to quickly ascertain whether or not the changes made to the algorithm have been successful.

The results presented below in Figures 5.4 (a) are as expected with the standard deviation value initially increasing as the vehicles implement the modified *repulsive* behavioural control law (Equation (5.20)). After initially increasing, Figure 5.6(a) demonstrates that the minimum and maximum values for σ decrease in incremental steps from approximately 70s onwards. Once again, this behaviour is expected and is due to the fixed interaction topology shown in Figure 5.2 resulting in neighbouring vehicles sequentially converging to a d_1 value within the confines of the orientation zone.



(a) (b)

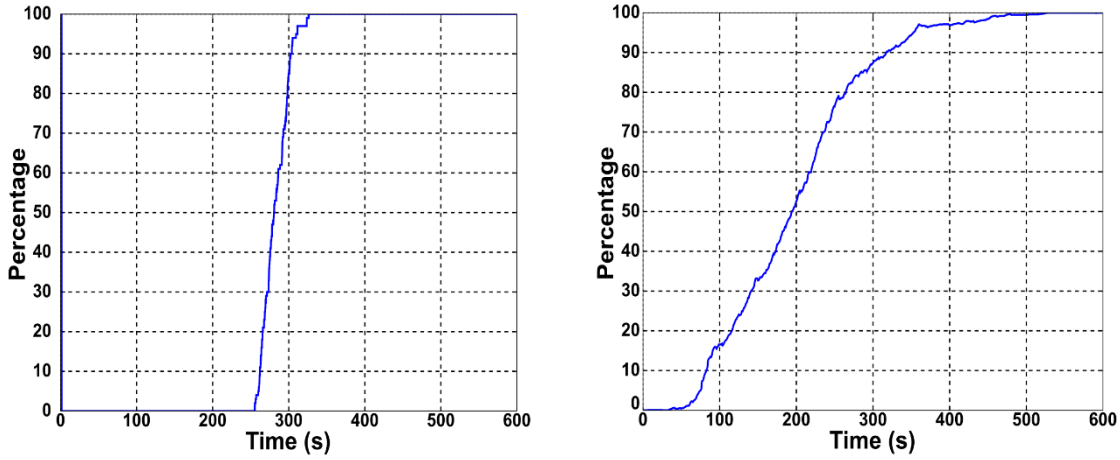
Figure 5.6- Evolution of (a) σ and (b) maximum, minimum & mean values for d_{abs}

Based solely on the results presented in Figure 5.6 (a), it is apparent that the modifications made to algorithm have not affected its ability to ensure that every vehicle (from within each group) converges to manoeuvring with the same heading angle. These results therefore demonstrate that the modified algorithm has produced a group structure that satisfies the first convergence criteria of Table 5.3.

As with Figure 5.6 (a), the results presented in Figure 5.6 (b) are as anticipated with the minimum, mean and maximum values for d_{abs} all converging to within the confines of the orientation zone. More importantly, Figure 5.6 (b) demonstrates that the minimum value for this parameter never decreases throughout the simulations. This demonstrates that the inclusion of the original *repulsive* behavioural control law to ensure neighbouring vehicles do not collide with one another has been successful. Similarly to the results presented in Figure 5.6(a), Figure 5.6(b) suggests that the modified algorithm has been able to coordinate the vehicles into a formation that satisfies the fourth convergence criteria of Table 5.3. This

is shown below in Figure 5.7 where the evolution of the percentage of simulations satisfying the first and last convergence criteria of Table 5.3 are presented.

While the results presented below demonstrate that both convergence criteria have been satisfied, they also illustrate that it takes significantly longer to achieve 100% convergence for the parameter, d_{abs}. Once again, this is an indication that the algorithm is operating correctly as it suggests that only after every vehicle has converged to an appropriate d_1 value, will the velocity control law begin to ensure every vehicle converges to a suitable d_{abs} value.



(a) (b)

Figure 5.7 - Evolution of the percentage of simulations satisfying criteria associated with (a) σ and (b) d_{abs}.

In order to prove that the required parallel line formation has indeed been generated, it is necessary to analyse the evolution of the parameters, d_1 and d_2, i.e. the *lateral* distances between each vehicle and its two nearest neighbours.

This is shown below in Figure 5.8 where similarly to the results presented above, the evolution of the minimum, mean and maximum values for these distances are presented. Furthermore, also shown in Figure 5.8 is the percentage of vehicles satisfying the relevant criteria associated with these distances as described in Table 5.3 above.

As anticipated, Figures 5.8 (a) and (c) clearly demonstrate that the modified algorithm has enabled *every* vehicle from within *each* simulation to converge to a d_1 value that is within the confines of the orientation zone.

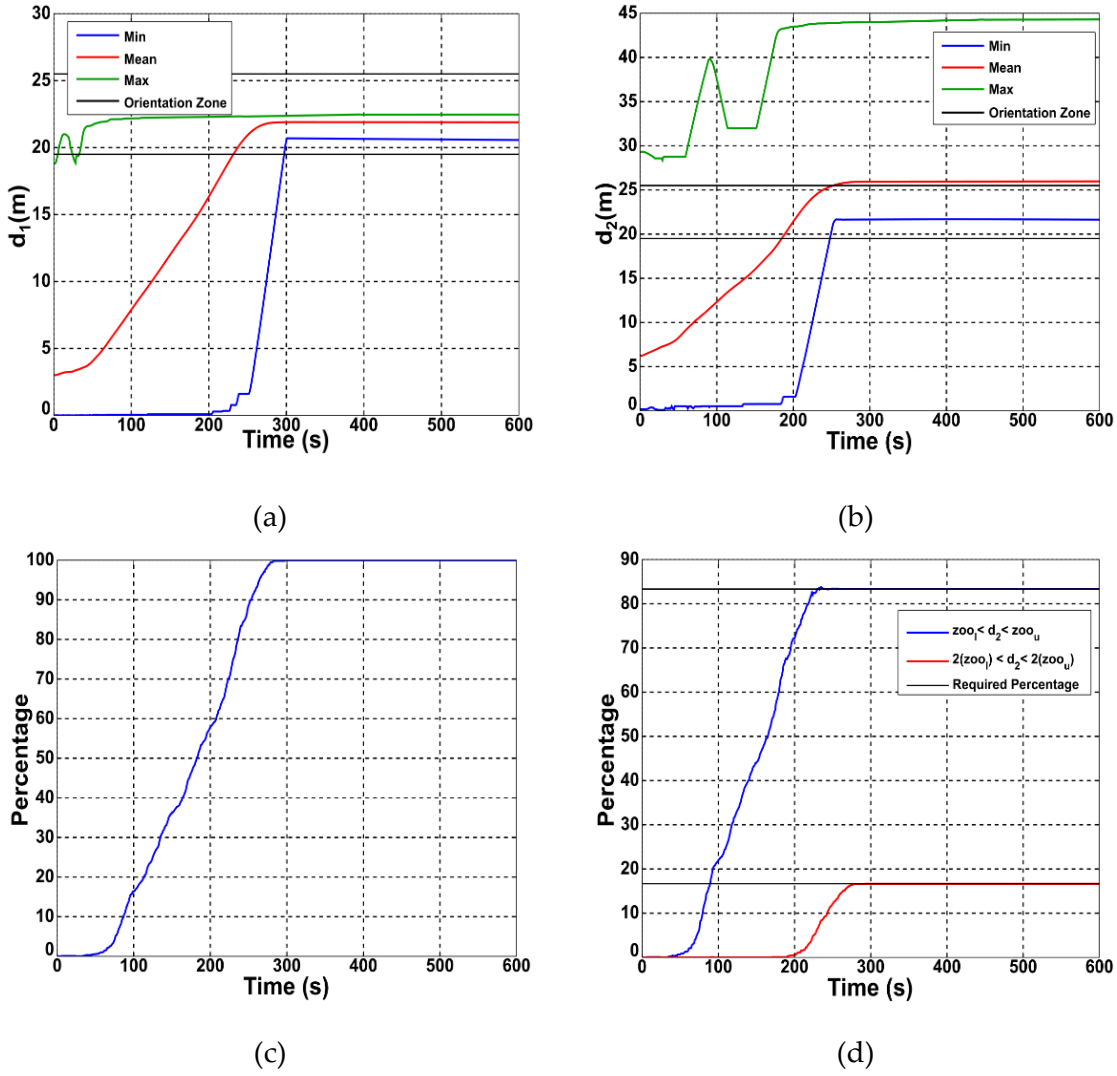


Figure 5.8- Evolution of (a) the minimum, mean and maximum values for d_1 (b) the minimum, mean and maximum values for d_2 (c) percentage of vehicles satisfying convergence criteria associated with d_1 (d) percentage of vehicles satisying criteria associated with d_2.

Conversely, the results shown in Figure 5.8(b) suggest that because the maximum and mean values for d_2 are above the confines of the orientation zone, the algorithm has been unable to coordinate the vehicles into the required parallel line formation. However, upon further analysis, this result was found to be misleading and caused by the vehicles positioned at either side of each group (whose d_2 value by default, will be at least 40m) skewing the results for the entire group. This is proven to be the case when the percentage of vehicles satisfying the criteria associated with the parameter, d_2 are analysed. The percentages presented in Figure 5.8 (d) clearly demonstrate that throughout the 100 simulations completed, 17% of vehicles (i.e. two vehicles per simulation) converged to a d_2 value within the required boundaries and similarly, 83% of vehicles converged to a d_2 value within the confines of the orientation zone, also as required.

These percentages combined with the fact that *every* vehicle has converged to a d_1 value within the orientation zone (Figure 5.8(a) and (c)) demonstrates that the modified formation control algorithm is capable of satisfying the necessary convergent criteria presented in Table 5.3. Furthermore, these results when combined with the fact that it has already been shown that every vehicle converges to the same heading angle (Figure 5.6 (a)) further illustrates that the modifications made to the algorithm have been successful. While not feasible to present the trajectories obtained from every simulation completed, shown below in Figure 5.9 are a selection of the trajectories obtained from a number of the simulations completed.

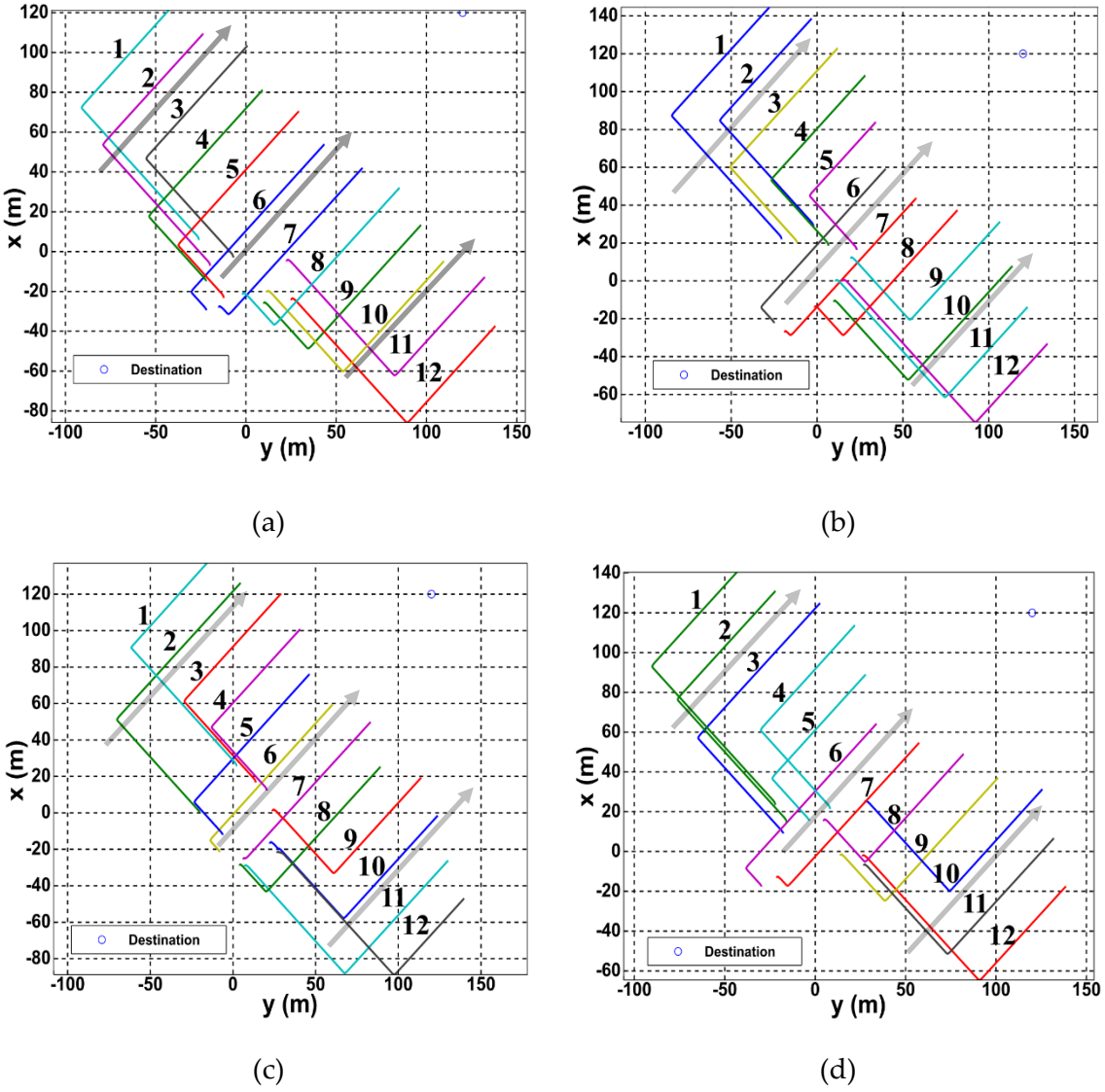


Figure 5.9- Example of vehicle trajectories obtained from modified formation control algorithm when every vehicle is initially using the repulsive behavioural control law.

Analysing the trajectories presented above in Figure 5.9, it is apparent that the modified *repulsive* behavioural control law is operating as designed with vehicles positioned at

opposite ends of the group moving in opposing directions in order to increase the *lateral* distance between themselves and their neighbouring vehicle as quickly as possible. The trajectories also demonstrate that Vehicles 6 and 7 are always, as expected, the first to implement the *orientating* behavioural control law. Thereafter, the trajectories demonstrate that the remaining vehicles all sequentially converge towards implementing the *orientating* control law. Once again, this behaviour is as envisioned and is due to the aforementioned interaction graph shown in Figure 5.2. Finally, the trajectories presented above in Figure 5.9 (a) – (d) demonstrate that once every vehicle is implementing the *orientating* control law, the required parallel line formation has been generated. Therefore, these results when combined with those presented in Figures 5.6-8 demonstrate that the modifications made to the *repulsive* and *orientating* control laws have been successful.

In order to demonstrate that the modified algorithm operates successfully regardless of the initial control law implemented, the following section will now present the results obtained from the second group of simulations

5.4.2 Analysing the Modified Attractive Behavioural Control Laws.

As discussed above, the aim of this section is to analyse the results obtained from the simulation where the majority of vehicles initially implemented the *attractive* behavioural control law. To achieve this, as in the previous section, shown below in Figure 5.10 is the evolution of the minimum, mean and maximum values for the parameters, σ, and d_{abs} and the associated evolution of the percentage of vehicles satisfying the necessary criteria associated with these parameters.

Once again, the results presented in Figure 5.10(a) are as expected, with the standard deviation increasing as the majority of the vehicles implement the *attractive* control law before decreasing as the number of vehicles using the *orientating* control law begins to increase. However, what it is notable from the results presented in Figure 5.10(a) is the substantially longer period of time required for the maximum value of σ to converge towards zero. This is not entirely unexpected and is due primarily to the fact that the algorithmic parameter, ψ_{att} is half the size of the parameter, ψ_{rep} used in the modified *repulsive* control law. As a result, the speed at which the *lateral* distance between neighbouring vehicles can be reduced at, is inherently smaller.

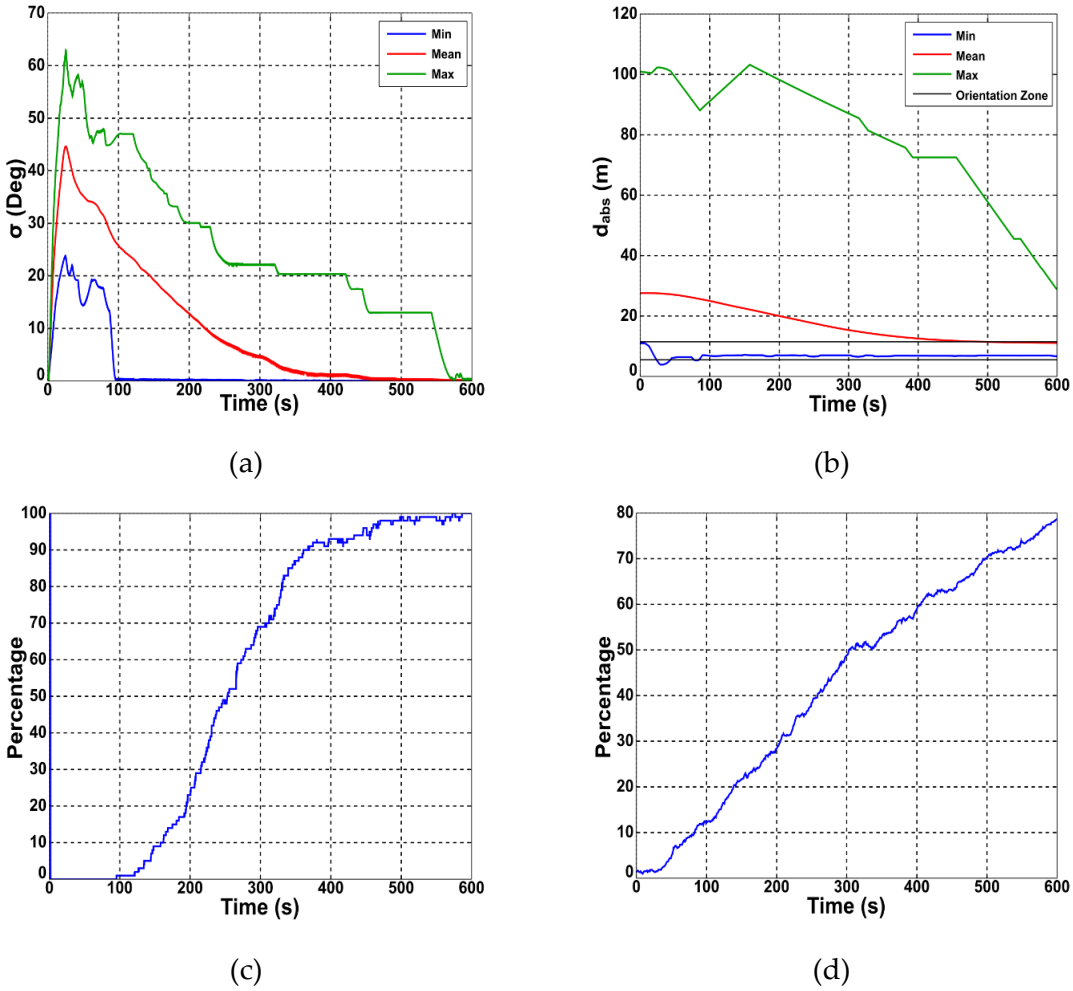


Figure 5.10- Evolution of (a) σ, (b) d_{abs}, (c) percentage of vehicles satisfying the criteria associated with σ and (d) percentage of vehicles satisfying the criteria associated with d_{abs}.

Nevertheless, Figures 5.10 (a) and (c) demonstrate that the modified formation control algorithm has, as required, ensured *every* vehicle converges to approximately the same heading angle and hence the first criteria of Table 5.3 has been satisfied by every vehicle. As with the results presented in the previous section, this suggests that both the *attractive* and *orientating* control laws are operating as expected.

While Figure 5.10 (b) demonstrates that both the minimum and mean values for the parameter, d_{abs} have converged to a value within the confines of the orientation zone, the results clearly demonstrate that the maximum value is still significantly greater than the upper boundary of the orientation zone. However, because the value is steadily decreasing, the results imply that the velocity control is operating as expected and if the simulation run time were to be extended, this parameter would also converge as required. This is further illustrated by the results presented in Figure 5.10 (d) where it is quite clear that the percentage of vehicles satisfying the associated convergence criteria is steadily increasing at the end of the simulations.

When combined, the results presented in Figure 5.10 suggest that because the minimum, mean and maximum standard deviation values have all converged to below the required limit, the modified algorithm has been able to coordinate the vehicles into the required parallel line formation. However, because of the longer convergence time, the velocity control law, while still operating as designed, has been unable to reduce the maximum value of d_{abs} to within the confines of the orientation zone within the given timeframe.

To definitively demonstrate that this is the case, shown below in Figure 5.11 is the evolution of the minimum, mean and maximum values obtained for the *lateral* distances, d_1 and d_2 as well as the evolution of the percentage of vehicles satisfying the associated convergence criteria.

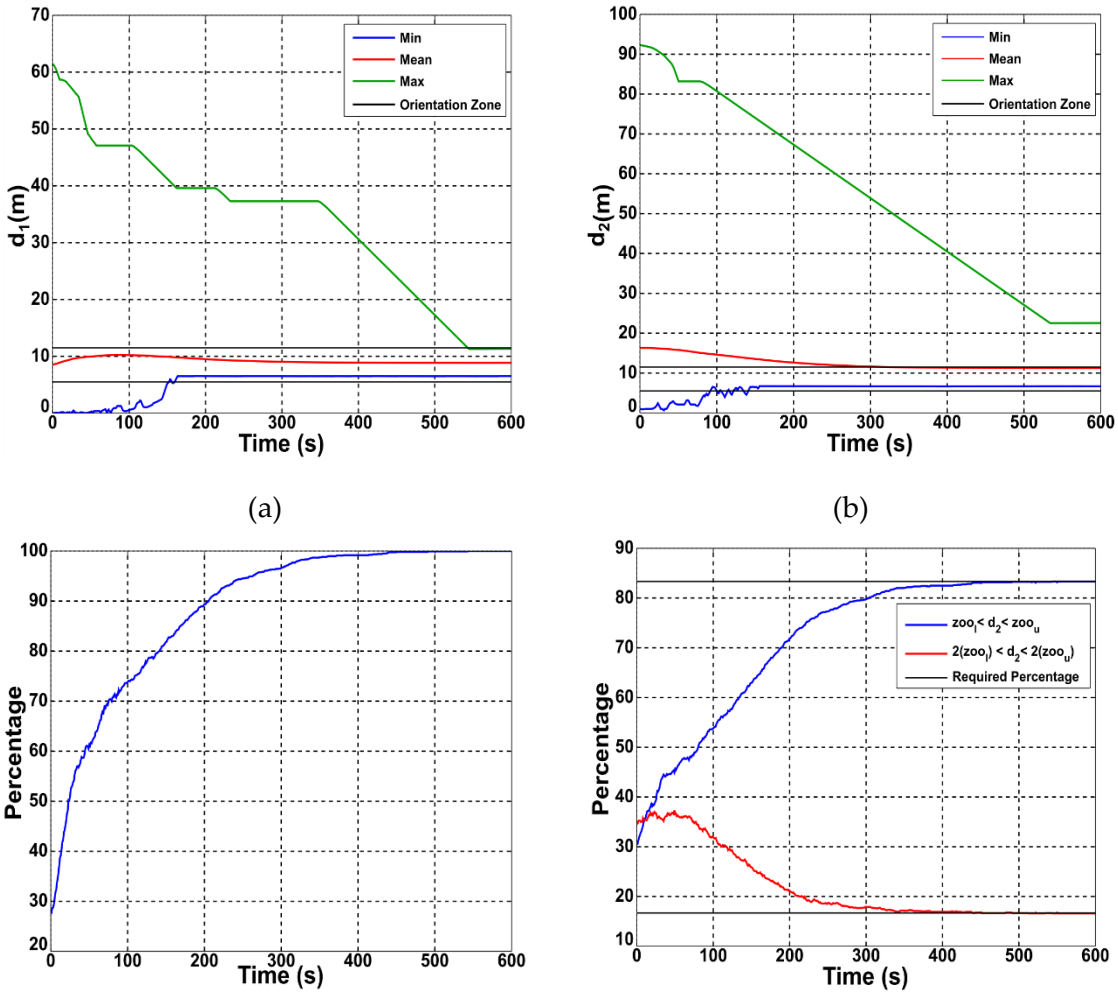


Figure 5.11- Evolution of (a) the minimum, mean and maximum values for d_1 (b) the minimum, mean and maximum values for d_2 (c) percentage of vehicles satisfying convergence criteria associated with d_1 (d) percentage of vehicles satisying criteria associated with d_2 when initially using the *attractive* control law.

Once again, the results presented in Figure 5.11 (a) illustrate that the minimum, mean and maximum values for d_1 have all converged to within the confines of the orientation zone and hence, as shown in Figure 5.11 (c), every vehicle has satisfied the second criteria presented in Table 5.3. Similarly, the results presented in Figure 5.11 (d) demonstrate that the necessary percentage of vehicles have satisfied the various criteria associated with the algorithmic parameter, d_2. Combining these results with those presented above in Figure 5.10 (a) and (c), it is apparent that the first three convergence criteria of Table 5.3 have now been satisfied and as a result, every simulation resulted in the required parallel line formation been generated. In summarising, the results presented above in Figures 5.10-11 demonstrate that the modifications made to the *attractive* behavioural control operate as envisioned. Furthermore, the results also demonstrate that the modifications made to the attractive behavioural control law (Section 5.2.7) ensure the non-convergence behaviour shown in Chapter 4 (Section 4.6.2) no longer occurs. Finally, as with the results presented in the previous section, shown below in Figure 5.12 are a selection of the group trajectories obtained from the simulations undertaken.

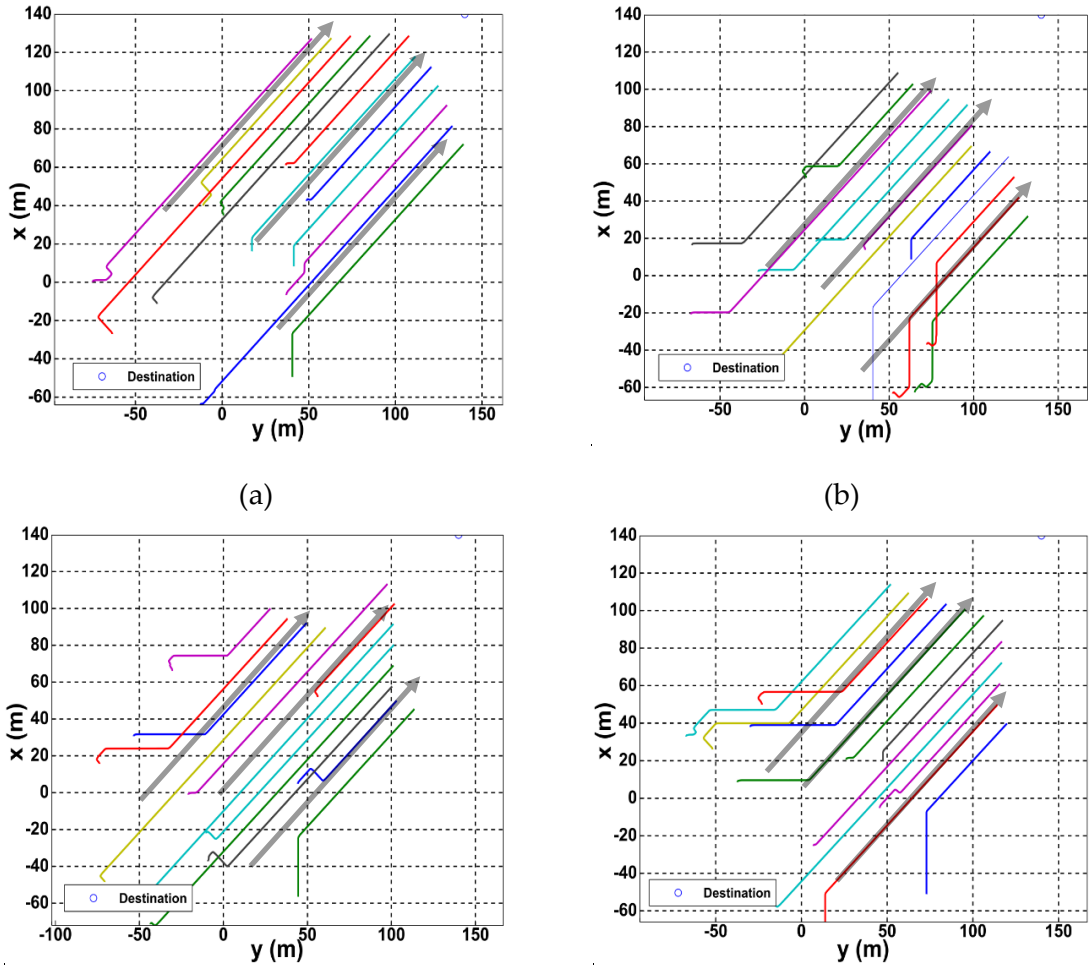


Figure 5.12- Example trajectories.

As designed, the trajectories presented above demonstrate that when the modified *attractive* control law is implemented, the vehicles positioned at either side of the group move with converging trajectories until the value for d_1 converges to within the confines of the orientation zone. Similarly to the results presented in Section 5.4.1, the trajectories presented above demonstrate that it is the middle two vehicles that converge first and thereafter, the remaining vehicles all sequentially converge to using the *orientating* control law. Finally, while the results presented in Figure 5.12 (a) – (d) demonstrate that the required parallel line formation has been generated, Figure 5.12 (c) illustrates the difference in trajectory obtained when the final convergence criteria of Table 5.3 cannot be satisfied. While the difference is noticeable, the trajectory illustrates that because the parallel line formation has still been generated, being unable to satisfy this final convergence criteria has no impact on the group's ability to complete the oceanic surveying missions outline in Chapter 1.

5.5 Summary

The results presented in Section 5.4 demonstrate unequivocally that the modifications made to the algorithm operate as envisioned and as a result, the algorithm, as required, is now capable of coordinating the vehicles (regardless of initial position) into the parallel line formation.

The aim of the following section is to summarise the modifications made to the algorithm and the reasons why they have resulted in the improved performance shown above.

- Section 5.2.3 demonstrates that in order to identify each vehicles *lateral* nearest neighbour, the vehicles are separated into two categories: those with more neighbouring vehicles positioned to their left and those with more neighbouring vehicles positioned to their right. Depending on which category each vehicle is within, it will identify its *lateral* nearest neighbour as the vehicle positioned immediately to its right or left-hand side.
- Importantly, this identification process always results in the interaction topology shown in Figure 5.2 always being generated among the vehicles.
- Sections 5.2.5 and 5.2.7 demonstrate that the modifications made to the *repulsive* and *attractive* control laws result in vehicles positioned at either side of the entire group moving with either converging or diverging trajectories.
- Crucially, these aforementioned convergent/divergent trajectories combined with the interaction topology of Figure 5.2 guarantees that the two vehicles positioned

at the centre of the group will always be the first to either converge or diverge to within the confines of the orientation zone.

- As Equations (5.20)-(5.21) demonstrate, this initial convergence results in subsequent neighbouring vehicles manoeuvring with trajectories that have a relative angle between them that is equal to the algorithmic parameters, $\psi_{rep,}$ $\psi_{att.}$
- By default, this difference in trajectories results in the *lateral* distances between these vehicles either increasing or decreasing depending on the control law implemented.
- Therefore, overtime and as demonstrated within Figures 5.6 (a) and 5.10 (a) this results in the sequential convergence of every vehicle to within the confines of the orientation zone.

While the results presented in this chapter demonstrate the ability of the modified formation algorithm to efficiently coordinate the vehicles into the required parallel line formation, it has done so under the unrealistic assumption that communication amongst the vehicles is instantaneous. This assumption has meant that whenever the algorithm has calculated the subsequent motion of each vehicle, it has done so with a completely accurate representation of the position and orientation of every other vehicle within the group. In reality, and especially in the underwater environment, this is not the case.

Therefore, the following chapter will now focus on the introduction of a realistic representation of the troublesome underwater communication channel within the formation control algorithm presented in this chapter. Thereafter, the work completed will analyse what effect its introduction has on the ability of the algorithm to coordinate the vehicles in the required parallel line formation.

Chapter 6

Effect of Underwater Communication Channel on Algorithm Efficiency

6.1 Introduction

As discussed in Chapter 5, the work presented thus far has assumed that communication among the vehicles is instantaneous. This has meant that when the formation control algorithm has evaluated each vehicles subsequent heading angle and velocity, it has done so with a continuously accurate representation of both the position and orientation of every other vehicle in the group. In reality though, this is not possible and as a result, the formation control algorithm is unlikely to perform as well as previously indicated in Chapter 5.

The purpose of this chapter therefore is to implement a realistic representation of the underwater communication and test the effect its introduction has on the formation control algorithms ability to generate the required parallel line formation.

To achieve this, the Chapter is structured as follows. Section 6.2 provides a review of the literature associated with the underwater communication channel. Section 6.3 describes the steps taken to ensure a realistic representation of this channel is incorporated within the algorithm. Section 6.4 summarises the simulation setup, assumptions made and parameters varied throughout the simulations completed. Section 6.5 presents and analyses the results obtained from these simulations and Section 6.6 provides a summary of the main outcomes obtained from the work completed in this chapter.

6.2 Underwater Communication Channel

Due to their rapid attenuation in water, radio waves cannot be used as a reliable communication method in the underwater environment (Heidemann et al. 2012). As a result, the vast majority of underwater vehicles are designed to use acoustic communication (Akyildiz et al. 2004). While adopting this method allows data to be transferred over greater distances more reliably (Cella et al. 2009) (Vasilescu et al. 2005; Farr et al. 2010), acoustic communication is nevertheless characterised by low bandwidth (Redfield 2013), large propagation delays (Diamant & Lampe 2011) and unreliability (Brignone et al. 2009). With this work proposing the implementation of acoustic communication, these inherent limitations will be discussed in more detail below.

6.2.1 Bandwidth Limitations

In the underwater environment, bandwidth limitations refer to the relatively slow rate at which acoustic modems can transmit data at. As shown below in Table 6.1, these transmission rates and their associated transmission times vary significantly when compared to the bandwidth available to networks using traditional radio communication.

Table 6.1 - Comparison of transmission rates (MB/s) and associated transmission time for acoustic and radio wave transmission (Kilfoyle & Baggeroer 2000; Parkvall & Astely 2009).

Acoustic		Radio	
Rate (MB/s)	Time (s)	Rate (MB/s)	Time (s)
0.0039	256s	300	0.003s

The values presented above demonstrate that presently in the underwater environment, it takes approximately 256s for an acoustic modem to transmit 1 Mbyte of data. Unsurprisingly, this transmission time is problematically long and as a result, in order to reduce the associated delay as much as possible, the amount of data (packet size) transferred by each vehicle needs to be kept to an absolute minimum.

This is currently achieved by the implementation of a compact control language (Stokey 2005) which restricts the size of each data packet to a value no greater than 32 bytes. As shown below in Table 6.2, this packet size is more than sufficient to transfer the required data associated with the formation control algorithm presented thus far. Subsequently, assuming the use of an acoustic modem with a transmission rate of 31.2 kbits/s (Evo Logics

2014), the transmission time associated with each packet of data used in this work is approximately 0.005s.

Table 6.2- Communication packet used with parameter description and associated size.

Parameter	Size (bytes)
Vehicle ID	4
X-Coordinate	4
Y-Coordinate	4
Z-Coordinate	4
Heading Angle	4

6.2.2 Propagation Delays

In the previous section, the time taken for a commercially available acoustic modem to transmit a data packet of 20 bytes was calculated to be approximately 0.005s. This value though doesn't take into consideration the time it then takes for that same data packet to travel from the transmitting node to the receiving node, i.e from one vehicle to another.

This period of travel is referred to as the propagation delay of the packet and due the speed of sound in water (approx. 1500m/s) being five orders of magnitude smaller than radio communication (Akyildiz et al. 2004) , this parameter has a significant impact on the efficiency of the acoustic communication channel (Stojanovic & Beaujean 2016). As shown below in Equation (6.1), the magnitude of this delay varies depending on the distance between the two communicating nodes, i.e. d_n in Equation (6.1) (Burrowes et al. 2007).

$$T_{Prop} = \frac{d_n}{1500} \tag{6.1}$$

Combining Equation (6.1) with the transmission time presented in Section 6.2.1, the total time between a vehicle beginning to transmit its message and its neighbouring vehicle receiving that message can be written using Equation (6.2).

$$T_{Tot} = T_{Trans} + T_{Prop} \tag{6.2}$$

Analysing Equation (6.2), it becomes apparent why it is impossible for vehicles operating in the underwater environment to have, as assumed, a continuously accurate (real time) representation of its neighbouring vehicle's positional data. Instead, the above equation demonstrates that the data received by each vehicle at time *t*, will actually represent the data relevant at time, *t*-T_{Tot} .

6.2.3 Unreliability

As well as the low bandwidth and large propagation delays discussed above, acoustic communication is also known to be extremely unreliable. This unreliability means that even if a packet of data is transmitted successfully, there is no guarantee that the recipient will successfully receive the data. For example, experimental results presented in (Vasilescu et al. 2005) and (Brignone et al. 2009) demonstrate that success rates can vary between 18% and 100%. The reasons for this unreliability are mainly due to changes in the chemical-physical composition of the underwater environment causing wave phenomena such reflection, refraction, attenuation (Ayaz 2012) as well as multi-path propagation (Partan et al. 2007).

6.2.4 Channel Access Methods

So far, the discussions presented above have only taken into consideration the problems associated with unidirectional communication and not the additional complications associated with bidirectional communication. In order to facilitate bidirectional communication and hence allow the various vehicles to communicate with one another effectively, a suitable *channel access method* has to be implemented.

As discussed in (Stojanovic & Beaujean 2016), channel access methods can be separated into two categories depending on how frequently communication among the various vehicles is deemed necessary. In scenarios where communication is required on a frequent basis, *deterministic access method*s are used. Conversely, where sporadic communication is believed to be sufficient, *random access method*s are commonly used. Since the work presented in this thesis has been based on the assumption that every vehicle has a continuously accurate representation of every other vehicle's positional data, only *deterministic access methods* have been considered in this work.

Currently, the most common of these channel access methods in the underwater domain are Frequency Division Multiple Access (FDMA), Time Division Multiple Access (TDMA) and Code Division Multiple Access (CDMA) (Sozer et al. 2000) (Stojanovic & Beaujean 2016).

As the names suggests, FDMA operates by dividing the available acoustic frequencies among the different users (vehicles) to provide a unique communicating frequency for each user of the channel (Sozer et al. 2000). However, this method has never been used in reality due to the fact that it reduces the already limited bandwidth available to each vehicle and

as a result, produces longer transmission times (Akyildiz et al. 2006; Stojanovic & Beaujean 2016).

The CDMA protocol meanwhile allows each user to simultaneously communicate across the same frequency band with users identified by their unique codes once the data has been received (Heidemann et al. 2012). While promising due to its ability to allow every user simultaneous access to the channel, CDMA suffers from a number of drawbacks including larger bandwidth requirements, greater power consumption and expensive set up costs (Andrews et al. 2007). Furthermore, the implementation of the CDMA is likely to produce a scenario where the problematic near-far problem arises (Muqattash et al. 2003). The near-far problem arises when the transmission of one vehicle blocks the transmission of a second vehicle being received by an additional third vehicle.

Lastly, the TDMA protocol operates by providing each user a unique time slot during which it can broadcast to the remaining users of the channel (Heidemann et al. 2012). These timeslots are predefined and usually include conservative guard times to take into consideration the time delays associated with the aforementioned transmission and propagation delays of each message (Burrowes et al. 2007). Presently, due to its simplicity, the TDMA protocol is the only channel access method to have been implemented experimentally to facilitate inter-vehicle communication (Brignone et al. 2009). Furthermore, as highlighted in (Tena 2018), the majority of current research projects associated with multi-vehicle collaboration in the underwater environment are using the TDMA protocol. Therefore, as the TDMA protocol is currently the most popular protocol for the underwater environment, an accurate representation of this protocol will be incorporated within the work completed in this chapter.

Importantly, while the TDMA protocol is known for its simplicity, it is also the least efficient protocol in terms of channel throughput and utilisation (Chen et al. 2014). This means that it is the protocol that provides each vehicle with the least number of opportunities to communicate with the other members of the group. As a result, incorporating the TDMA protocol presents the overall worst case scenario in terms of inter-vehicle communication when compared with the FDMA and CDMA protocols.

6.3 TDMA Communication Protocol

As discussed above and shown below in Figure 6.1, the TDMA protocol operates by assigning each vehicle a unique *timeslot* during which it has guaranteed and uninterrupted

access to the communication channel. It is during these timeslots that each vehicle broadcasts to the other members of the group.

As illustrated in Figure 6.1, once every vehicle has broadcast its first message, the entire process is restarted and is continuously repeated until the end of the mission. The period between successive communication updates from the *same* vehicle is referred to as the *Communication Cycle Length* and as Figure 6.1 demonstrates, the magnitude of this parameter is dependent on the chosen timeslot size and the number of vehicles within the group.

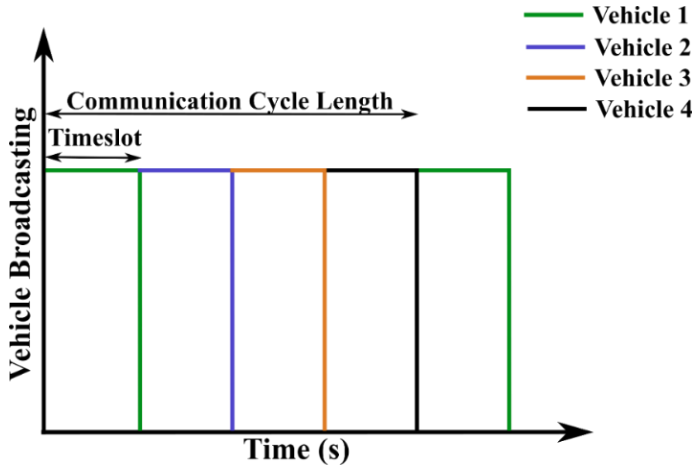


Figure 6.1- Evolution of vehicle broadcasting pattern using TDMA protocol.

6.3.1 Calculating Timeslot Size

As discussed in (Chen et al. 2014; Burrowes et al. 2007), the timeslot size used in the TDMA protocol should be set to minimise the likelihood of a phenomena known as *packet collisions* from occurring. As shown below in Figure 6.2, this phenomena occurs when data sent from two separate sources arrive simultaneously at the same receiving node (Burrowes et al. 2007).

Figure 6.2- Spatial-Temporal diversity between communicating vehicles in the underwater environment (Burrowes et al. 2007). (Figure has been removed due to copyright restrictions).

To reduce the likelihood of these collisions occurring, (Syed et al. 2007; Yackoski & Shen 2008) have demonstrated that both the *Transmission Time* (Section 6.2.1) and *Propagation Delay* (Section 6.2.2) should be taken into consideration when calculating the timeslot size, $T_{Timeslot}$. In addition, it is also widely accepted that a third term known as the *Guard Time* should be included as an extra precaution to further reduce the likelihood of *packet collisions* occurring (Heidemann et al. 2012; Diamant & Lampe 2011; Kilfoyle & Baggeroer 2000; Stojanovic 2003). Combining this *Guard Time*, with the aforementioned *Transmission Time* and *Propagation Delay*, the timeslot size can be calculated using the following equation:

$$T_{Timeslot} = T_T + T_P + T_G \tag{6.3}$$

Where T_T represents the *Transmission Time* presented in Section 6.2.1, T_P is the *Propagation Delay* calculated using Equation (6.1) and T_G is the *Guard Time* and is commonly made equal in magnitude to the propagation delay (Yackoski & Shen 2008).

By evaluating and combining the three parameters shown in Equation (6.3), an appropriate timeslot size can now be evaluated. Before this can be achieved though, it is important to select an appropriate value for the parameter, T_P. The reason that this is important is due to the fact that the value chosen for this parameter is based on the distance between the transmitting and receiving nodes, d_n (Equation (6.1)). As this parameter will be continuously evolving throughout each mission as the vehicles coordinate themselves, a conservative estimate, equal to the maximum range of the acoustic modem is used.

Presently, the range of acoustic modems can be anywhere in the region of 350m to 8 km. As a result, the associated theoretical timeslot sizes will range from 0.5s through to 11.7s. However, figures quoted in the literature suggest that in reality, these values can vary between 6s and 45s (Caiti et al. 2012; Brignone et al. 2009; Sotzing, C.C, Lane 2010; Tsiogkas et al. 2015)(Tena 2018).

Using these figures, it is apparent that in the theoretical best case scenario, each vehicle will receive a communication update every 0.5s and in the worst case, ever 45s. Crucially however, as a result of the protocols cyclical nature, each vehicle will only receive a communication update from its *single* nearest neighbour once every communication cycle. Thus, using the best and worst case scenarios quoted above means that instead of having a continuously accurate representation of its nearest neighbour's position, each vehicle will now only have, at best, an accurate representation once every 6s and at worst, once every 6 minutes.

6.3.2 Incorporating TDMA Communication Protocol

As in Chapter 4, shown below in Figure 6.3 is a simplified representation of the system architecture used to represent the RoboSalmon vehicle.

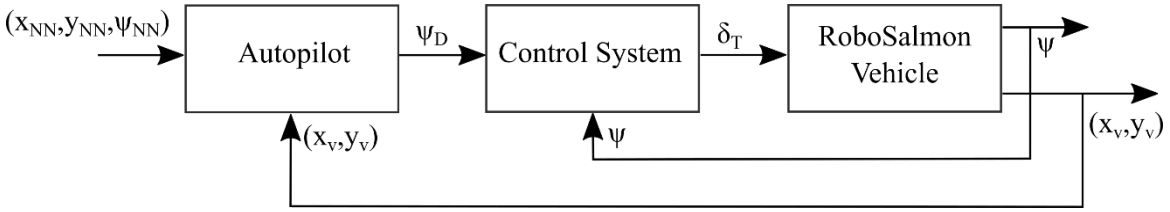


Figure 6.3- Simplified RoboSalmon system architecture.

As discussed throughout Chapters 4 & 5, the *Autopilot* subsystem provides a desired heading angle, ψ_d to the *Control System* that is based on the position and orientation of the vehicles nearest neighbours ($\mathbf{x}_{NN}$, $\mathbf{y}_{NN}$ and $\boldsymbol{\psi}_{NN}$) as well as the vehicle's own position ($\mathbf{x}_v$, $\mathbf{y}_v$). Thereafter, the *Control System* evaluates the necessary tail centreline deflection angle, δ_t that will manoeuvre the vehicle in the required direction. As shown below in Equation (6.4) the positional and heading data used by the formation control algorithm can be compiled into three column vectors

$$\mathbf{x_U} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \end{bmatrix} \quad \mathbf{y_U} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \\ y_{12} \end{bmatrix} \quad \boldsymbol{\psi_U} = \begin{bmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \psi_4 \\ \psi_5 \\ \psi_6 \\ \psi_7 \\ \psi_8 \\ \psi_9 \\ \psi_{10} \\ \psi_{11} \\ \psi_{12} \end{bmatrix} \tag{6.4}$$

So far throughout this work, the values contained in these vectors have represented the most up to date positional information associated with each vehicle in the group. As shown below in Figure 6.4 this means that the positional information used by the formation control algorithm has thus far represented the data applicable to each vehicle during the current time-step, k within the mathematical model.

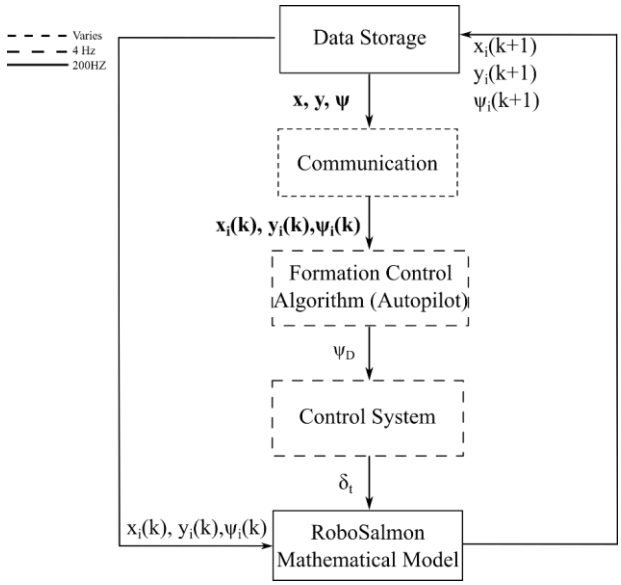


Figure 6.4- Data flow within *Guidance System* with communication for vehicle i at time step, k.

However, in order to implement a realistic representation of the TDMA protocol, it is essential that the data contained within these vectors is now only updated once at the beginning of each timeslot. Moreover, it is also imperative that only the data associated with the vehicle currently broadcasting be updated at the start of each timeslot. Therefore, once the TDMA protocol has been implemented, the frequency at which the communication system will update the vectors shown in Equation (6.4) will have to be equal to the reciprocal of the timeslot size. Furthermore, due to the nature of the TDMA protocol, the data contained within each row of these vectors has to represent the data associated with that vehicle from when it last broadcasted its data.

In order to ensure that this is the case, the data storage functionality shown above in Figure 6.4 has been included. This system operates by storing the values calculated for the parameters, x, y and ψ for each vehicle, at each time step, k in the matrices, $\mathbf{x}$, $\mathbf{y}$ and $\boldsymbol{\psi}$. As a result, these matrices contain the complete time histories of the positional data associated with each vehicle in the group. This means that at any time step, k the data associated with each vehicle when it last broadcast, m number of time steps previously can be acquired. As a result, this guarantees that the data past to the formation control algorithm will represent the data associated with each vehicle the instance they last broadcasted their data packet.

Once a realistic representation of the TDMA protocol has been incorporated, the vectors presented in Equation (6.4) become equal to those shown in Equation (6.5).

$$\mathbf{x}_{\mathbf{U}_j} = \begin{bmatrix} x_1(k-t_i) \\ x_2(k-t_i) \\ x_3(k-t_i) \\ x_4(k-t_i) \\ x_5(k-t_i) \\ x_6(k-t_i) \\ x_7(k-t_i) \\ x_8(k-t_i) \\ x_9(k-t_i) \\ x_{10}(k-t_i) \\ x_{11}(k-t_i) \\ x_{12}(k-t_i) \end{bmatrix} \quad \mathbf{y}_{\mathbf{U}_j} = \begin{bmatrix} y_1(k-t_i) \\ y_2(k-t_i) \\ y_3(k-t_i) \\ y_4(k-t_i) \\ y_5(k-t_i) \\ y_6(k-t_i) \\ y_7(k-t_i) \\ y_8(k-t_i) \\ y_9(k-t_i) \\ y_{10}(k-t_i) \\ y_{11}(k-t_i) \\ y_{12}(k-t_i) \end{bmatrix} \quad \boldsymbol{\psi}_{\mathbf{U}_j} = \begin{bmatrix} \psi_1(k-t_i) \\ \psi_2(k-t_i) \\ \psi_3(k-t_i) \\ \psi_4(k-t_i) \\ \psi_5(k-t_i) \\ \psi_6(k-t_i) \\ \psi_7(k-t_i) \\ \psi_8(k-t_i) \\ \psi_9(k-t_i) \\ \psi_{10}(k-t_i) \\ \psi_{11}(k-t_i) \\ \psi_{12}(k-t_i) \end{bmatrix} \tag{6.5}$$

The vectors presented above clearly demonstrate that as a result of incorporating the TDMA protocol, the formation control algorithm will no longer have access to the most up-to-date positional information, i.e. $\mathbf{x_U(k)}, \mathbf{x_U}(\boldsymbol{k}), \boldsymbol{\psi}_{\mathbf{U}}(\boldsymbol{k})$ but instead, the data accurate from t_i seconds previously. As shown below in Equation (6.6) due to the cyclical nature of the TDMA protocol, the parameter t_i varies depending upon which vehicle is currently broadcasting.

$$t_i = \begin{cases} (|n-i|T_{Timeslot} - T_{Tot}) \, , if \;\; i \leq \text{ n} \\ \left((12-|n-i|)T_{Timeslot} - T_{Tot}\right), if \;\; i > \text{ n} \end{cases} \tag{6.6}$$

Where the parameter, n represents the vehicle currently broadcasting and i identifies the particular row within the vectors. Applying these general equations to the vector, $\mathbf{x_U}$ while assuming n is equal to eight produces the vector shown below in Equation (6.7).

$$\mathbf{x_U} = \begin{bmatrix} x_1\left(k-(7T_{Tmeslot} - T_{Tot})\right) \\ x_2\left(k-(6T_{Timeslot} - T_{Tot})\right) \\ x_3\left(k-(5T_{Timeslot} - T_{Tot})\right) \\ x_4\left(k-(4T_{Timeslot} - T_{Tot})\right) \\ x_5\left(k-(3T_{Timeslot} - T_{Tot})\right) \\ x_6\left(k-(2T_{Timeslot} - T_{Tot})\right) \\ x_7\left(k-(1T_{Timeslot} - T_{Tot})\right) \\ x_8\left(k-(0T_{Timeslot} - T_{Tot})\right) \\ x_9\left(k-(11T_{Timeslot} - T_{Tot})\right) \\ x_{10}\left(k-(10T_{Timeslot} - T_{Tot})\right) \\ x_{11}\left(k-(9T_{Timeslot} - T_{Tot})\right) \\ x_{12}\left(k-(8T_{Timeslot} - T_{Tot})\right) \end{bmatrix} \tag{6.7}$$

Equation (6.7) illustrates that as the vehicle broadcasting progresses from one vehicle to the next, the data associated with the vehicles that previously broadcasted becomes increasingly older. Intuitively, the row associated with the vehicle next in line to broadcast will contain the most out-of-date data, i.e. row nine in Equation (6.7).

Importantly, the eighth row of Equation (6.7) demonstrates that the data received from Vehicle 8 by the remaining members of the group does *not* contain the most up to date representation of the vehicle's position and orientation but instead, the data accurate from T_{Tot} seconds previously. As discussed in Sections 6.2.1-2, this additional term has been included to take into consideration the transmission and propagation delays associated with the underwater environment.

Finally, it is important to note that the vector presented in Equation (6.7) doesn't take into consideration the assumption that each vehicle maintains an accurate representation of its own position and orientation. As a result, the data contained within the vectors, $\mathbf{x}_U$, $\mathbf{y}_U$ and $\boldsymbol{\psi}_U$ will differ for each vehicle in the group. This is shown below in Equation (6.8) where a comparison between the data available to a transmitting vehicle (i.e. Vehicle 8) is compared with the data contained within a vehicle receiving the same data (Vehicle 2).

$$\mathbf{x}_{\mathbf{U_8}} = \begin{bmatrix} x_1(k-(7T_{Timeslot}-T_{Tot})) \\ x_2(k-(6T_{Timeslot}-T_{Tot})) \\ x_3(k-(5T_{Timeslot}-T_{Tot})) \\ x_4(k-(4T_{Timeslot}-T_{Tot})) \\ x_5(k-(3T_{Timeslot}-T_{Tot})) \\ x_6(k-(2T_{Timeslot}-T_{Tot})) \\ x_7(k-(1T_{Timeslot}-T_{Tot})) \\ x_8(k) \\ x_9(k-(11T_{Timeslot}-T_{Tot})) \\ x_{10}(k-(10T_{Timeslot}-T_{Tot})) \\ x_{11}(k-(9T_{Timeslot}-T_{Tot})) \\ x_{12}(k-(8T_{Timeslot}-T_{Tot})) \end{bmatrix} \quad \mathbf{x}_{\mathbf{U_2}} = \begin{bmatrix} x_1(k-(7T_{Timeslot}-T_{Tot})) \\ x_2(k) \\ x_3(k-(5T_{Timeslot}-T_{Tot})) \\ x_4(k-(4T_{Timeslot}-T_{Tot})) \\ x_5(k-(3T_{Timeslot}-T_{Tot})) \\ x_6(k-(2T_{Timeslot}-T_{Tot})) \\ x_7(k-(1T_{Timeslot}-T_{Tot})) \\ x_8(k-T_{Tot}) \\ x_9(k-(11T_{Timeslot}-T_{Tot})) \\ x_{10}(k-(10T_{Timeslot}-T_{Tot})) \\ x_{11}(k-(9T_{Timeslot}-T_{Tot})) \\ x_{12}(k-(8T_{Timeslot}-T_{Tot})) \end{bmatrix} \tag{6.8}$$

These vectors clearly demonstrate that as a result of introducing a realistic representation of the TDMA protocol, the only continuously accurate data available to each vehicle is its own position and heading angle. Unsurprisingly, for the remaining data, its accuracy varies depending upon the current point in the communication cycle and the magnitude of the algorithmic parameters $T_{Timeslot}$ and T_{Tot}.

The above section has described the changes made to the algorithm to ensure a realistic representation of the TDMA protocol has been incorporated within the formation control

algorithm. Moreover, the work presented above has also demonstrated that the two parameters that influence the accuracy of the data available to each vehicle are the chosen timeslot size and the combined transmission and propagation time, T_{Tot}. The following section will now describe the exact values used for these parameters throughout the simulations completed as well as the metrics used to ascertain to what extent introducing the TDMA protocol affects the algorithms ability to coordinate the vehicles into the required parallel line formation.

6.4 Simulation Setup

In order to ensure the results presented in this chapter can be compared with those presented previously in Chapter 5, the initial conditions, i.e. vehicle position & orientation have remained unchanged. As a result, once more the simulations can be separated into two groups depending on the initial control law implemented by the vehicles.

Table 6.3 - Variation in simulation parameters.

Parameter	Group 1	Group 2
Initial Behaviour Utilised	**(Repulsive)**	**(Attractive)**
Behavioural Zones Size (m)	[20,25,200]m	[6,11,200]m
Timeslot Size ($T_{Timeslot}$)	[0.5, 1, 2, 4, 8, 16, 24]s	
Transmission & Propagation Delay (T_{Tot})	0.205s	
Nearest Neighbours Utilised (I_{NN})	11	11
Reference Heading Angle (ψ_{ref})	$\frac{\psi}{4}$	$\frac{\psi}{4}$

Importantly, Table 6.3 also illustrates that the values used for the algorithmic parameter, $T_{Timeslot}$ were varied considerably. As discussed in (McColgan & McGookin 2017), the reason for this variation is to ensure that both theoretically possible values as well as more realistic values are investigated within the study undertaken. Conversely, the value for the parameter, T_{Tot} is set to and kept constant at 0.205s. This specific value is obtained by implementing Equation (6.1) and setting the parameter, d_N equal to the maximum range of the acoustic modem, which for the work completed in this thesis, is equal to 300m (Evo

Logics 2014). This value is then added to the transmission time of 0.005s calculated in Section 6.2.1.

All other algorithmic parameters remain unchanged from those used in Chapter 5. Once again, as with the simulations completed throughout this Thesis, the two groups of simulations presented in Table 6.3 are each composed of 100 different simulations.

6.5 Simulation Results

Throughout this thesis, the main aim of the formation control algorithm has been to ensure that each group of vehicles is able to satisfy the requirements presented in Table 5.3. So far, as the results presented in Chapter 5 demonstrate, the modified algorithm has been able to achieve this due to the fact that each vehicle had a continuously accurate representation of its nearest neighbours position and orientation. However, due to the implementation of the TDMA protocol, it is now apparent that each vehicle will now only have an accurate representation of its nearest neighbour's data once every communication cycle. Therefore, in the intermittent period between successive communication updates, an error will exist between where the vehicle *calculates* its nearest neighbour to be and where it *actually* is. Intuitively, this error will translate into an error in the evaluation of each vehicle's *absolute* and *lateral* nearest neighbour distances. Consequently, the results presented below will begin by analysing how these errors evolve throughout the course of the various simulations completed. Thereafter, to demonstrate whether or not these errors effect the algorithms performance, the evolution of the percentage of vehicles satisfying the various requirements presented in Table 5.3 will be presented and analysed. As with the results presented in the previous chapters, this analysis will be completed by firstly analysing the results obtained when every vehicle is using the *repulsive* control law. Thereafter, Section 6.5.2 will analyse the results obtained when the majority of vehicles initially implemented the *attractive* control law.

6.5.1 Effect of TDMA Protocol on Repulsive Control Law

To begin analysing the results obtained from the simulations discussed above, the results presented below will analyse the evolution of the error associated with the evaluation of each vehicles *lateral* and *absolute* nearest neighbour distances. To achieve this, Section 6.5.1.1 will analyse the results associated with the *lateral* distances while Section 6.5.1.2 will focus on the results obtained for each vehicle's *absolute* nearest neighbour distance.

6.5.1.1 Analysing *Lateral* Distance Error

Shown below in Figure 6.5 is the evolution of the mean error associated with the evaluation of each vehicles *lateral* nearest neighbour distances as the timeslot size is increased. Before the results presented below are analysed it is important to emphasise that the results presented in Figure 6.5 represent the evolution of the mean error evaluated across the 100 simulations completed for each timeslot size. This means that the results presented below represents the data from 700 simulations.

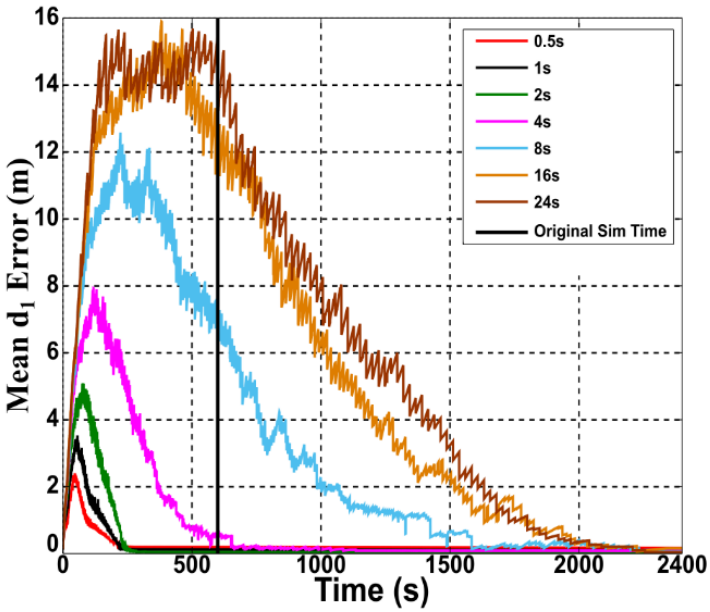
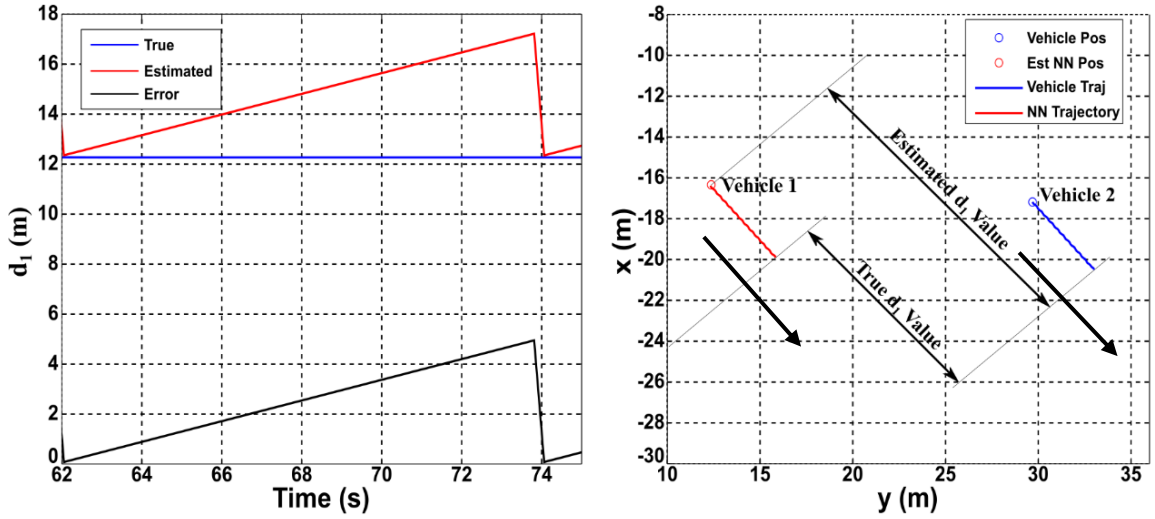


Figure 6.5 - Evolution of the mean error associated with the evaluation of each vehicles lateral nearest neighbour.

The results presented in Figure 6.5 (a) demonstrate that although the error associated with the *lateral* nearest neighbour distance always initially increases, it also subsequently always converges to a value that is approximately equal to zero. In order to understand why this is the case, it is necessary to explain the relationship between the estimated d_1 value, its true value and how the difference between these two values, i.e. the error, evolves throughout a communication cycle. To achieve this, shown below in Figure 6.6 is the evolution of these parameters and the associated vehicle trajectories when the timeslot size is equal to 1s, i.e. a communication cycle length of 12s.

At the start of the communication cycle, i.e. Time = 62s, Figure 6.6 (a) clearly demonstrates that as a result of having just received a communication update from Vehicle 1, the evaluation of d_1 from within Vehicle 2 (red line) is very similar to the true *lateral* distance between the vehicles (blue line). This is to be expected and the reason that they are not

identical is due to the delay in the time taken for the data packet to firstly transmit from the modem of the Vehicle 1 and then travel (propagate) to Vehicle 2, i.e. T_{Tot} in Equation (6.7).



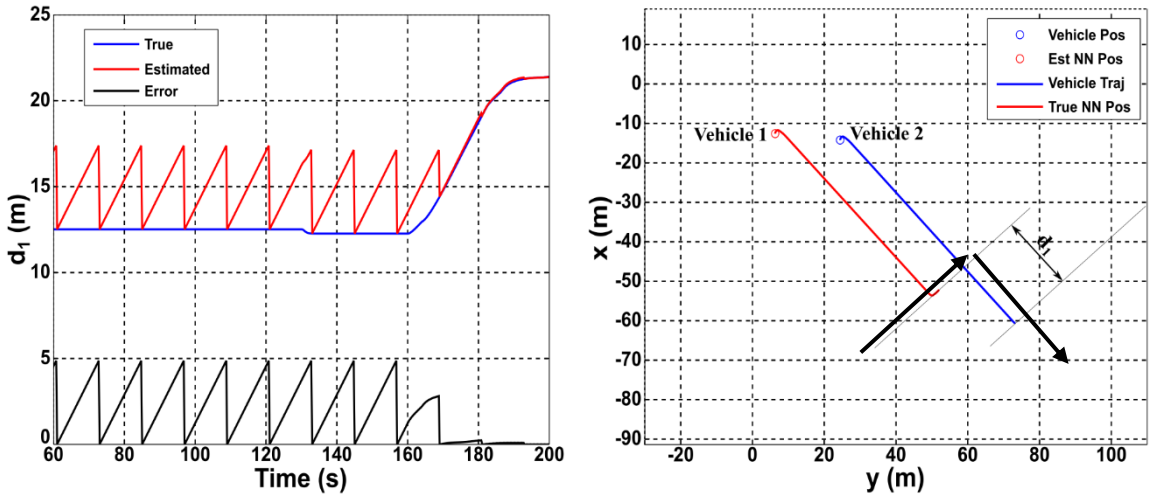
(a) (b)

Figure 6.6 - Comparison of (a) the evolution of the true d_1 value, the estimated d_1 value and the resulting error during one communication cycle and (b) neighbouring vehicles trajectories.

In the subsequent seconds after the communication update, Figure 6.6 (a) demonstrates that the difference between the estimated and true value begins to increase linearly. This behaviour is due to the fact that in the period between successive communication updates from Vehicle 1, Vehicle 2 continues to use the last transmitted position of its nearest neighbour. As Figure 6.6(b) illustrates, because Vehicle 2 is moving away from the last transmitted position of Vehicle 1, the estimated value for d_1 will continue to increase throughout the communication cycle. In reality however, because Vehicle 1 is also using the *repulsive* control law, it too will move in the same direction as Vehicle 2 (Figure 6.6 (b)) and with the same velocity. Intuitively, as shown in Figure 6.6 (a), this means that the true value for d_1 remains unchanged throughout the communication cycle. As a result, the difference between the true and estimated value will continue to increase until the next communication update at 74s. Thereafter, this increase and momentary resetting of the error continues to occur as long as both vehicles continue to use the *repulsive* control law.

However, as soon as one of the vehicles begins to use the *orientating* control law, the aforementioned error no longer increases but instead, as shown below in Figure 6.7 (a), begins to converge towards zero. The reason the error now converges towards zero becomes apparent when the geometry associated with calculating the *lateral* distance between the two vehicles is analysed. If Figure 6.7 (b) is analysed under the assumption

that Vehicle 1 continues to use the *orientating* control law and maintains its perpendicular trajectory to Vehicle 2 (Figure 6.7 (a)), the only way in which the *lateral* distance, d_1, as shown in Figure 6.7(b) can vary is through changes in Vehicle 2's trajectory. As a result, because each vehicle continues to have an accurate representation of its own position and orientation, as soon as Vehicle 2 receives a subsequent communication update from Vehicle 1, it will thereafter be able to accurately calculate the *lateral* distance between the two vehicles. This is demonstrated in Figure 6.7(a) where at approximately 170s – 10 seconds after Vehicle 1 has started using the *orientating* control law - Vehicle 2 receives a communication update from Vehicle 1 and there afterwards, the error, as expected, reduces close to zero and remains unchanged throughout the rest of the simulation.



(a) (b)

Figure 6.7- Comparison of (a) the evolution of the true d_1 value, the estimated d_1 value and the resulting error during multiple communication cycles and (b) neighbouring vehicles trajectories.

The results presented above explain the reasons why the error associated with the parameter, d_1 always converges close to zero. Furthermore, the reason the magnitude of the error increases as the timeslot size increases as shown in Figure 6.5 is due simply to the increase in the communication cycle length associated with a larger timeslot size.

Using the results presented above and the accompanying analysis, it is apparent that the only explanation available for the error in Figure 6.5 always converging towards zero is that each vehicle has successfully transitioned (as required) to implementing the *orientating* behavioural control law. These results imply that despite the implementation the TDMA protocol, the formation control algorithm is still capable of coordinating the vehicles into the required parallel line formation. To demonstrate that this is indeed the case, shown

below in Figure 6.8 is the variation in the evolution of the percentage of simulations satisfying the criteria associated with ensuring the parallel line formation has been generated.

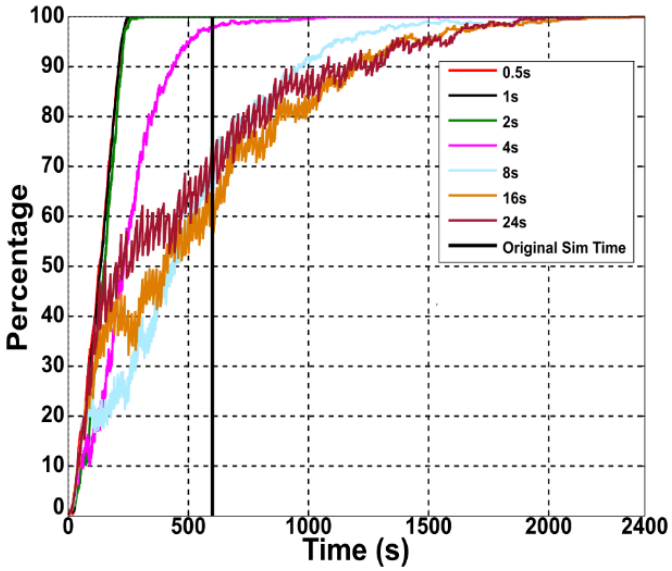


Figure 6.8 - Variation in the percentage of simulations satisfying necessary criteria as timeslot size is increased.

The results presented above clearly demonstrate that despite the introduction of the TDMA protocol, the formation control algorithm is still capable of coordinating the vehicles into the required parallel line formation. However, it is also evident that the convergence time of the algorithm increases considerably from 250s (approx. 4mins) when the timeslot size is equal to 0.5s through to 2200s (approx. 37mins) when the timeslot size is 24s.

In order to understand why the algorithm is still capable of generating the required formation, the following section will now analyse the interaction topology associated with the algorithm and explain why it assists in ensuring the required parallel line formation can be generated.

6.5.1.2 Analysing Group Convergence

The results presented in the previous section demonstrate that despite the introduction of the TDMA protocol, the formation control algorithm is still capable of coordinating the vehicles into the required parallel line formation. However, the results presented above do not explain why, despite the introduction of these significant delays, convergence is still possible. The aim of this is section is to explain in detail the mechanisms associated with

the formation control algorithm that still facilitates the vehicles still being able to generate the required formation.

In order to achieve this though, it is necessary to analyse the relationship between the vehicles trajectories, the decision making process used to select each vehicles nearest neighbour, the evolution of the positional data available to each vehicle and finally, the required vehicle interaction topology. To achieve this, presented below in Figure 6.9 is the required interaction topology of the group as well as an indication of the vehicles heading direction when the *repulsive* control law is implemented.

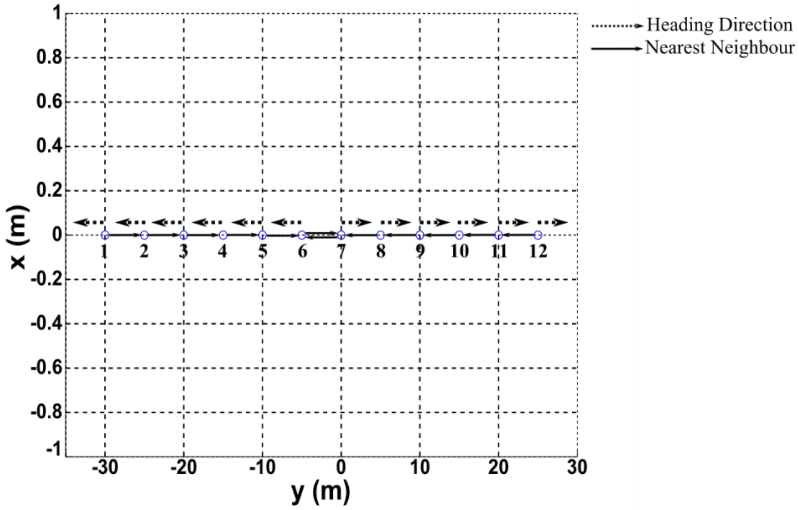


Figure 6.9- Diagram demonstrating interaction topology and vehicle heading angle when every vehicle is using the repulsive control law.

Upon analyses of Figure 6.9, it becomes immediately apparent that once the cyclical nature of the TDMA protocol is incorporated, vehicles implementing the *repulsive* control law are likely to begin selecting the wrong nearest neighbour as the timeslot size is increased. To demonstrate this, the following discussion will focus primarily on the motion of Vehicle 7 and its nearest neighbours, Vehicles 6 and 8 in Figure 6.9.

As shown in Figure 6.9, as a result of implementing the *repulsive* control law, Vehicle 7 will manoeuvre directly towards the last transmitted location of Vehicle 8. As the scenario evolves and if no subsequent communication updates are received from Vehicle 8, Vehicle 7 will eventually manoeuvre past the last transmitted position of Vehicle 8 and as a result, will incorrectly estimate that Vehicle 8 is now positioned immediately to its left hand side. When this occurs Vehicle 7 will alter its nearest neighbour from Vehicle 6 to Vehicle 8 and thus, the required interaction topology is no longer maintained.

Moreover, Figure 6.9 illustrates that if all the vehicles in the group are initially using the *repulsive* control law (as is the case in this work), this incorrect switching of the vehicles nearest neighbour is likely to occur throughout the entire group. The exception to this is the two vehicles positioned at the extreme left and right hand side of the group.

Nevertheless, Figure 6.9 demonstrates that when this is the case and vehicles do begin selecting the wrong nearest neighbour, they will by default, continue to implement the *repulsive* control law. Importantly, this means that the *lateral* distance between Vehicles 6 and 7 will continue to increase. This will continue until Vehicle 7 receives a communication update from Vehicle 8. At which point, it will revert back to using Vehicle 6 as its nearest neighbour. Depending on the initial *lateral* distance between Vehicles 6 and 7 and the length of the communication cycle, it is possible that when the new estimated *lateral* distance between these two vehicles is calculated, it will still be below the boundary of the *orientation* zone. If this is the case, the process described above of Vehicle 7 selecting the wrong nearest neighbour but continuing to implement the *repulsive* control law will be repeated.

At some point, when Vehicle 7 begins to reuse Vehicle 6 as its nearest neighbour, the new estimated *lateral* distance between these vehicles will eventually diverge to a value that is within the confines of either the *orientation* or *attraction* zones. Crucially, as shown below in Figure 6.10, when this occurs and due to the implementation of a different control law, Vehicle 7 will now no longer manoeuvre in the same direction as the last transmitted positions of its neighbouring vehicles (8-12) but instead, in a direction that is either towards, parallel or perpendicular to Vehicle 6, i.e. it's correct nearest neighbour.

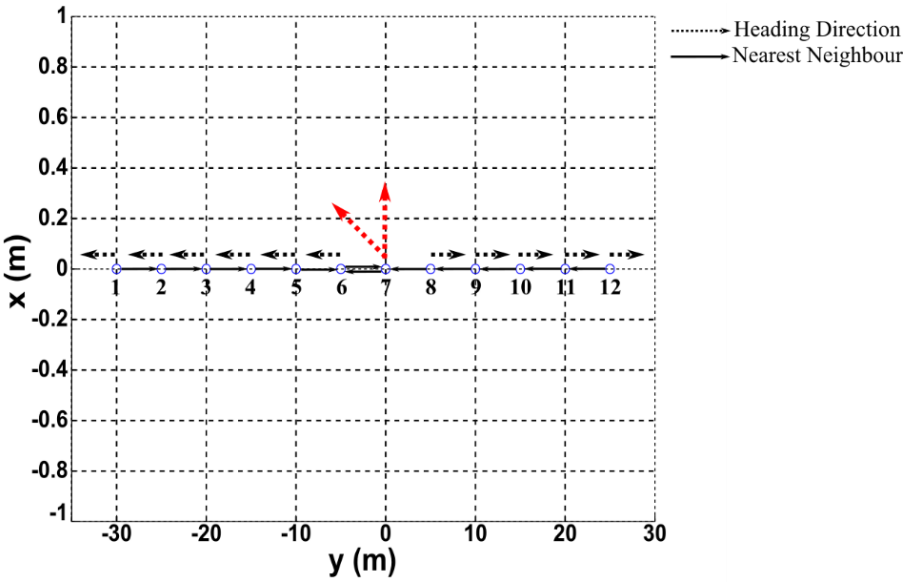


Figure 6.10- Diagram demonstrating interaction topology and Vehicle Seven's potential heading angle when using the attractive or orientating control laws.

Therefore, as a result of Vehicle 7 implementing either the *attractive* or *orientating* control laws, it will now be impossible for it to incorrectly select a vehicle positioned to its right-hand side. This means that once vehicle 7 has transitioned to implementing either the *orientating* or *attractive* control laws, it is guaranteed to thereafter always use the required nearest neighbour.

While the discussion presented above has focussed primarily on Vehicle 7, an identical process will occur with Vehicle 6 where if the timeslot size is sufficiently large, it too, will rely on implementing either the *attractive* or *orientating* control laws in order to guarantee convergence to using the correct nearest neighbour. However, once these two centrally positioned vehicles have converged to using each other as nearest neighbours (as required), the geometry associated with evaluating the *lateral* nearest neighbour distance – as shown in Figure 6.7 - will ensure that the two vehicles are subsequently capable of correctly converging to a d_1 value that is within the confines of the orientation zone.

Thereafter, a similar process to that discussed above will occur for the remaining vehicles until every vehicle has converged to not only the required interaction topology but also the correct *lateral* nearest neighbour distance, d_1. This delayed sequential convergence to the required parallel line formation is shown below in Figure 6.11.

Importantly, the trajectories presented in Figure 6.11 have been taking from a simulation where the timeslot size was set to 24s and as a result, they represent the worst case scenario in terms of the timeslot size and communication cycle length used in this study.

Nevertheless, the results clearly demonstrate that as discussed above, once the middle two vehicles have successfully converged to using the *orientating* control law (Figure 6.11 (a)), one by one, the remaining vehicles also converge (Figures 6.11 (b) – (d)) and the required parallel line formation is eventually generated (Figure 6.11 (e)). As discussed above, the trajectories presented below clearly demonstrate that due to the timeslot size implemented, the vehicles are only able to successfully converge to the required formation after they have implemented the *attractive* control law.

The analysis presented above has demonstrated that if the timeslot size is sufficiently large, the required interaction topology shown in Figure 6.9 can no longer be maintained and if this is the case, the formation control algorithm becomes reliant on the implementation (correct or otherwise) of either the *attractive* or *orientating* control laws to ensure convergence. Furthermore, convergence to the required formation will only occur after the

middle two vehicles (Vehicles 6 & 7 in Figure 6.10) have both converged to the using the *orientating* control law.

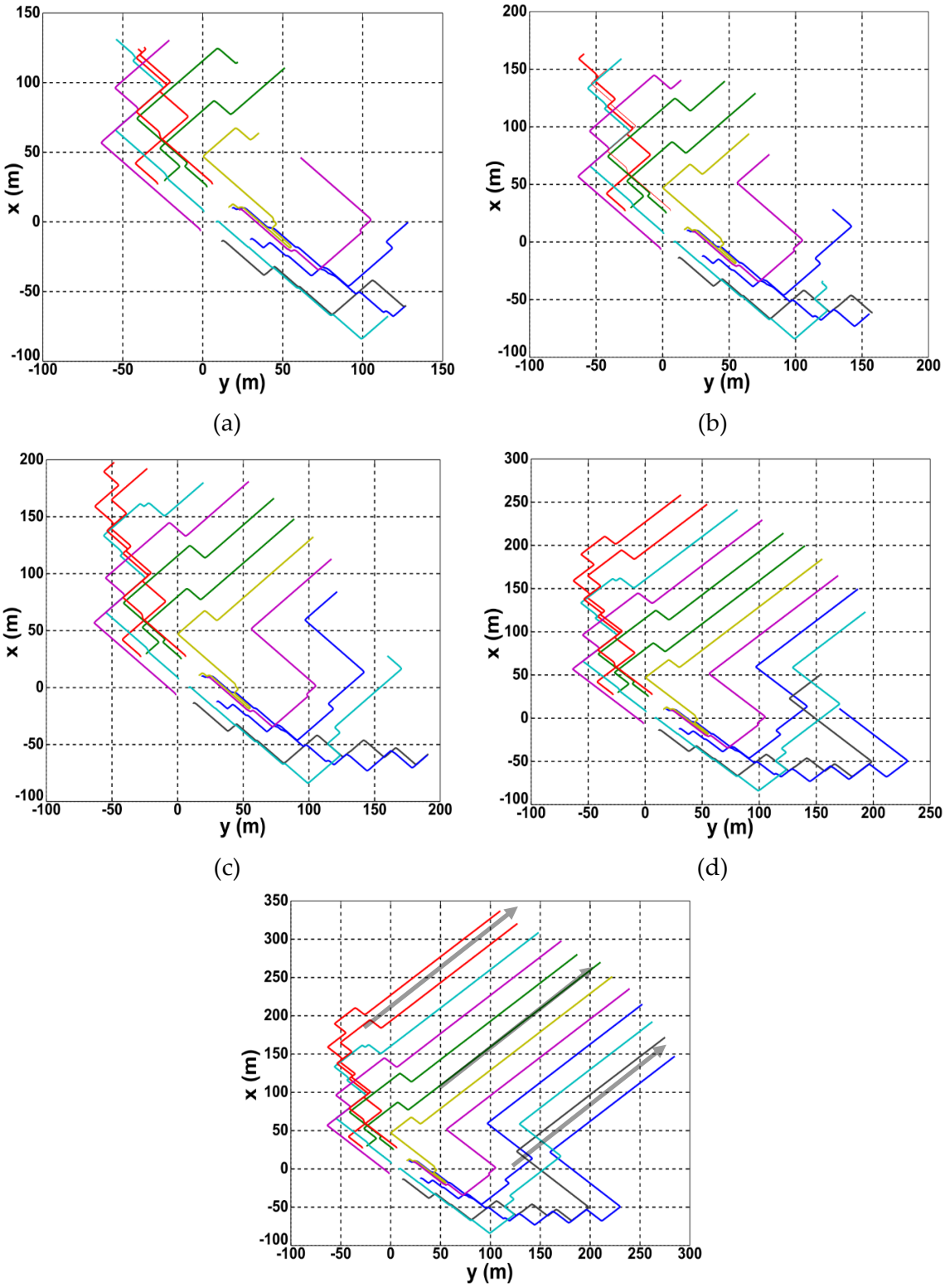


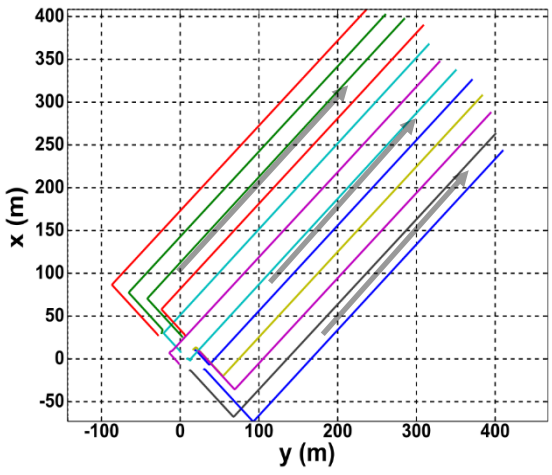
Figure 6.11 - Vehicle trajectories obtained at (a) 690s (b) 900s (c) 1200s (d) 1800s and (e) 2400s when Timeslot Size is equal to 24s.

In terms of the increased convergence time shown in Figure 6.8, it is apparent from Section 6.3 that as the timeslot size is increased, the vehicles will receive less communication updates from each member of the group. As a result, based on the above analysis, it is apparent that when the timeslot size is increased, the vehicles using the wrong nearest neighbour will do so for increased periods of time simply due to the increased communication cycle length. Intuitively therefore, when these vehicles do receive an update, the trajectory corrections they will be required to make will be greater and as a result will require more time correct. To demonstrate that this is indeed the case, shown below in Figure 6.12 is the variation in a group's trajectory as the timeslot size is increased form 1s to 24s. Before analysing the trajectories presented below, it is important to note that the only simulation parameter varied for each trajectory is the timeslot size implemented.

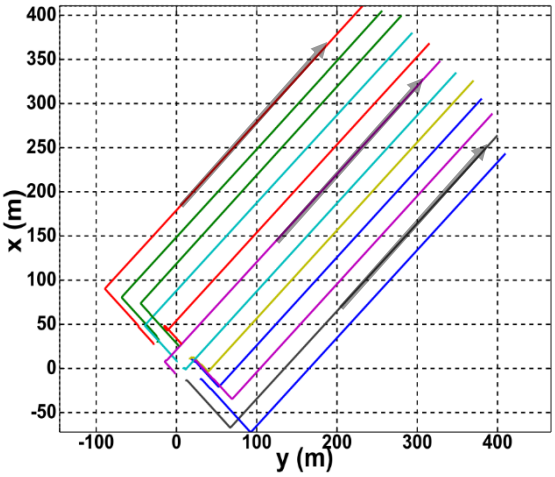
As expected, the trajectories presented below in Figure 6.12 demonstrate that when the timeslot size is relatively small, i.e. less than or equal to two seconds, the difference in the trajectories obtained appears to be minimal with each vehicle, as required, successfully transitioning from the *repulsive* control law to the *orientating* one without significant delay. However, as the timeslot size is increased further to 4s as is the case in Figure 6.12 (d), it becomes apparent that due to the increased communication cycle length, a number of vehicles (at certain points throughout the simulation) have diverged to a *lateral* nearest neighbour distance that is within the confines of the *attraction* zone. Nevertheless, as discussed above, due to the design of the *attractive* control law, the vehicles are subsequently capable of manoeuvring back to within the confines of the orientation zone. In the remaining graphs, i.e. Figures 6.12 (e), (f) and (g), the effect of the increased communication cycle length becomes ever more apparent with the vehicles having to travel greater distances when using the *attractive* control law to correct the extended and incorrect implementation of the *repulsive* control law

In summarising, the results presented in this section have illustrated and explained why, despite the introduction of a realistic representation of the TDMA protocol, the formation control algorithm is still capable – albeit less efficiently – of manoeuvring the vehicles into the required parallel line formation.

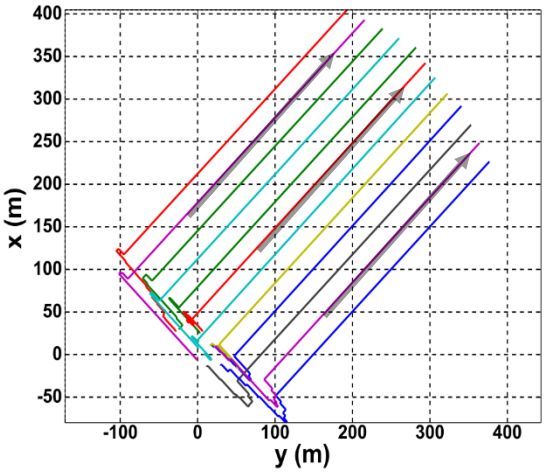
The following section will now analyse the errors associated with evaluation of each vehicles *absolute* nearest neighbour distance and how they influence the algorithms ability to manoeuvre the vehicles as required.



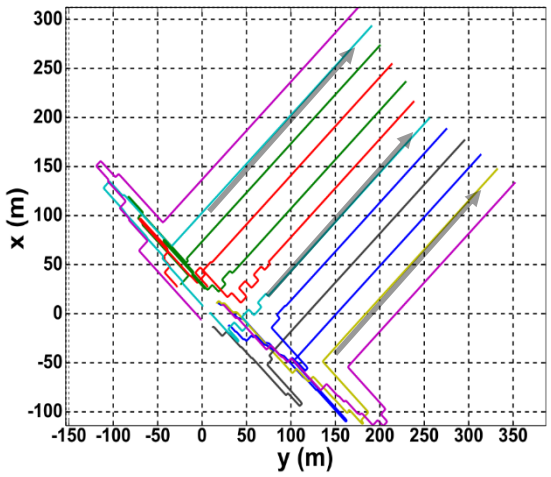
(a) Timeslot – 1s



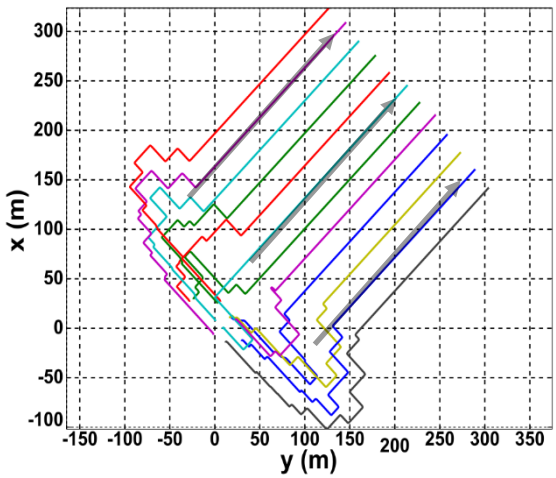
(b) Timeslot – 2s



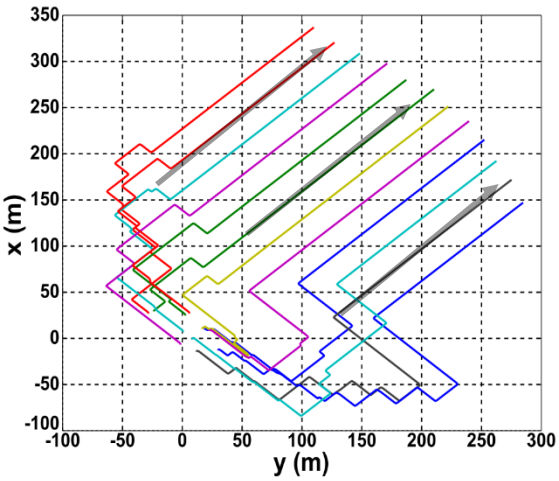
(c) Timeslot – 4s



(d) Timeslot – 8s



(e) Timeslot – 16s



(f) Timeslot – 24s

Figure 6.12- Variation in vehicle trajectories obtained as timeslot size is increased.

6.5.1.3 Analysing *Absolute* Distance Error

So far, the results presented have only taken into consideration the requirements associated with each vehicle's *lateral* nearest neighbour distance, d_l and not those associated with each vehicle's *absolute* nearest neighbour distance. Consequently, this section will use the results presented below in Figure 6.13 as a basis to explain what effect incorporating the TDMA protocol has on the algorithm's ability to ensure each vehicle converges to an appropriate, d_{abs} value.

Contrary to the results associated with the *lateral* nearest neighbour distance, the results presented below demonstrate that the error associated with d_{abs} never converges to zero but instead, to a non-zero value that increases with the timeslot size.

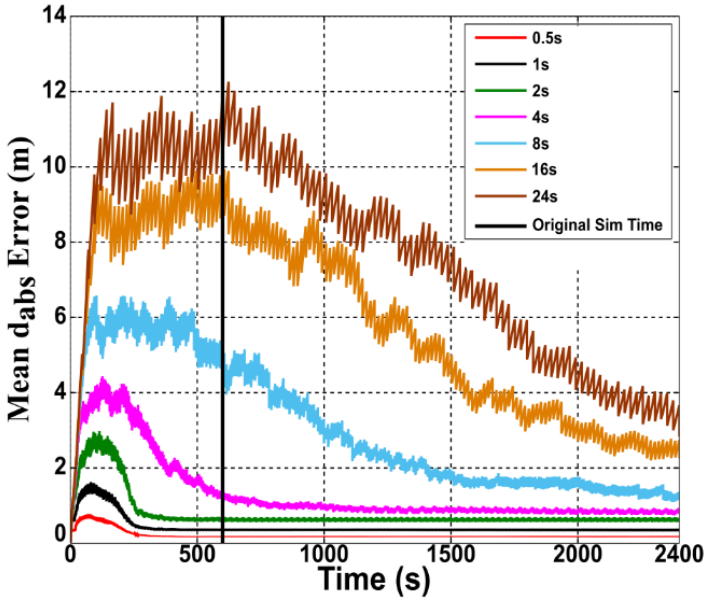
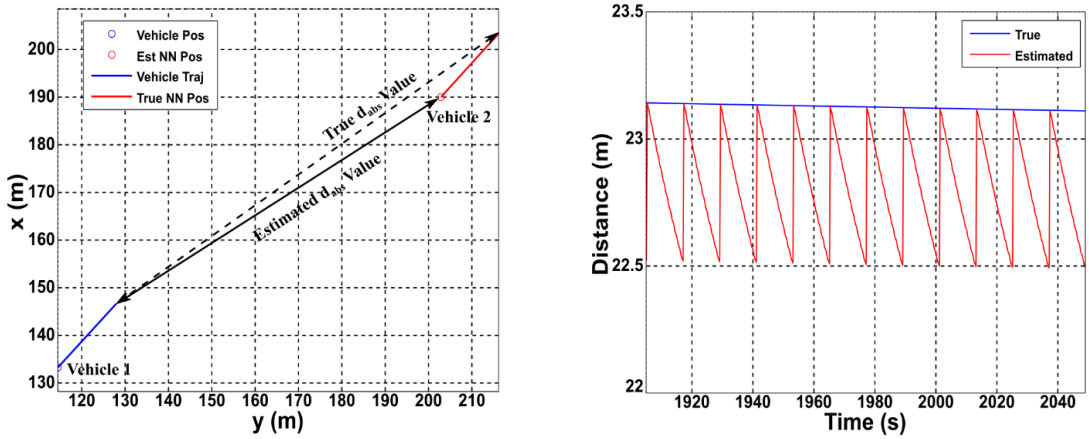


Figure 6.13 - Evolution of the mean error associated with the evaluation of d_{abs}.

As with the results presented above in the previous section, in order to explain why this is the case, it is necessary to analyse the evolution of the estimated *absolute* nearest neighbour distance, the true *absolute* nearest neighbour distance, the associated error between these two values and the relative trajectories of the two vehicles being analysed. Once again, while the results presented below are obtained from a single simulation, they are nevertheless representative of the results obtained throughout the various simulations completed.

The results presented below in Figure 6.14 are similar to those shown previously in Figures 6.6-7 with a cyclical error existing due to the fact that throughout each communication cycle, Vehicle 1 incorrectly believes that Vehicle 2 is stationary.



(a) (b)

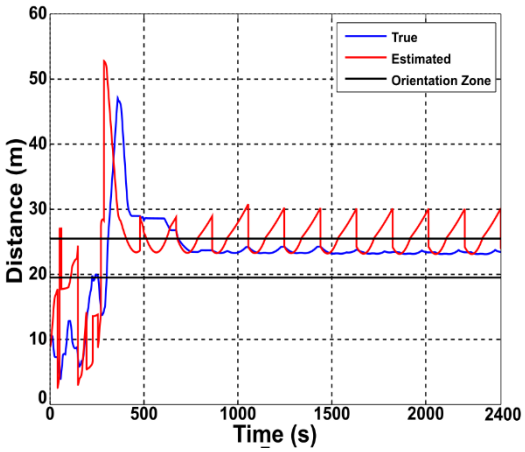
Figure 6.14 - Comparison of (a) neighbouring vehicles trajectories (Exaggerated for clarity) and (b) evolution of true and estimated values for d_{abs}.

However, due to the different geometry associated with evaluating the *absolute* nearest neighbour distance, it is apparent from Figure 6.14 (b) the associated error will never converges to zero but instead, increase steadily throughout each communication cycle before momentarily resetting when a communication update form its nearest neighbour is received.

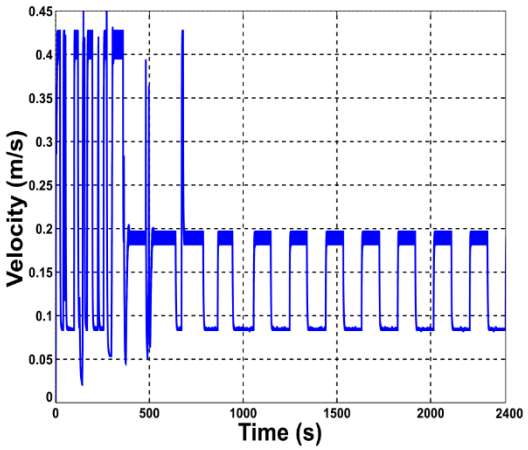
In terms of the algorithm being able to satisfy the relevant criteria associated with the parameter, d_{abs}, the results presented above demonstrate that while an error exists between the estimated and true distances, both are continuously within the confines of the orientation zone, i.e. between 20 and 25 metres. Therefore, in this particular instance, the error has had no effect on the algorithms ability to ensure the parameter, d_{abs} converges to within the confines of the orientation zone. However, if the timeslot size were to be increased, the associated error at the end of each communication cycle length would also increase. As a result, when this happens, it is increasingly likely that at some point throughout each communication cycle, the estimated *absolute* distance will diverge to a value that is no longer within the confines of the orientation zone. This is shown to be the case below in Figure 6.15 where the data presented is from the same simulation as Figure 6.14 except the timeslot size has been increased to 16s, i.e. a communication cycle length of 192s.

The results shown below illustrate that because the estimated value for d_{abs} switches between the *attraction* and *orientation* zones, the vehicles velocity also continuously. This is due to velocity control law trying to get the value for d_{abs} to converge. This behaviour is

extremely inefficient as it means the vehicle is continuously and unnecessarily consuming additional power altering the *Tail Beat Amplitude* of the vehicle.



(a) – Evolution of Estimated & True Values obtained for d_{abs}



(b) - Associated Variation in Vehicle Velocity

Figure 6.15– Relationship between evaluation of d_{abs} and associated variation in vehicle velocity.

The results presented above demonstrate that due to the different geometry associated with the evaluation of the *absolute* nearest neighbour distance, the error associated with this parameter will never converge to zero. The results also demonstrate that as the timeslot size increases, it becomes increasingly likely that this error will result in the estimated value for d_{abs} continuously fluctuating between the different behavioural zones. Therefore, as shown below in Figure 6.16, the percentage of vehicles able to satisfy the algorithmic requirement associated with this parameter decreases as the timeslot size is increased.

Finally, it is important to note that the results presented above are thus far only representative of the scenario where every vehicle is initially using the *repulsive* control law. While the results presented above have demonstrated that the implementation of the *attractive* control law is a prerequisite to ensuring convergence to the required formation (when the timeslot size is increased), the following section will nevertheless focus on analysing the results obtained from this second group of simulations.

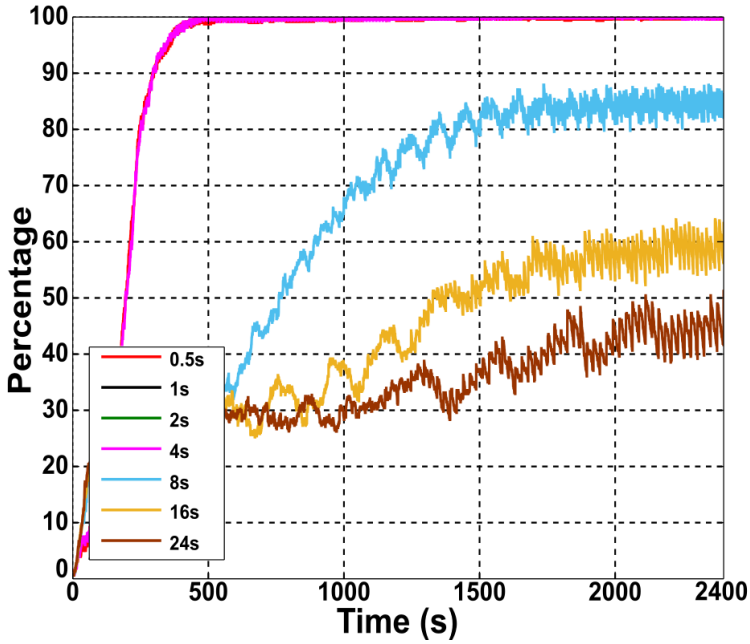
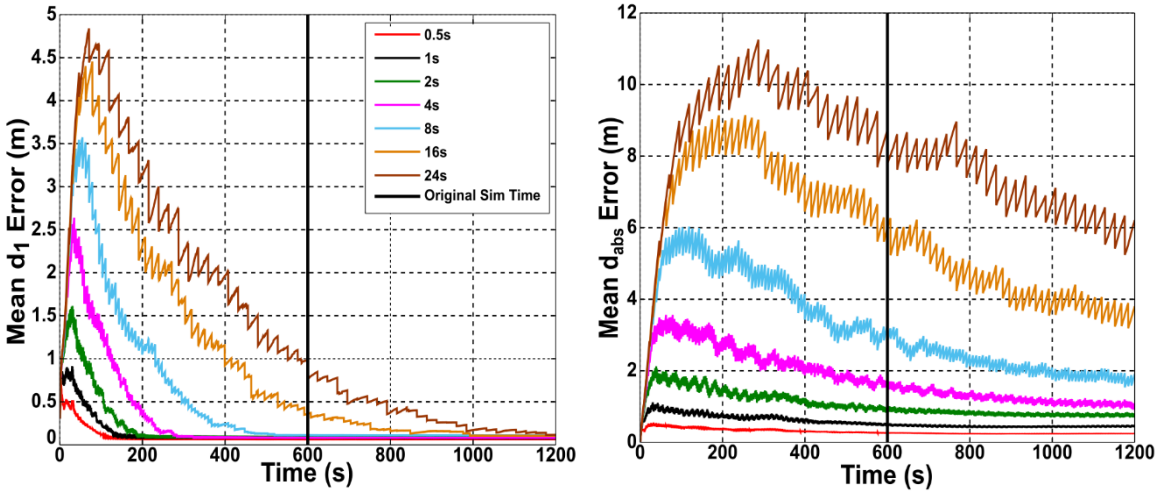


Figure 6.16- Variation in the evolution of the percentage of vehicles satisfying the necessary criteria associated with the d_{abs}.

6.5.2 Effect of TDMA Protocol on Attractive Control Law

To analyse the effect of introducing the TDMA protocol on the vehicles initially using the *attractive* control law, shown below in Figure 6.17, is the evolution of the errors associated with the evaluation of the *lateral* and *absolute* nearest neighbour distances as the timeslot size is increased from 0.5s through to 24s.



(a) (b)

Figure 6.17- Evolution of the mean error associated with the evaluation of the parameters (a) d_1 and (b) d_{abs} as the timeslot size is increased.

The results presented in Figure 6.17 (a) are similar to those presented previously in Figure 6.5 with the error value always converging to a value approximating zero. Similarly, the results presented in Figure 6.17 (b) demonstrate that the error associated with the parameter d_{abs} is unable to converge towards zero but instead increases with the timeslot size. Once again, this is analogous to the results presented above in Figure 6.13 when the *repulsive* control law was initially implemented. As discussed in Sections 6.5.1.1 and 6.5.1.3, this variation in convergence is due to the different geometry associated with the evaluation of the *lateral* and *absolute* nearest neighbour distances. While the results presented in Figure 6.17 suggest that the effect of the TDMA protocol on the *attractive* control law is no different to that of the *repulsive* control law, the following two sections will analyse these results in more detail in order to ensure they are consistent with those obtained in the previous section.

6.5.2.1 Analysing *Lateral* Distance Error

While the general trends presented in Figure 6.17 (a) are similar to those shown in Figure 6.5, it is apparent that the magnitude of the errors associated with the *lateral* distance in Figure 6.17 (a) are significantly smaller. For example, the maximum error presented in Figure 6.17 (a) is just over 4.5m while in Figure 6.5, the maximum error is approximately 16m. This difference is due primarily to the fact that as presented in Chapter 5, vehicles using the *attractive* behavioural control law, manoeuvre towards their nearest neighbour at an angle of 45º. This is half of the 90º angle used within the *repulsive* control law. As a result, because neighbouring vehicles are moving towards one another at a reduced rate, the associated error increases at a slower rate and as a result, the magnitude of the error at the end of each communication cycle length will inherently be smaller.

Nevertheless, the results presented in Figure 6.17 (a) clearly illustrate that the error value associated with the algorithmic parameter, d_1 always converges towards zero. As the results presented in Section 6.5.1.1 demonstrate, this is indicative of the formation control algorithm generating the required parallel line formation. To demonstrate that this is indeed the case, shown below in Figure 6.18 is the evolution of the percentage of simulations that were able to satisfy the necessary criteria that ensures the parallel line formation was generated by the vehicles.

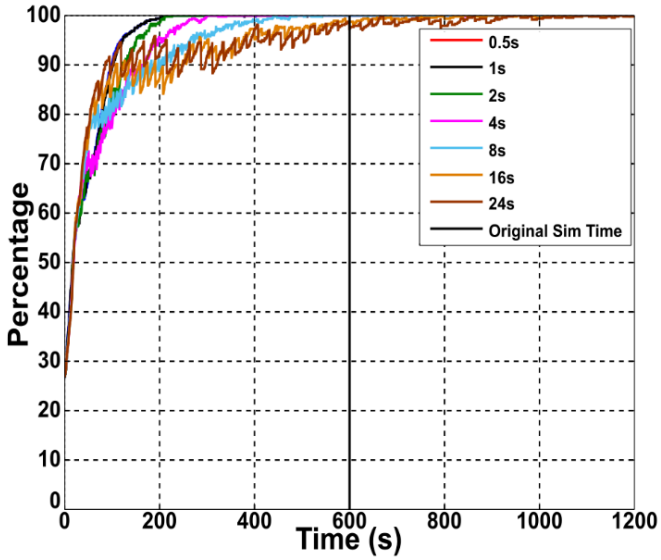


Figure 6.18- Variation in the percentage of simulations satisfying necessary criteria associated with each d_1 as timeslot size is increased and vehicles are initially using attractive control law.

As expected, the results presented above illustrate that regardless of the timeslot size implemented, the formation control algorithm has been to satisfy the necessary criteria. However, once again, similarly to the results obtained in Section 6.5.1.1, the results presented in Figure 6.18 demonstrate that as the timeslot size is increased, the time taken for the formation control algorithm to generate the required formation increases significantly. Unsurprisingly, this increase in the time taken is due to the exact same reasons discussed in the previous section, i.e. vehicles receiving communication updates less frequently as a result of the timeslot size being increased.

Therefore, combining the results presented in Figure 6.17 (a) with those shown in Figure 6.18, it is apparent that in terms of the algorithmic parameter, d_1, the TDMA protocol has the exact same effect on the algorithms performance as it did in the previous section.

6.5.2.2 Analysing *Absolute* Distance Error

Comparing the results presented in Figure 6.17 (b) with those in Figure 6.13, it is apparent that the general trends presented in both graphs are very similar with the error increasing with timeslot size. This is not unexpected and is due to the fact that the velocity control law operates exactly the same regardless of whether the *attractive*, *orientating or repulsive* control law is used. As a result, the explanation provided in Section 6.5.1.3 as to why the error associated with the *absolute* distance increases with the timeslot size is applicable to the results presented in this section. To demonstrate that this is indeed the case, as with the results presented in Figure 6.16, shown below in Figure 6.19 is the evolution of the

percentage of vehicles satisfying the necessary criteria associated with each vehicle's *absolute* nearest neighbour distance.

As expected, Figure 6.19 demonstrates that as the timeslot size is increased, the velocity control algorithm finds it increasingly difficult to ensure every vehicle converges to an *absolute* nearest neighbour distance that is within the confines of the orientation zone. Furthermore, the cyclical nature of the responses presented in Figure 6.19 is indicative of the behaviour presented in Figure 6.16 where the value calculated for d_{abs} is continuously switching between the *orientating* and *attractive* behavioural zones.

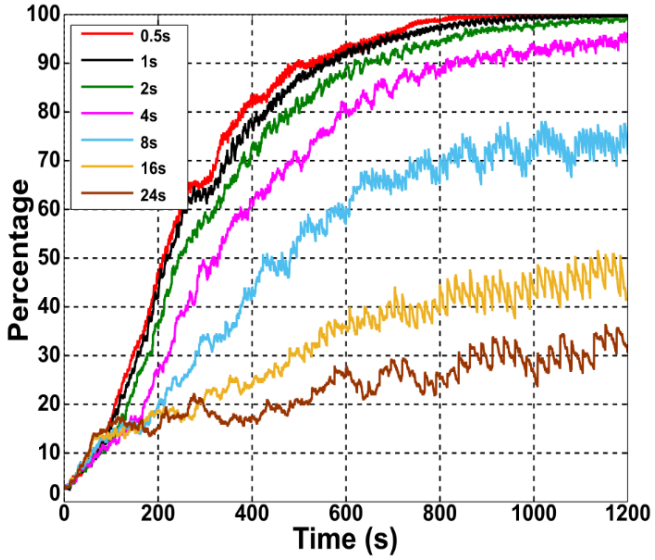


Figure 6.19- Variation in the evolution of the percentage of vehicles satisfying the necessary criteria associated with d_{abs} as timeslot size is increased and vehicles are initially using attractive control law.

6.6 Summary

The results presented above in Section 6.5 have provided a detailed analysis of the results obtained from the simulations undertaken to investigate what extent incorporating a realistic representation of the TDMA protocol effects the algorithm's ability to generate the required parallel line formation. The aim of this section is to now provide a summary of the pertinent results obtained from this analysis and to subsequently provide any recommendations that if incorporated, may alleviate the degradation in the algorithms performance presented in this chapter.

- The results presented in Sections 6.5.1.1-3 demonstrate that due to a combination of the geometry associated with evaluating the *lateral* nearest neighbour distance, the cyclical nature of the TDMA protocol and the design of the *attractive* control law, it

is still possible for the formation control algorithm to generate the parallel line formation regardless of the timeslot sizes implemented.

- This convergence demonstrates that the delays associated with the transmission and propagation delays of each vehicles message have no effect on the algorithms ability to generate the required formation.
- Figures 6.8, 6.12 and 6.18 illustrate that while convergence to the parallel line formation still occurs, the time taken and distances travelled by the vehicles before the algorithm generates the required formation increases significantly with the timeslot size implemented.
- Section 6.5.1.3 and Figures 6.14, 6.15 & 6.16 demonstrate that due to the geometry associated with evaluating each vehicle's *absolute* nearest neighbour distance and the cyclical nature of the TDMA protocol, convergence of this parameter to a value that is continuously within the confines of the orientation zone cannot be guaranteed particularly for the larger, more realistic timeslot sizes, i.e. 8s, 16s & 24s.
- Furthermore, Figure 6.16 (b) demonstrates that this inability to converge results in the formation control algorithm continuously altering the velocity of the associated vehicle.
- The only difference between the two sets of results presented in Sections 6.5.2.1 and 6.5.2.2 is that magnitude of the error associated with the parameter, d_1 was significantly smaller when the *attractive* control law was implemented. As discussed though, the reason for this smaller error was due to the different values used for the algorithmic parameters, ψ_{rep} and ψ_{att}.

Therefore, the results presented in this chapter demonstrate that while the formation control algorithm is still capable of generating the required parallel line formation, its performance is severally degraded, particularly when realistic values are used for the timeslot size, i.e. 8s, 16s and 24s. This degradation is due to the severely restricted communication that exists between each vehicle and its nearest neighbour, i.e. once every communication cycle.

Based on these results, it is apparent that in order to ensure the formation control algorithm operates as efficiently as possible regardless of the timeslot size implemented, additional functionality has to be implemented within the algorithm. To work, this functionality needs to be capable of providing each vehicle with an accurate representation of the positioning and orientation of neighbouring vehicles during each communication cycle. Conversely, an alternative control method which would allow the vehicles to satisfy the various convergence criteria without requiring access to up to date information would also alleviate

the problems discussed in this chapter. Subsequently, the following chapter introduces the implementation of two such methods and the associated analysis to determine whether or not their incorporation improves the performance of the algorithm when the TDMA protocol is incorporated.

Chapter 7

Comparison of Delay Tolerant Formation Control Algorithms

7.1 Introduction

The results presented in Chapter 6 illustrate that as a result of implementing a realistic representation of the underwater communication channel, the ability and efficiency of the formation control algorithm deteriorates as the timeslot size is increased. As discussed in Chapter 6, this deterioration is caused by each vehicle receiving less communication updates when the timeslot size is increased resulting in an increase in the number of vehicles making incorrect decisions for increasingly longer periods of time. Furthermore, while the results demonstrate that the formation control algorithm is still capable of generating the required parallel line formation, the time taken to do so increases significantly. The results also demonstrate that due to a combination of the cyclical nature of the TDMA protocol and the geometry associated with evaluating the *absolute* nearest neighbour distance, convergence of this parameter to a value that is within the confines of the orientation zone isn't guaranteed.

The purpose of this chapter is to introduce and compare two different methodologies that once incorporated, it is envisioned will allow the formation control algorithm to operate as efficiently as possible regardless of the timeslot size implemented. The first of these methodologies will introduce functionality that will attempt to predict the position and orientation of each vehicles nearest neighbours in the period between successive communication updates. In doing so, it is anticipated that the algorithm will operate similarly to when instantaneous communication was assumed in Chapter 5.

The second methodology meanwhile, will seek to assign a waypoint to each vehicle that once assigned, will guarantee that the required parallel line formation is generated. By assigning each vehicle a unique waypoint, it is anticipated that the formation control algorithm will no longer need an accurate representation of each vehicles nearest neighbour's position and orientation.

Unsurprisingly, the main purpose of including these methodologies is to ensure that convergence to the required formation is achieved as efficiently as possible *and* regardless of the communication timeslot size implemented, i.e. with minimum delay and control system effort.

Chapter 7 presents this comparative study as follows. Section 7.2 describes the implementation of these two methodologies within the modified formation control algorithm presented in chapter 5. Section 7.3 presents a summary of the simulation setup and the algorithmic parameters varied throughout the simulations completed. Section 7.4 analyses and compares the results obtained from the implementation of these two different methodologies. Section 7.5 provides a summary of the outcomes from this comparative study.

7.2 Implementation of Delay-Tolerant Functionality

The purpose of this section is to describe the implementation of the two methodologies discussed above that have been designed specifically to ensure the successful convergence of the group to the required parallel line formation. In addition, these algorithms should also coordinate each vehicle to the required *absolute* nearest neighbour distance as efficiently as possible *and* regardless of the nuances associated with the underwater communication channel.

As alluded to in Chapter 6 and discussed above, the first of these methodologies involves introducing predictive functionality within the formation control algorithm. The aim of this predictive functionality is to provide each vehicle with a more accurate estimate of the positioning and orientation of neighbouring vehicles in the intermittent period between successive communication updates. The intention of introducing such functionality is to allow the required parallel line formation to be generated as quickly as possible despite the nuances of the underwater communication channel.

Conversely, the second methodology seeks to remove the need for each vehicle to have a continuously accurate representation of the position and orientation of neighbouring vehicles altogether. Instead, this methodology implements a consensus based formation

control algorithm that provides each vehicle with a single waypoint to manoeuvre towards. This is achieved by each vehicle selecting a waypoint based on its relative position within the group. Thereafter, the vehicles can only begin manoeuvring to their waypoints once it has been established that each vehicle has their own unique waypoint and hence, position in space to manoeuvre towards.

7.2.1 Implementation of Trajectory Prediction Functionality

As discussed in Chapter 6, each vehicle only has the most accurate representation of its single nearest neighbour's position once every communication cycle. This reduces the efficiency of the coordination algorithm as the timeslot size is increased. In order to combat this problem it is proposed that a predictive methodology is employed to predict the position and orientation of each vehicle's nearest neighbours in the interim period between successive communication updates. The structure of this prediction functionality and how it is incorporated within the formation control algorithm is presented below in Figure 7.1.

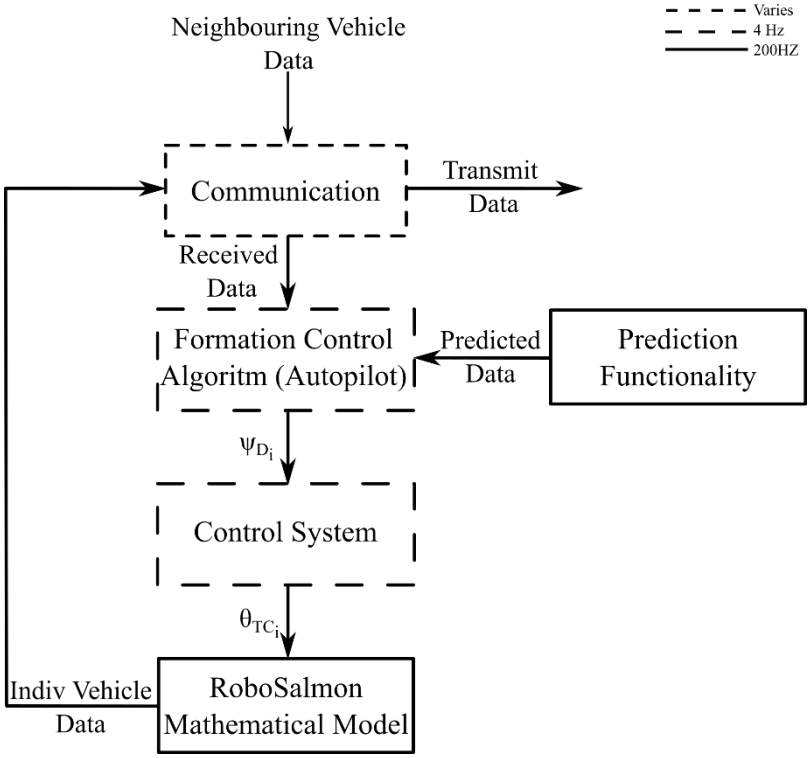


Figure 7.1- System structure with predictive functionality included.

Figure 7.1 demonstrates that as well as having access to the data broadcast by each vehicle, the formation control algorithm will now also have access to the data provided by the newly incorporated predictive functionality. Subsequently, it is expected that in the period between communication updates, the data evaluated from this new functionality will now be used within the formation control algorithm to determine which particular control law

the vehicle should implement. This predicted data will then be replaced by the actual value when the appropriate vehicle's data is received throughout each communication cycle. Thereafter, the predictive functionality will be used once more and this entire process will be repeated throughout each mission.

In order to ensure that the predicted values provided by this new functionality are as accurate as possible, the functionality incorporated will not only have to contain an accurate representation of the dynamics of the individual vehicles but also an accurate representation of the formation control algorithm presented in Chapter 5. This is to ensure that as the simulation evolves, the predictive functionality continues to produce an accurate estimate of the positional and heading data associated with the vehicles nearest neighbours.

To satisfy these requirements, the dynamics of each vehicle is evaluated using the exact same reduced fidelity mathematical model of the RoboSalmon vehicle presented in Chapter 3. Furthermore, the predictive functionality also contains an identical copy of the formation control algorithm presented in Chapter 5. Therefore, theoretically, this predictive functionality should be able to accurately model and predict the motion of each vehicle's nearest neighbours. As a result, the prediction functionality once incorporated, should be able to provide the formation control algorithm with an accurate representation of each vehicle's position and orientation in the intermittent period between communication updates.

However, in order for this predictive functionality to provide an accurate estimate of the position and orientation of the vehicles nearest neighbour, it also needs to be able to accurately evaluate the position and orientation of every other vehicle in the group. As shown below in Figure 7.2, this is due to the interaction topology of the vehicles.

Analysing the interaction topology presented above, it becomes apparent that in order to ensure that Vehicle 7 no longer incorrectly alters its nearest neighbour as discussed in the previous chapter (Section 6.5.1.2), the predictive functionality implemented will have to predict the motion of each vehicle positioned to its right hand side, i.e. Vehicles 8-12. Furthermore, in order to ensure that Vehicle 7 can successfully transition from using the *repulsive* control law to the *orientating* one, it also needs to have a continuously accurate estimate of the position of Vehicle 6. Intuitively, based on the discussion presented above, in order to achieve this, the position of Vehicles 1-5 will therefore also have to be estimated. This means that, in order to ensure every vehicle has a continuously accurate estimate of its single nearest neighbour, the motion of every other vehicle within the group has to be predicted.

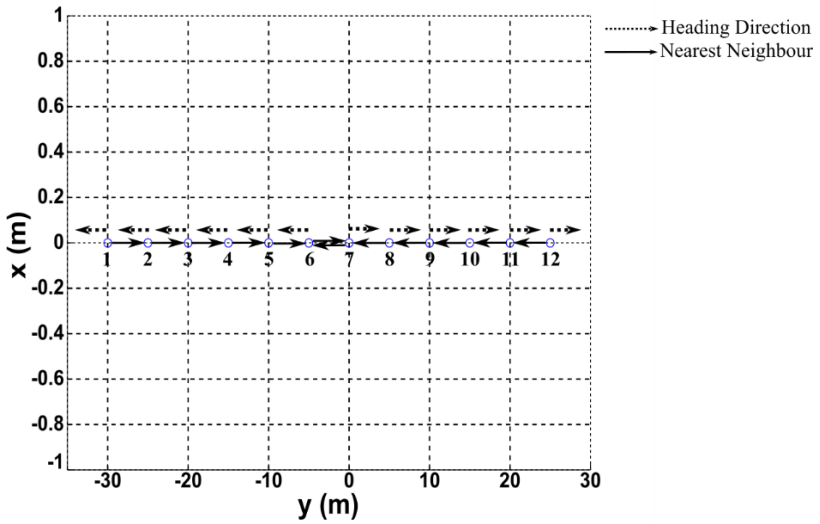


Figure 7.2- Diagram demonstrating interaction topology and vehicle heading direction when every vehicle is using the repulsive control law.

Subsequently, within the predictive functionality shown in Figure 7.1, the mathematical model of the RoboSalmon vehicle and the formation control algorithm has to be evaluated N-1 times, i.e. once for every neighbouring vehicle within the group. This is presented diagrammatically in Figure 7.3.

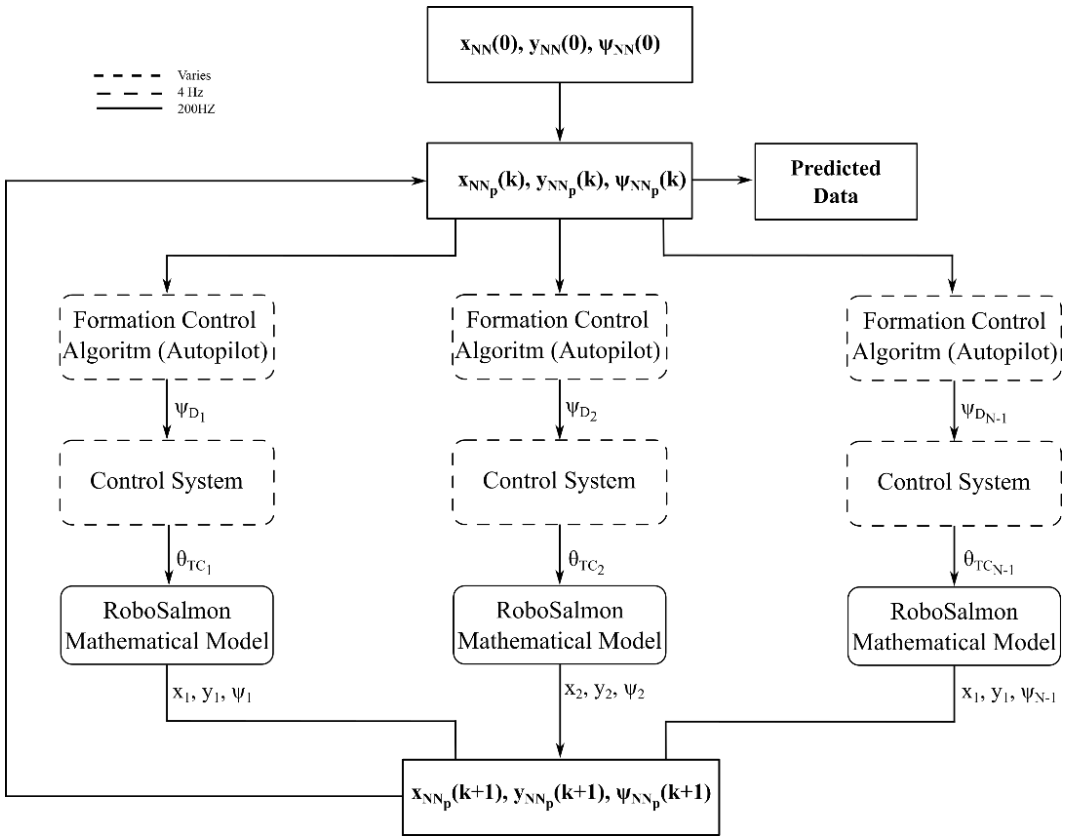


Figure 7.3- Flow diagram representing functionality contained within predictive methodology.

In the above diagram, the vectors $\mathbf{x_{NN_p}}$, $\mathbf{y_{NN_p}}$ and $\boldsymbol{\psi}_{\mathbf{NN_p}}$ represent the coordinates and heading angles of all vehicles as calculated from within the predictive functionality. Importantly, Figure 7.3 demonstrates that at the start of each simulation, i.e. k = 0, the initial values used within the predictive functionality are equal to the true values associated with each vehicle. It is important to note that at the start of each simulation every vehicle is assumed to have an accurate representation of the initial position and orientation of every other vehicle within the group.

Thereafter, based on the initial values contained within $\mathbf{x_{NN_p}}$, $\mathbf{y_{NN_p}}$ and $\boldsymbol{\psi}_{\mathbf{NN_p}}$ the predictive functionality evaluates each vehicles desired heading angle, ψ_{D_i}, the associated control surface deflection, δ_{TC_i} and then finally, the current position and orientation of each vehicle, x_i, y_i and ψ_i. Thereafter, the vectors, $\mathbf{x_{NN_p}}$, $\mathbf{y_{NN_p}}$ and $\boldsymbol{\psi}_{\mathbf{NN_p}}$ are updated and packaged as the predicted data. As Figure 7.1 demonstrates, this predicted data is then used in the intermittent period between successive communication updates to provide the vehicle with an estimate of the position and orientation of its nearest neighbour.

7.2.2 Consensus Based Waypoint Selection Formation Control Algorithm

In order to negate the effects of incorporating the TDMA protocol, the work presented in this section proposes the introduction of a waypoint consensus methodology. Instead of vehicles selecting the appropriate control laws based on the distance to their nearest neighbour, this methodology will alter the control law based on the *lateral* distance to the vehicles chosen waypoint. To achieve this though, functionality has to be implemented that will firstly produce the necessary waypoints and then secondly, allow each vehicle to select the most appropriate waypoint for it to track and manoeuvre towards. As well as completing these tasks, the functionality incorporated will also have to ensure that every vehicle is assigned a unique waypoint. This is to ensure that a number of vehicle do not select and then manoeuvre to the same waypoint. The creation and implementation of the functionality that will achieve this is presented in the following sections.

7.2.2.1 Creation of Waypoints

Before the method used to create the aforementioned waypoints is presented, it is necessary to emphasise two important facets of information with regards to the mission profile outlined in Figure 1.3. The first is that the initial position and orientation of each vehicle within the deployment area are known by each member of the group. The second important assumption is that the recovery area shown in Figure 1.3 is represented by a single user defined coordinate, (x_{des}, y_{des}) which is known to each vehicle. As shown below in

Equations (7.1) and (7.2), this coordinate is particularly important as it is used to calculate the first of the waypoints used in this methodology.

$$Y_{Waypoint_1} = y_{des} - \frac{N}{2}\left(\frac{zoo_u + zoo_l}{2}\right)\cos(\psi_{ref}) \tag{7.1}$$

$$X_{Waypoint_1} = x_{des} + \frac{N}{2}\left(\frac{zoo_u + zoo_l}{2}\right)\sin(\psi_{ref}) \tag{7.2}$$

Here N represents the number of vehicles within the group, x_{des} and y_{des} represents the desired location of the group, i.e. the recovery area, zoo_u and zoo_l represent the upper and lower boundaries of the orientation zone and ψ_{ref} is the desired heading angle of the group and is calculated using Equations (5.13)-(5.15).

Once the above waypoint has been calculated, the remaining, N-1 coordinates are evaluated using the following equations:

$$Y_{Waypoint_i} = Y_{Waypoint_1} + (w-1)\left(\frac{zoo_u + zoo_l}{2}\right)\cos(\psi_{ref}) \tag{7.3}$$

$$X_{Waypoint_i} = X_{Waypoint_1} - (w-1)\left(\frac{zoo_u + zoo_l}{2}\right)\sin(\psi_{ref}) \tag{7.4}$$

Here w represents the particular waypoint being calculated and ranges from 2 through to N.

Applying the above equations to a simulation similar to those used throughout Chapters 5 and 6 and plotting the waypoints generated produces the graph shown in Figure 7.4.

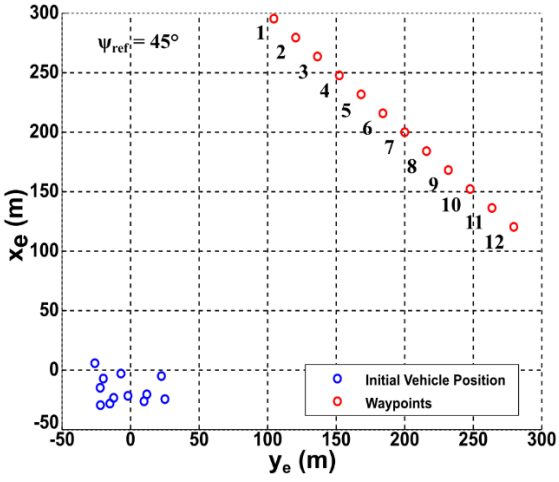


Figure 7.4- Illustration of waypoints generated using Waypoint Consensus Technique.

The graph presented above illustrates the initial positions of the group (blue circles) as well as the twelve equally spaced waypoints calculated using Equations (7.1)-(7.4) (red circles). Furthermore, the graph shows that the desired heading angle of the group has been set to 45º.

Furthermore, it is important to note that the waypoints presented in Figure 7.4 have been calculated without requiring any data from neighbouring vehicles. Instead, these waypoints have been calculated using the groups desired location, the number of vehicles in the group and the size of the orientation zone. As discussed above, all these parameters are mission specific and will be defined at the start of each mission.

Once each vehicle has calculated the various waypoints using Equations (7.1)-(7.4), they must subsequently decide which particular waypoint they are going to use and manoeuvre towards. This selection process and the functionality implemented within the algorithm to achieve it is discussed in the following section.

7.2.2.2 Creation of Waypoint Selection Process

The main aim of the formation control algorithm is to coordinate the group of vehicles into the parallel line formation presented in Figure 1.3. The particular selection process implemented within this waypoint consensus methodology has been designed specifically to ensure that this formation can be generated as quickly as possible. The only way to achieve this definitively is for each vehicle to select the waypoint that is analogous to its position within the group. Subsequently, applying this selection process to the vehicles presented in Figure 7.4 means that the vehicle positioned at the extreme left hand side of the group will select Waypoint 1. Thereafter, each subsequent vehicle positioned laterally to the right will select the next waypoint until the vehicle at the extreme right hand side of the group has selected Waypoint 12. Implementing this particular decision making process ensures that each vehicle travels the minimum distance possible before the required parallel line formation can be generated.

In order for each vehicle to select the most appropriate waypoint, it needs to be able to accurately evaluate the number of neighbouring vehicles positioned to its right and left hand side. As presented in Chapter 5 (Section 5.2.3.1), to achieve this the following two conditional statements are implemented.

$$NN_L = NN_L + 1 \quad ,if \quad -\pi \leq \psi_{R_{NN}} < 0 \tag{7.5}$$

$$NN_R = NN_R + 1 \quad ,if \quad 0 \leq \psi_{R_{NN}} \leq \pi \tag{7.6}$$

Importantly, and as shown below in Figure 7.5, the only way to ensure that accurate values can be obtained for the above parameters - regardless of the timeslot size implemented - is to ensure that every vehicle is moving with parallel trajectories. In the diagram presented below, the estimated position of the vehicles nearest neighbours can lie anywhere on the dashed lines and reflects the fact that due to the TDMA protocol, the estimated value will not be an accurate representation of the neighbouring vehicles actual position (green circle). Nevertheless, the diagram also demonstrates that due to the vehicles moving with parallel trajectories, this error will never affect the algorithm's ability to accurately determine how many of its nearest neighbours are positioned to its left or right hand side.

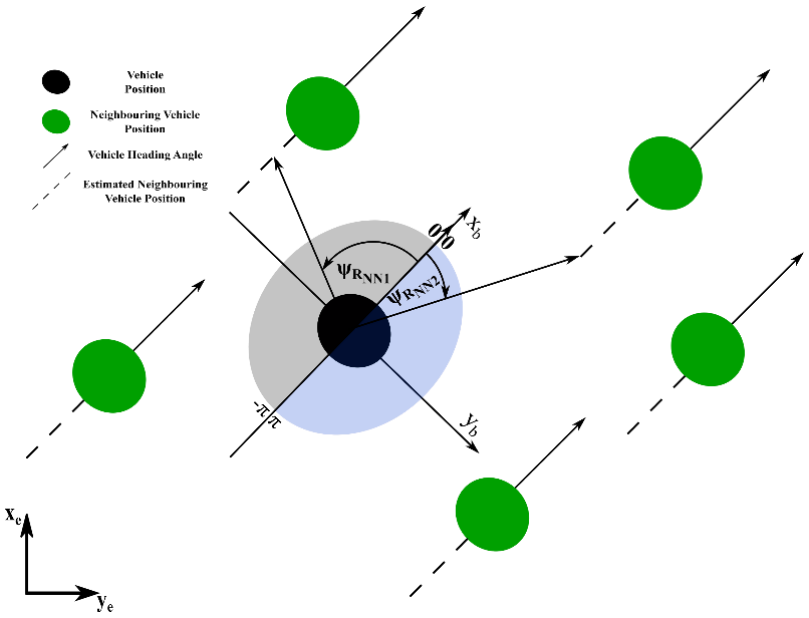


Figure 7.5- Diagrammatic representation of the evaluation of the relative positioning of neighbouring vehicles when vehicles are moving with parallel trajectories.

For this reason it has been decided that until each vehicle has selected its waypoint and consensus among the entire group has been achieved, every vehicle manoeuvres with a heading angle that is equal to group's desired heading angle, ψ_{ref}. . Once each vehicle has determined the number of neighbouring vehicles on either side, the selection of the vehicle's chosen waypoint can be calculated using Equation (7.7) below.

$$\text{Waypoint}_i = NN_L + 1 \tag{7.7}$$

The above equation simply states that each vehicle should select the waypoint that has a value one greater than the number of neighbouring vehicles positioned to its left hand side. Analysing this equation with the waypoints presented in Figure 7.4, it is apparent that this selection process assigns the waypoint analogous to each vehicles' position within the

group. Once each vehicle has selected its waypoint, it broadcasts this information to the rest of the group in order to prevent another vehicle from also selecting it.

7.2.2.3 Ensuring Group Consensus on Waypoint Selection

In order to prevent two vehicles from selecting the same waypoint, each vehicle, when selecting its chosen waypoint needs to determine whether or not it has been already selected by another member of the group.

To achieve this, the autopilot within each vehicle contains a column vector of size N by 1 which, at the start of each mission is populated by zeros. As the mission evolves, and the vehicles begin to receive communication updates from neighbouring vehicles, the rows relating to the vehicles that have already transmitted will become populated with their assigned waypoint. Thereafter, the vehicles still to broadcast are able to determine whether or not its chosen waypoint has already been selected. If it hasn't, the formation control algorithm prepares to broadcast its original chosen waypoint. If its chosen waypoint has already been selected, the methodology sequentially goes through the remaining unselected waypoints until the one closest to it's originally chosen one is selected. This process therefore guarantees that the same waypoint cannot be selected by more than one vehicle.

After each vehicle has successfully transmitted its preferred waypoint, the formation control algorithm (within each vehicle) becomes aware of this as the vector containing each vehicles waypoints will no longer contain any zeros. Once this has been established, the formation control algorithm will then begin manoeuvring each vehicle towards its desired waypoint.

7.2.2.4 Calculating Vehicle Heading Angle

As described in Chapter 4 (Section 4.3.2), the autonomous guidance of the RoboSalmon vehicle is achieved using a line-of-sight, waypoint following algorithm (Healey & Lienard 1993; Healey, A.J Marco 1992). This technique uses the difference in position between the vehicle and its desired waypoint to calculate the corresponding reference heading angle of the vehicle. However, as illustrated below in Figure 7.6., because the waypoints used in this work are positioned in the vehicle recovery area, the implementation of a line-of-sight algorithm would only result in the vehicles manoeuvring to a d_1 within the confines of the orientation zone at the end of the mission.

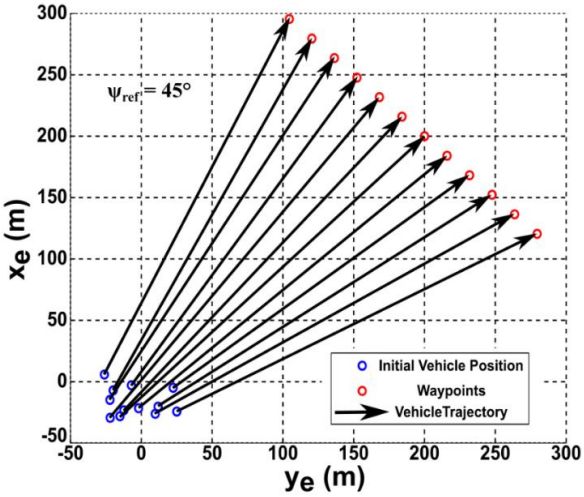


Figure 7.6- Diagrammatic representation of vehicle trajectory if standard line of sight algorithm where to be used.

In order to ensure the vehicles generate the required parallel line formation as quickly as possible, the control law shown below in Equation (7.8) has to be implemented instead.

$$\psi_D = \begin{cases} \psi_{ref} + \psi_{rep} & if \ \psi_{R_{Waypoint}} > 0 \\ \psi_{ref} - \psi_{rep} & if \ \psi_{R_{Waypoint}} < 0 \end{cases} \tag{7.8}$$

In the above equation, the angle, $\psi_{R_{Waypoint}}$ represents the angle between each vehicle's current location and its desired waypoint. As throughout this work, ψ_{ref} represents the desired heading angle of the group and ψ_{rep} represents the same user defined parameter presented in Chapter 5 (Section 5.2.5) which in this instance, dictates the rate at which each vehicle manoeuvres *laterally* towards its chosen waypoint. Finally, ψ_D is the vehicles desired heading angle and is used to calculate the necessary tail centre deflection angle of the vehicle. As the aim is to generate the required formation in the shortest period of time, this parameter, ψ_{rep} as in Chapter 5, is set to 90º when the *repulsive* control law is implemented and 45º when the *attractive* control law is being used.

The above equation demonstrates that if a vehicle's chosen waypoint is positioned to its right hand side, i.e. $\psi_{R_{Waypoint}} > 0$, the vehicle will manoeuvre to the right at an angle of either 45º or 90º from the reference heading angle, ψ_{ref}. Conversely, if a vehicles chosen waypoint is positioned to its left hand side, i.e. $\psi_{R_{Waypoint}} < 0$, the vehicle will manoeuvre to the left at an angle of 45º or 90º from ψ_{ref}.

Analysing Equation (7.8), it's apparent that its design is analogous to the *attractive* and *repulsive* control laws presented in Chapter 5 with the only difference being that Equation

(7.8) uses the relative position of the vehicles waypoint as opposed to its nearest neighbour's position when deciding which particular angle to move with.

Nevertheless, as with all waypoint tracking manoeuvres, the control law presented above will ensure the vehicles continue to manoeuvre in the required direction. For the work completed in this section, this will remain the case until the *lateral* distance between the vehicle and its chosen waypoint decreases to a value that is below a certain threshold - commonly referred to as the *acceptance radius*.

When this occurs, the autopilot usually provides either another waypoint for the vehicle to manoeuvre towards or a particular desired heading angle for it to move with. For the waypoint consensus algorithm, it was decided that once a vehicles *lateral* distance to its chosen waypoint is below the *acceptance radius*, the vehicle will thereafter move with a heading angle equal to the reference heading angle, ψ_{ref}, i.e. the desired heading angle of the group.

7.2.2.5 Algorithm Structure

Sections 7.2.2.1-4 above have described the waypoint consensus formation control algorithm. As discussed previously, this algorithm has been designed specifically to ensure that a group of vehicles, regardless of the nuances of the TDMA protocol, is capable of generating the required parallel line formation. As with the modified formation control algorithm presented in Chapter 5, this algorithm must go through a number of decision making processes before it is able to select the vehicles desired heading angle. This decision making processes and associated calculations executed during the implementation of this algorithm are illustrated in Figure 7.7.

The flow diagram presented below illustrates that no vehicle selects a heading angle other than ψ_{ref} until the entire group of vehicles has selected a unique waypoint. Also apparent from the flow diagram presented above is the fact that so far, no functionality has been included to ensure each vehicle, as required, will converge to an *absolute* nearest neighbour distance that is within the confines of the orientation zone. As the convergence of this parameter is one of the original algorithmic requirements presented in Chapter 4, the following section will now describe the functionality implemented within this consensus based algorithm to ensure that this criteria can be satisfied.

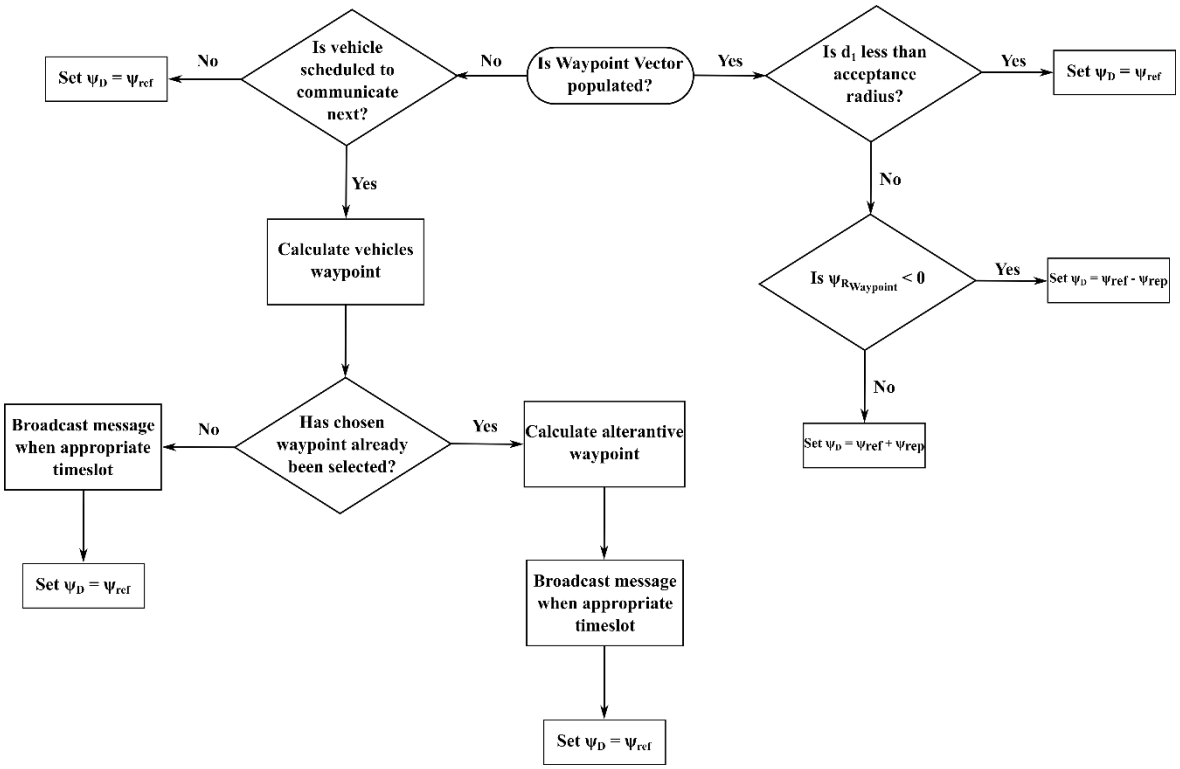


Figure 7.7 - Decision making processes evaluated within the waypoint consensus algorithm.

7.2.2.6 Velocity Control Law

As discussed in Chapter 5, the only way to guarantee convergence to the required *absolute* nearest neighbour distance is through the introduction of a suitable velocity control law. While the results presented in Chapter 5 illustrate that the velocity control law is capable of ensuring convergence when instantaneous communication is assumed, the results presented in Chapter 6 demonstrate that this is no longer the case once a realistic communication protocol is taken into consideration. Furthermore, as discussed in Section 6.5.1.5, the main reason for this non-convergence is due to the inherent cyclical nature of the TDMA protocol causing the algorithm to wrongly estimate that each vehicles nearest neighbour is out with the confines of the zone of orientation.

Although the consensus based formation control algorithm presented above is intended to use as little data from neighbouring vehicles as possible, it is impossible not to when trying to satisfy this final convergence criteria. Nevertheless, a number of precautions are taken to ensure that any alterations made to each vehicle's velocity is done so with the most accurate data available. The decision making process and functionality contained within the velocity control law to achieve this is shown below in Figure 7.8.

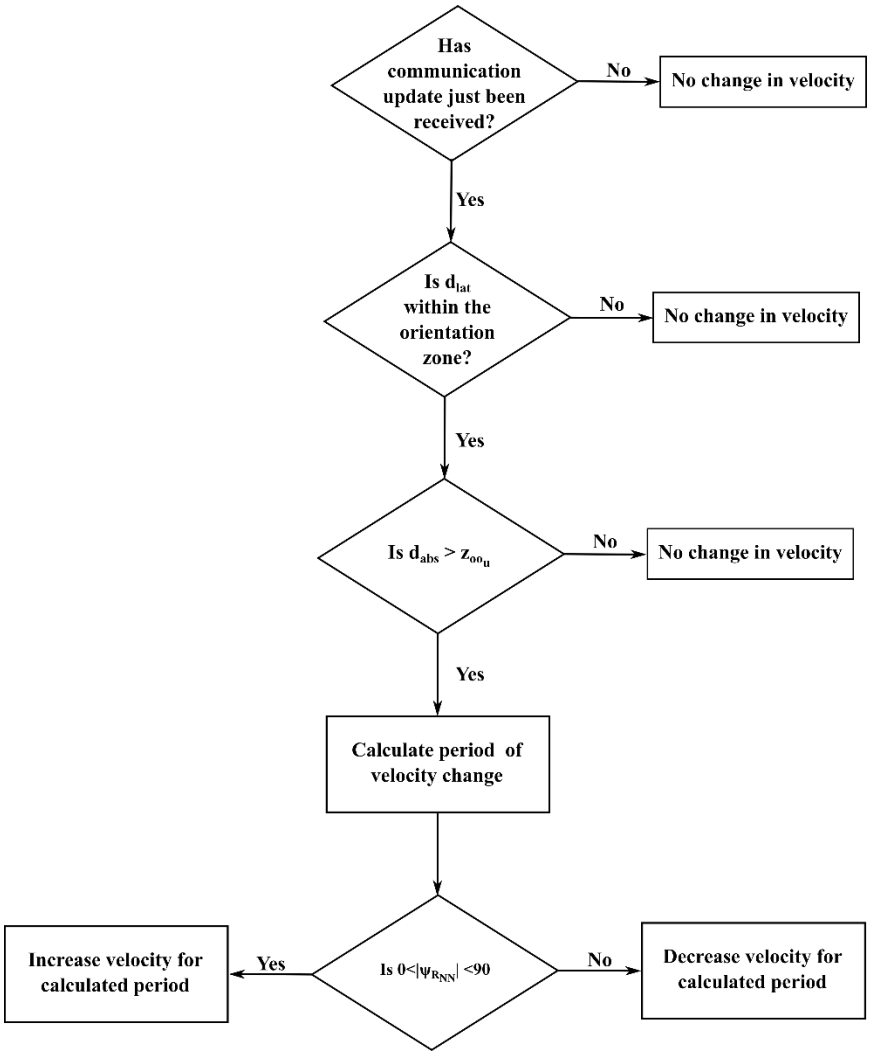


Figure 7.8- Decision making process and functionality contained within the velocity control law of the waypoint consensus algorithm.

The most notable change to this flow diagram is that the each vehicle's velocity is only altered once a communication update has been received from its neighbouring vehicle. As mentioned above, the reason for implementing this particular restriction is to ensure that the formation control algorithm is making velocity alterations based on the most accurate representation of its neighbouring vehicle's position.

It is also apparent that only after an additional three criteria have been satisfied will the algorithm decide to alter the vehicles velocity. The first of these criteria establishes whether or not the *lateral* distance, $d_{lat,}$ to the vehicle currently broadcasting is within the confines of the orientation zone. If it is not, then as the diagram illustrates, the vehicle maintains its current velocity. If, on the other hand, the distance is within the confines of the orientation zone, the algorithm then decides whether or not the associated *absolute* distance is greater than the upper boundary of the orientation zone. Once again, if this criteria is not satisfied, the algorithm does not make any changes to the vehicles velocity. Conversely, if this criteria

is met, then the final decision to be made is to establish whether the associated neighbouring vehicle is positioned to the front or behind. Intuitively, if the neighbouring vehicle is positioned to the front, the algorithm increases the vehicle's velocity and conversely, the velocity control law will reduce the vehicle's velocity if the neighbouring vehicles is positioned to the rear.

In order to prevent the vehicle from continuously altering its velocity as shown in Chapter 6, Figure 7.8 shows that once each vehicle has decided it needs to alter its velocity, the algorithm calculates the period of time the vehicle should alter its velocity for. As shown below in Equation (7.9), this period of time is calculated using the relative velocity of the two vehicles.

$$\mathrm{t}_{abs} = \frac{\mathrm{d}_{abs} - zoo_u}{\vec{\mathrm{v}}_{i|NN}} \tag{7.9}$$

Here, $\mathrm{d}_{\mathrm{abs}}$ represents the *absolute* distance between the two vehicles, zoo_u is the upper boundary of the orientation zone, $\vec{\mathrm{v}}_{i|NN}$ is the relative velocity of the two vehicles.

Due to Equation (7.9) being evaluated when the formation control algorithm has the most accurate representation of the distance, d_{abs}, the value calculated using Equation (7.9) will represent the time taken for the vehicle – based on its selected velocity - to reduce the distance, $\mathrm{d}_{\mathrm{abs}}$ to within the confines of the orientation zone. However, it is important to note that one final assumption has been made with regard to the value used for the relative velocity, $\vec{\mathrm{v}}_{i|NN}$. This assumption is that regardless of the actual relative velocity between the two vehicles, the value used within Equation (7.9) will always represent the maximum possible relative velocity. This is to prevent the scenario arising whereby two vehicles – unbeknown to one another –are moving towards each other at a greater rate than expected. Consequently, this behaviour could result in the two vehicles having already altered their velocity according to Figure 7.8 still having a $\mathrm{d}_{\mathrm{abs}}$ that is out with the boundary of the orientation zone. Therefore, by always using the maximum possible value for $\vec{\mathrm{v}}_{i|NN}$ it is unlikely that this situation will arise.

However, a consequence of designing the velocity control law in this manner is that if the neighbouring vehicles are not manoeuvring towards one another at the maximum relative velocity, the value calculated for t_{abs} will be insufficient to ensure the vehicle converges to within the confines of the orientation zone. As a result, the algorithm will have to wait an entire communication cycle before it can alter its velocity again.

The functionality described above has presented two different methodologies designed specifically to overcome the nuances associated with the incorporation of the TDMA protocol as presented in Chapter 6. The following section will now present and analyse the simulation results obtained from the incorporation of these methodologies.

7.3 Simulation Setup & Results

As discussed above, the main aim of this chapter is to demonstrate whether or not the different methodologies presented in Section 7.2 operate as expected and as a result, establish whether the algorithm is now capable of satisfying the various requirements presented previously in Table 5.3 regardless of the timeslot size implemented. In order to definitively demonstrate that this is the case, it is important that the simulations used are identical to those in Chapter 6. The following section will provide a brief review of these simulations and highlight any differences incorporated within this analysis. This chapter will also compare the results from the two methodologies in an effort to ascertain whether or not one is more efficient than the other.

7.3.1 Simulation Setup

As presented in Sections 5.3 and 6.4, the simulations undertaken thus far have been separated into two groups depending on the particular control law initially implemented by the majority of the vehicles, i.e. either the *repulsive* or *attractive* control law. As this chapter is focussing on testing and analysing whether or not the techniques described above operate as envisioned, the results presented in this chapter can no longer be categorised in this manner. Instead, they are defined with relation to the particular technique implemented to overcome the effects of incorporating the TDMA protocol. Nevertheless, the results presented in this section have been obtained from the exact same simulations used throughout Chapter 6. As shown below in Table 7.1, the only difference in this case is that the results contained within each group now contain data from simulations that use two different sets of behavioural zones.

The values used for the algorithmic parameters, $T_{Timeslot}$, T_{Tot} and ψ_{ref} remain unchanged from those used in Chapter 6. Furthermore, due to the fact that the two groups of simulations undertaken in Chapter 6 have now been combined, the newly created groups shown in Table 7.1 are now composed of 200 unique simulations, i.e. 100 simulations initially using the *repulsive* control law and 100 simulations initially using the *attractive* control law. Due to these simulations been repeated for each timeslot size presented, the results presented below are composed of data obtained from 2400 simulations.

Table 7.1 - Variation in simulation parameters.

Parameter	Group 1	Group 2
Technique Implemented	Trajectory Prediction	Waypoint Consensus
Behavioural Zones Size (m)	[20,25,200]m & [6,11,200]m	
Timeslot Size ($T_{Timeslot}$)	[1, 2, 4, 8, 16, 24]s	
Transmission & Propagation Delay (T_{Tot})	0.205s	
Reference Heading Angle (ψ_{ref})	$\frac{\psi}{4}$	

7.3.2 Simulation Performance Metrics

In order to demonstrate whether or not the techniques presented above have been successful, the initial results presented will simply illustrate the evolution of the percentage of vehicles satisfying the four convergence criteria presented below in Table 7.2 as the timeslot size is increased.

Table 7.2 - Algorithm convergence criteria.

Parameter	Convergence Criteria
Standard Deviation of Vehicle Heading Angle - σ	$\sigma \leq 1^{o}$
Lateral Distance to Nearest Neighbour – d_1	$zoo_l \leq d_1 \leq zoo_u$
Perpendicular Distance to Second Nearest Neighbour – d_2	For N-2 vehicles: $zoo_l \leq d_2 \leq zoo_u$ For 17% of vehicles: $2(zoo_l) \leq d_2 \leq 2(zoo_u)$
Absolute Nearest Neighbour Distance - d_{abs}	$zoo_l \leq d_{abs} \leq zoo_u$

As discussed throughout this thesis, if the four criteria presented abvove are all simultaneously satisfied, the formation control algorithm has not only guaranteed to have

coordinated the vehicles into the required parallel line formation but it has also ensured each vehicle has converged to the required *absolute* nearest neighbour distance. Therefore, presenting the evolution of these percentages provides the simplest and most efficient way of demonstrating whether or not the two methodologies described in Section 7.2 operate as envisioned.

Once the results analysing the ability of the above techniques to satisfy the necessary criteria have been presented, the results will then focus on determining whether or not one of the techniques is more efficient than the other. In order to achieve this and as shown below in Table 7.3, a number of performance metrics have been evaluated from simulations and the results obtained will be compared.

Table 7.3 - Algorithmic performance metrics.

Performance Metric	**Description**
Formation Convergence Time	This parameter is calculated to ascertain the difference in the time taken for the two methodologies to ensure every convergence criteria in Table 7.2 is satisfied.
Processing Resources Required to Implement Specific Methodology	Calculating this particular metric provides a method by which to directly compare the increase in computational resource incurred by each vehicle in order to implement the methodologies described above. The only way to measure this metric is by calculating the processing time required.

As well as presenting the metrics described above and the evolution of the percentage of vehicles satisfying the necessary criteria, the results will also present the trajectories obtained from the same simulations as presented throughout Chapter 6. This is to present a direct comparison of the variation in trajectories obtained from the implementation of the two techniques described in this chapter.

7.4 Results

The aim of this section is to analyse the results obtained from the simulations described above and demonstrate whether or not the two methodologies implemented have been successful in ensuring the necessary convergence criteria are satisfied. The results presented will also analyse a number of performance metrics associated with each of these methodologies in an effort to ascertain whether or not one particular methodology is more efficient than the other.

7.4.1 Analysing the Convergence Ability of Methodologies

As discussed above, to demonstrate whether or not the methodologies implemented have been successful, the evolution of the percentage of vehicles satisfying the necessary criteria are to be presented. Subsequently, shown below in Figure 7.9 is the evolution of these percentages for the simulations completed when the trajectory prediction methodology of Section 7.2.1 was implemented.

The results demonstrate that when the timeslot size is equal to 24s, every vehicle is able to satisfy each of the necessary convergence criteria described in Table 7.2. It is important to note that while each of the graphs in Figure 7.9 appear to only include the results obtained from one timeslot size (i.e. 24s), the graphs do in fact include the data from the remaining timeslot sizes. The reason that this data is indistinguishable is due to the fact that as described above, the trajectory prediction methodology has been designed to always provide each vehicle with an accurate representation of its nearest neighbour's position regardless of the timeslot size implemented. The fact that the results presented below demonstrate that the results are identical suggest that the prediction methodology is operating as expected. It is also apparent that similarly to the results presented in Chapter 5, every vehicle is able satisfy the necessary criteria in less than 600s. Therefore, when combined, these factors imply that as envisioned, the formation control algorithm is operating as if communication among the vehicles is instantaneous.

Once again, it is important to note that the results presented in Figure 7.9 have been obtained from the first group of simulations discussed in Table 7.1 and as a result, they represent the data obtained from the exact same simulations analysed throughout Chapter 6.

Based on the results presented below, it can be concluded that for the algorithmic parameters presented in Table 7.1, the trajectory prediction methodology is capable of ensuring that each vehicle is able to satisfy the necessary criteria that result in the vehicles generating the required parallel line formation.

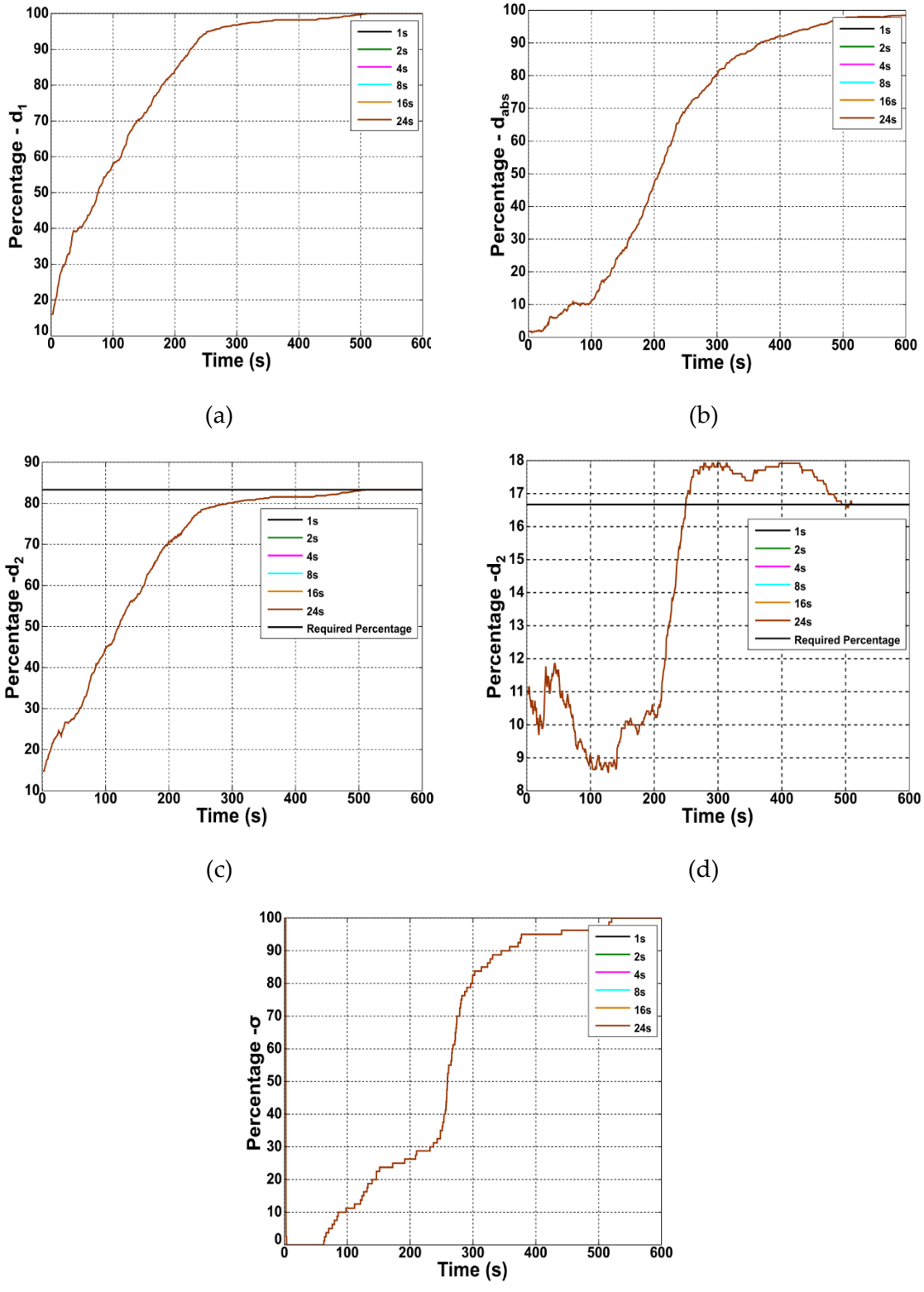


Figure 7.9 - Evolution of the percentage of vehicles satisfying the various convergence criteria when the predictive functionality is incorporated.

The results presented above also clearly demonstrate that the ability of this predictive functionality is not adversely affected by an increase in the timeslot size used. Moreover,

due to the results presented above being identical for each timeslot size, it is expected that the associated trajectories obtained for the same initial conditions will also be identical. This is indeed shown to be the case below in Figure 7.10.

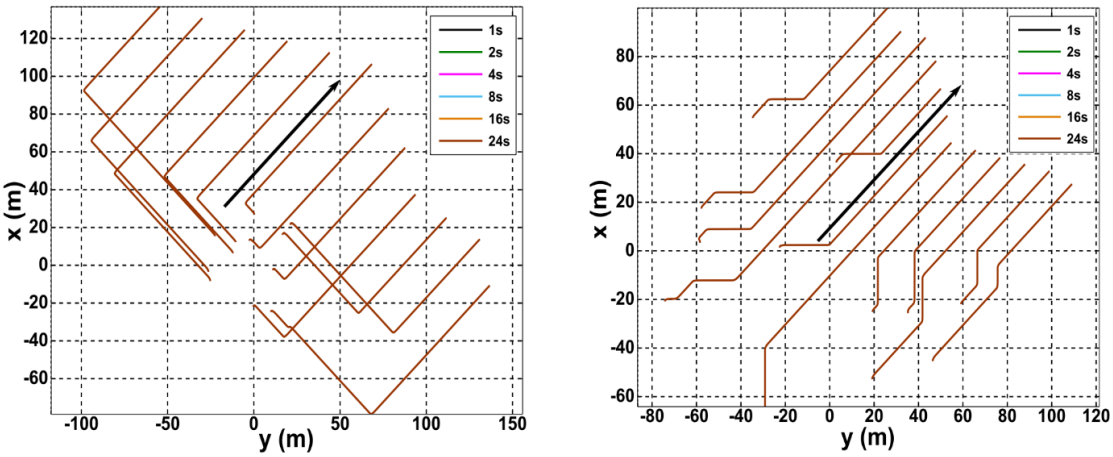


Figure 7.10- Variation in group trajectory as timeslot size is increased when vehicles are initially using (a) repulsive control law and (b) attractive control law when trajectory prediction method is used.

In Figure 7.10 (a) the trajectories presented have been obtained from a simulation where initially, the majority of vehicles are using the *repulsive* control law. Conversely, Figure 7.10 (b) illustrates the trajectories obtained when the majority of vehicles are initially using the *attractive* control law. Regardless, both figures clearly illustrate that the formation control algorithm - with the incorporation of the predictive functionality - has been able to coordinate the vehicles into the required parallel line formation. It is important to note that although the trajectories presented above have only been obtained from two of the 1200 simulations completed throughout this study, the previous results presented in Figure 7.9 demonstrate the trajectories presented above are nevertheless representative of the results obtained from the remaining 1188 simulations.

While the results presented above demonstrate that the predictive functionality operates as envisioned, the results obtained for the waypoint consensus methodology are still to be analysed. Subsequently, similarly to the results presented in Figure 7.9, shown below in Figure 7.11 are the evolutions of the percentage of vehicles satisfying the necessary criteria when the waypoint consensus methodology is incorporated within the algorithm.

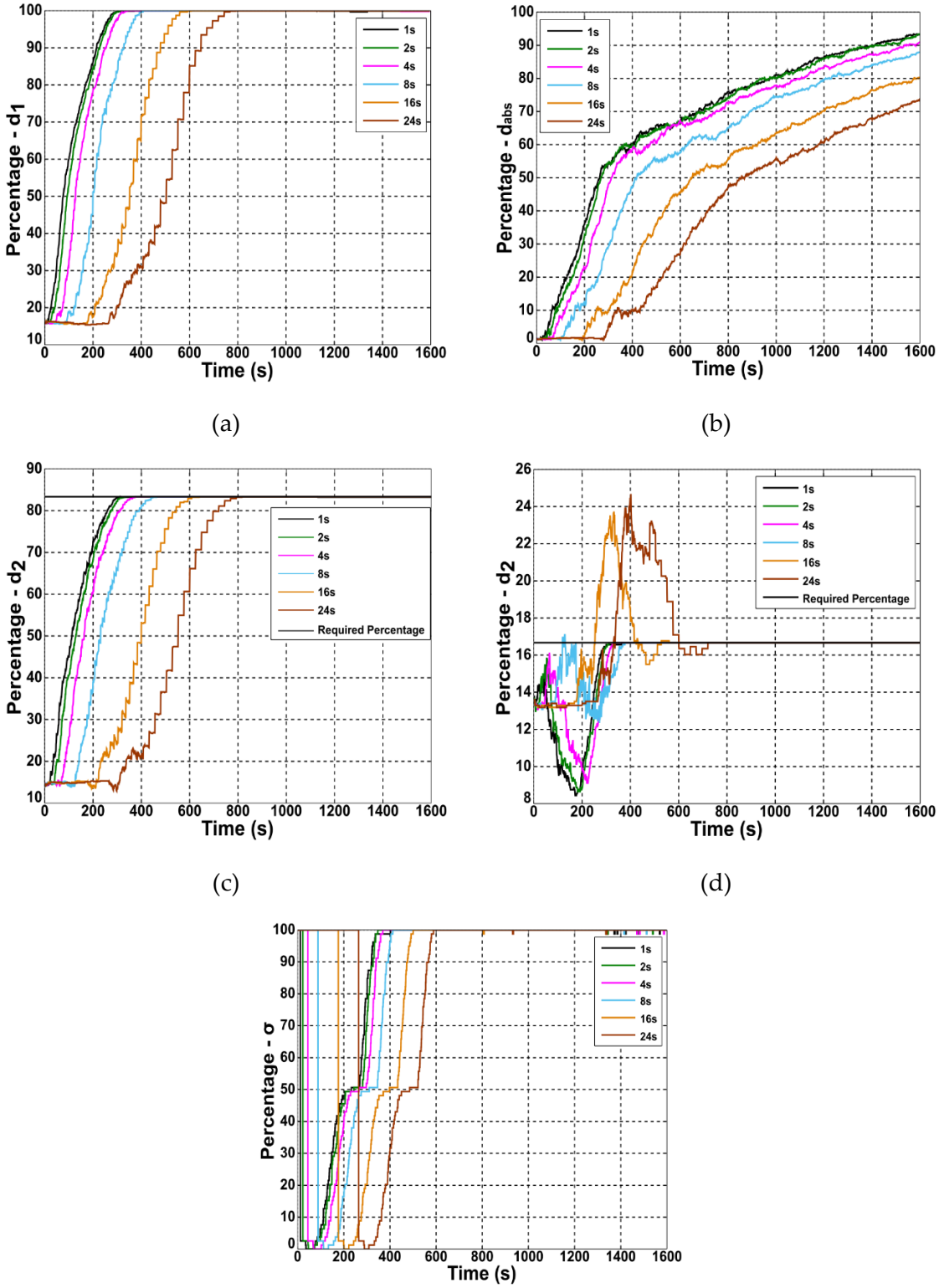


Figure 7.11- Evolution of the percentage of vehicles satisfying the various convergence criteria when the waypoint consensus methodology is incorporated.

Immediately, it is apparent that unlike the results associated with the predictive functionality, the convergence performance of the waypoint consensus technique varies as

the timeslot size is increased. For the majority of the convergence criteria, this variation results in a steady increase in the time taken for the required criteria to be satisfied.

However, these results are not surprising and are in fact evidence that the waypoint consensus algorithm is operating as expected. This is due to the fact that as shown in Figure 7.7 and discussed throughout Section 7.2.2, this method can only begin manoeuvring the vehicles into the required formation once it has been agreed that every vehicle has chosen a unique waypoint. Due to the nature of the TDMA protocol, this consensus is only achieved (at best) after one complete communication cycle. Therefore, with the communication cycle length increasing with the timeslot size implemented, the time taken for the individual vehicles to begin manoeuvring in the required direction will also inherently increase. This behaviour is reflected throughout the results presented in Figure 7.11 where it is apparent that as the timeslot size is increased, the time taken for the percentage values to begin to converge, gradually increases with the timeslot size.

Nevertheless, the results demonstrate that once every vehicle has chosen a unique waypoint, the algorithm is capable of coordinating the vehicles into a formation that ensures that four out of the five necessary convergence criteria are satisfied. As Figure 7.11 (b) demonstrates, the exception to this is the criteria associated with each vehicle's *absolute* nearest neighbour distance. The results demonstrate that the number of vehicles able to satisfy this criteria reduces considerably as the timeslot size is increased.

This inability to converge within the given timeframe is due to each vehicle only being permitted to alter its velocity when it has an accurate representation of the current positioning of its nearest neighbour, i.e. once every communication cycle. As discussed in Section 7.2.2.6 and shown in Equation (7.9), thereafter, the amount of time each vehicle is able to alter its velocity for has also been deliberately underestimated. This underestimation was incorporated to ensure every vehicle would eventually be capable of converging to the required *absolute* nearest neighbour distance. These two factors when combined explain why the waypoint consensus methodology is unable (within the simulation timeframe) to ensure every vehicle is able to satisfy the criteria associated with this parameter.

Nevertheless, it is important to note that while the evolution of the percentage values associated with d_{abs} have not converged, they continue to steadily increase towards the end of the simulations. This implies that while the design of the velocity control law could be improved, in its current guise, it is still capable of ensuring that a group of vehicles are able to satisfy this convergence criteria. This is shown below in Figure 7.12.

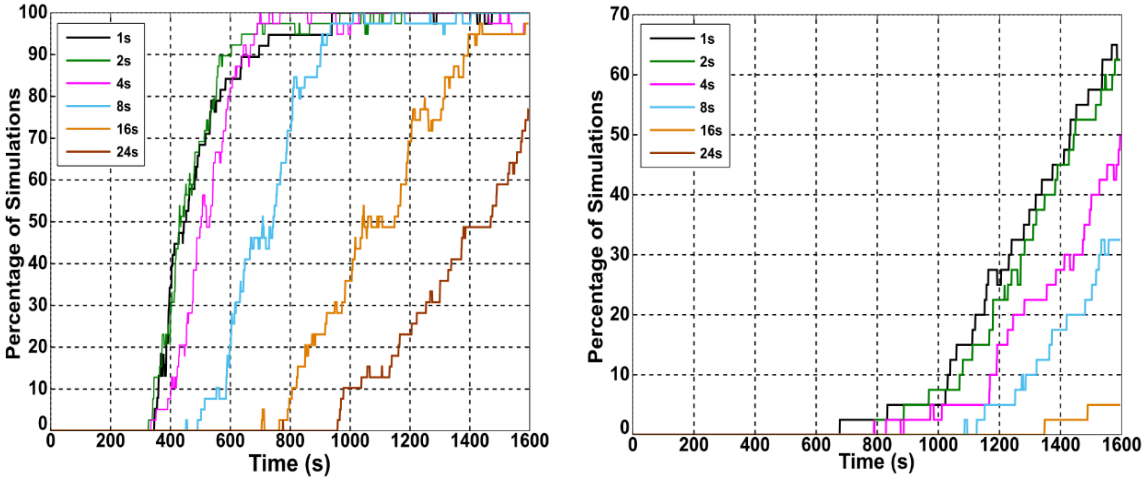


Figure 7.12 - Variation in the evolution of the percentage of simulations satisfying the convergence criteria associated with d_{abs} when vehicles are initially using (a) repulsive control law and (b) attractive control law.

The results presented in Figure 7.12 (a) represent the evolution of the percentage of simulations satisfying the criteria associated with d_{abs} when the vehicles are initially using the *repulsive* control law. Similarly, Figure 7.12 (b) presents the same results but for the simulations whereby the vehicles are initially using the *attractive* control law. Regardless, the results, particularly those in Figure 7.12 (a), demonstrate that even when the timeslot size is equal to 24s, the consensus waypoint methodology has created a group formation that satisfies this criteria.

The reason for the discrepancy in the results presented above is due to the inherent difference in the relative positioning of neighbouring vehicles when the *attractive* and *repulsive* control laws are used. As shown below in Table 7.4, on average, vehicles that have initially used the *attractive* control law are required to travel a much greater distance to ensure their d_{abs} value converges to within the confines of the orientation zone.

Combining this greater distance with the fact that the parameter, t_{abs} is deliberately underestimated, it is easy to comprehend that each vehicle will require multiple communication updates from its nearest neighbour before it can converge to an *absolute* nearest neighbour distance value that is within the confines of the orientation zone. As discussed above, due to the inherent increase in the communication cycle that results from increasing the timeslot size, this process will take significantly longer as the timeslot size is increased.

Table 7.4 - Variation in the average distance each vehicle is required to travel to ensure convergence of the parameter, d_{abs}.

Initial Behaviour	
Repulsive	**Attractive**
2.6m	38m

Overall, when the results presented in Figures 7.11 and 7.12 are combined, it can be summarised that the waypoint consensus algorithm is capable of coordinating the vehicles into the required parallel line formation. However, the results also demonstrate that the time taken to ensure each vehicle converges to the required *absolute* nearest neighbour distance is dependent on not only the timeslot size implemented but also the initial *absolute* distance between neighbouring vehicles. This behaviour is reflected in the trajectories presented below in Figures 7.13 and 7.14.

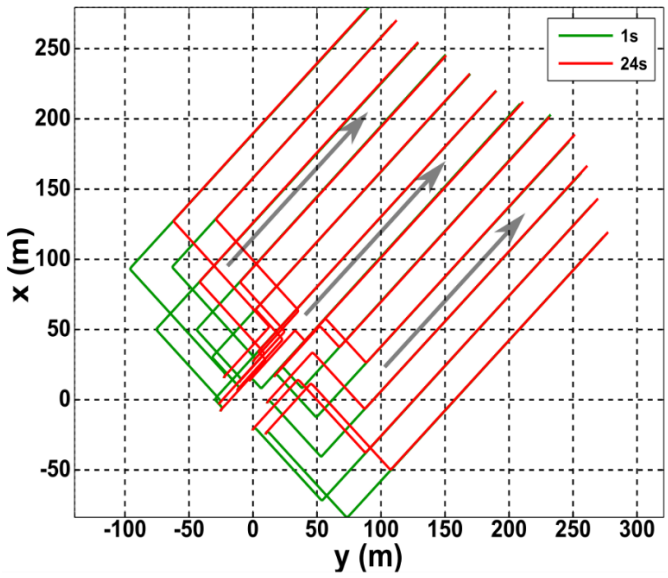
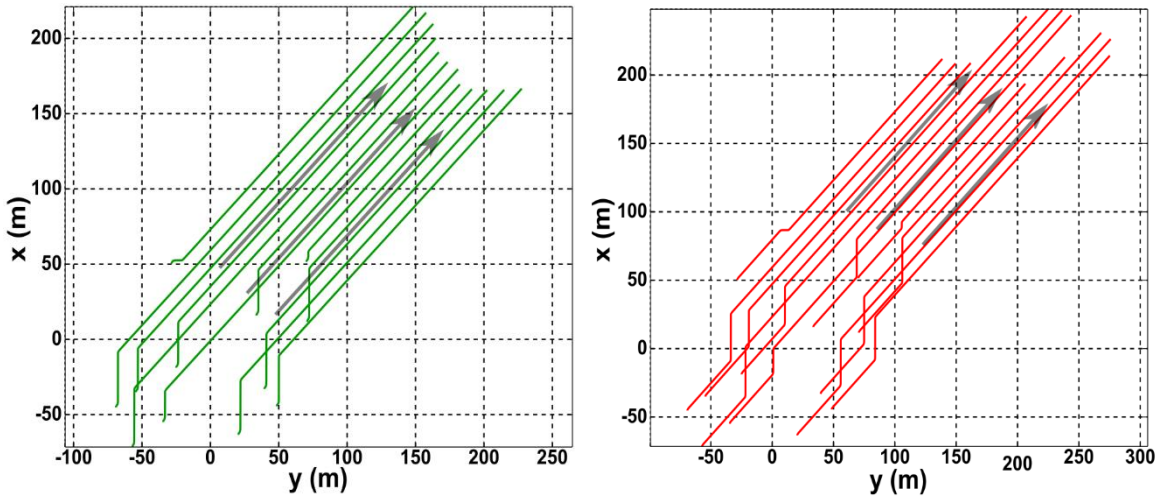


Figure 7.13- Variation in group trajectory as timeslot size is increased and vehicles are initially using the repulsive control law and the waypoint consensus algorithm.

As expected, the trajectories presented above illustrate that regardless of the timeslot size implemented, the algorithm has been able to not only generate the required parallel line formation but also ensure that neighbouring vehicles are positioned directly alongside one another.

Conversely, as shown below in Figure 7.14, when the vehicles are initially using the *attractive* control law and the initial *absolute* distance between neighbouring vehicles is

significantly greater, the resulting trajectories obtained vary depending on the timeslot size implemented.



(a) (b)

Figure 7.14 - Example of group trajectory obtained when vehicles are initially using attractive control law and timeslot size is equal to (a) 1s and (b) 24s.

Comparing the trajectories presented above in Figure 7.14 it is apparent that while both scenarios have resulted in the generation of the required parallel line formation, the ability of the algorithm to ensure each vehicle converges to the required *absolute* nearest neighbour distance (within the same time frame) deteriorates as the timeslot size is increased. This, as discussed above is due to the fact that when the timeslot size is increased, each vehicle has less opportunities to alter its velocity.

In summarising, the results presented above demonstrate that both the trajectory prediction and waypoint consensus methodologies operate as envisioned and as a result, are capable of ensuring that the required parallel line formation can be generated regardless of the timeslot size implemented. In this respect, both methodologies significantly improve the ability of the algorithm to cope with the nuances associated with the underwater communication channel. It is important to note though that with respect to the ability of the two methodologies to ensure the parameter, d_{abs} converges, the results clearly illustrate that the predictive methodology provides the more efficient solution.

As well as ensuring that every vehicle is able to satisfy the criteria associated with the parameter, $d_{abs,}$ the results presented in Figures 7.9 and 7.11 also illustrate that the trajectory prediction methodology is able to satisfy the four convergence criteria presented in Table 7.2 faster. This is demonstrated below in Table 7.5 where the mean, maximum and minimum convergence times associated with the trajectory prediction and waypoint

consensus methodologies are presented. Before analysing the results though, it is important to note that the values presented in Table 7.5 represent the overall average, minimum and maximum convergence times obtained from the 200 simulations completed for each timeslot size. This means that the values presented represent the entire envelope of convergence times for each timeslot. Importantly, as shown in Figure 7.12, not all the simulations completed using the waypoint consensus technique were are able to satisfy the necessary criteria and as a result, the convergence times presented for this particular methodology have only been taken from those simulations that did satisfy the necessary criteria. In order to obtain the values presented below, the convergence time of each simulation was taken to be the time each simulation satisfied the criteria presented in Table 7.2

Table 7.5 - Comparison of the minimum, mean and maximum convergence times obtained from the waypoint consensus and trajectory prediction methodologies.

Timeslot Size (s)	**Minimum Time (s)**		**Mean Time (s)**		**Maximum Time (s)**	
	Waypoint	Prediction	Waypoint	Prediction	Waypoint	Prediction
1	347	207	788	363	1569	590
2	328	207	778	363	1579	590
4	336	207	785	363	1591	590
8	453	207	853	363	1581	590
16	709	207	1064	363	1490	590
24	777	207	1298	363	1596	590

The results presented above clearly demonstrate that when the trajectory prediction functionality is implemented, the time taken for the formation control algorithm to satisfy the necessary criteria is significantly less when compared with the waypoint consensus methodology. Furthermore, the results also demonstrate that the disparity in the convergence time between the two methodologies increases significantly with the timeslot size. As discussed previously, this is due to the design of the velocity control law within the waypoint consensus algorithm and the fact that each vehicle can only alter its velocity once every communication cycle.

Lastly, the results also demonstrate that when the trajectory prediction functionality is incorporated the values obtained for the minimum, mean and maximum convergence times

are identical regardless of the timeslot size used. While initially this seems unlikely, the trajectory prediction methodology has been designed to provide each vehicle with an accurate representation of the positioning and orientation of neighbouring vehicles as if the transfer of data among the vehicles was instantaneous. As a result, this means that the formation control algorithm with the trajectory prediction methodology should produce the exact same results regardless of the timeslot size implemented.

In terms of comparing the efficiency of the two methodologies, the results clearly demonstrate that the formation control algorithm incorporating the trajectory prediction functionality achieves convergence approximately 2-3.5 times faster than the waypoint consensus technique. If the convergence time were to be the only metric to be taken into consideration, it is abundantly clear from these results that the trajectory prediction methodology is the most efficient methodology. However, before it is possible to make a definitive conclusion as to which methodology is the most efficient, the additional computer resource required to implement these two methodologies has to be analysed.

To quantify this additional computational effort, the amount of processing time required (in milliseconds) to implement these two methodologies has been evaluated and the minimum, mean and maximum values obtained presented below in Table 7.6.

The results clearly demonstrate that the processing time required for the trajectory prediction methodology is over an order of magnitude greater than that of the waypoint consensus algorithm. While significant, this difference is not unexpected and is due to the fact that the trajectory prediction functionality not only contains an identical copy of the RoboSalmon mathematical model but it is evaluated for each neighbouring vehicle (i.e. N-1) at a rate of 200Hz. This method is considerably more complex than that contained within the waypoint consensus technique where only the *lateral* distance to each vehicles chosen waypoint and the *absolute* distance to its nearest neighbour are required to be evaluated. Therefore, the results presented below clearly demonstrate that in terms of the computational effort required, the waypoint consensus technique is more efficient.

However, Table 7.6 nevertheless illustrates that in order to provide each vehicle with a continuously accurate representation of the positioning of every other vehicle within the group, the required processing time, on average, is only 6.1ms. While in isolation, this value may seem significant, the results presented in Table 7.5 and Figure 7.9 demonstrate that not only does implementing this technique ensure the vehicles will generate the required parallel line formation but it will do so 2-3.5 times faster than the waypoint consensus algorithm. The results also demonstrate that unlike the waypoint consensus methodology,

the trajectory prediction algorithm is unaffected by increases in the timeslot size being used. Therefore, the results presented suggest that in order to overcome the nuances associated with the implementation of the TDMA protocol as efficiently as possible, the trajectory prediction functionality presented in Section 7.2.1 should be incorporated.

Table 7.6 - Comparison of the processing time required to evaluate the functionality contained within the trajectory prediction and waypoint consensus methodologies.

Timeslot Size (s)	**Minimum Time (ms)**		**Mean Time (ms)**		**Maximum Time (ms)**	
	Waypoint	Prediction	Waypoint	Prediction	Waypoint	Prediction
1	0.4	5.4	0.5	6.3	0.7	6.6
2	0.4	5.4	0.5	6.3	0.7	6.6
4	0.4	5	0.5	5.8	0.6	6.6
8	0.4	5.8	0.5	6.3	0.6	6.6
16	0.4	5.8	0.5	6.3	0.6	6.6
24	0.4	5.4	0.5	5.8	0.6	6.6

7.5 Summary

The purpose of this chapter has been to describe the implementation of two methodologies designed specifically to overcome the nuances associated with the TDMA protocol and in doing so, ensure the required parallel line formation can be generated as quickly and efficiently as possible. After the functionality contained within these two methodologies was described in Sections 7.2.1 and 7.2.2, the analysis completed within Section 7.4.1 began by establishing whether or not the additional functionality implemented did indeed ensure the required parallel line formation could be generated. Thereafter, the analysis focussed on establishing whether or not one of the methodologies is particularly more efficient (and hence more suitable) than the other. Consequently, presented below is a brief summary of the pertinent results obtained from this analysis.

- The results presented within Figure 7.9 demonstrate that regardless of the timeslot size used, the formation control algorithm with the trajectory prediction methodology incorporated satisfied the varies convergence criteria presented in Table 7.2. Furthermore, the results also demonstrate that the algorithm achieves convergence as if the communication amongst the vehicle is instantaneous.

- Conversely, the results presented in Figure 7.11 demonstrate that while the waypoint consensus technique is able to generate the required parallel line formation due to the design of the modified velocity control law (Section 7.2.2.6), the ability of the vehicles to converge to the required *absolute* nearest neighbour distance deteriorates as the timeslot size is increased.
- Furthermore, unlike the results presented in Figure 7.9, Figure 7.11 demonstrates that due to its design, the convergence time of the waypoint consensus algorithm increases proportionally with the timeslot size used.
- In terms of comparing the convergence performance of the two methodologies, the results presented in Table 7.5 illustrate that on average, the trajectory prediction method satisfies the varies convergence criteria 2-3.5 times faster than the waypoint consensus method.
- Finally, while unsurprising, the results presented in Table 7.6 demonstrate that due to its increased complexity, the trajectory prediction methodology requires significantly more processing time in order to operate when compared with the waypoint consensus technique.

In concluding, this chapter has presented and analysed the implementation and associated performance of two methodologies designed specifically to overcome the nuances associated with the TDMA protocol. While the results have illustrated that both methodologies operate as envisioned, they also suggest that in terms of overall performance, the trajectory prediction methodology would seem to provide an overall better solution.

However, it is important to note that the results presented in this chapter, have not taken into consideration the effect of external disturbances such as obstacles and oceanic currents. Subsequently, the aim of the following chapter will now be to incorporate a realistic representation of these external disturbances and investigate what effect they have on the performance and efficiency of the two methodologies presented in this chapter.

Chapter 8

External Disturbances

8.1 Introduction

In the previous chapter, the results demonstrated that both the trajectory prediction and waypoint consensus techniques were able to successfully coordinate the vehicles as required despite the nuances associated with the underwater communication channel. However, while both techniques ensured convergence, they also illustrated that the trajectory prediction technique achieved convergence much faster. Importantly though, and as discussed within the summary of Chapter 7, the results presented were obtained from simulations that did not take into consideration the effect of additional external disturbances such as obstacles and ocean currents. This is particularly important as these disturbances are difficult/impossible to predict. Consequently, the efficiency of the trajectory prediction methodology may deteriorate as a result. Conversely, because the waypoint consensus technique does not rely on up to date positional information of neighbouring vehicles, the effect of these disturbances may be less severe. The aim of this chapter therefore is to describe the functionality implemented to accurately represent these external disturbances and thereafter, demonstrate whether or not their introduction impacts the efficiency of the two techniques described in Chapter 7.

To achieve this, the chapter is structured as follows. Section 8.2 presents the functionality implemented to provide an accurate representation of the external disturbances discussed above. This section also describes the subsequent changes made to the algorithm in order to ensure each vehicle is still capable of satisfying the necessary convergence criteria. Section 8.3 describes simulations setup and values used for the algorithmic parameters associated with these external disturbances. Section 8.4 then presents and analyses the results obtained from the simulations undertaken. Section 8.5 then describes the additional simulations completed to further test the robustness of the chosen methodology. Finally, Section 8.5 provides a summary of the work completed in this chapter.

8.2 Implementing External Disturbances

The purpose of this section is to not only describe the functionality implemented to incorporate a realistic representation of the external disturbances discussed above but also to describe the changes made to the algorithm to ensure the required parallel line formation can still be generated. To achieve this, the functionality associated with implementing a realistic representation of ocean currents will be presented in Section 8.2.1. Section 8.2.2 will then describe the associated changes made to the vehicles *Guidance System* to accommodate the inclusions of these currents. Thereafter, in Section 8.2.3, the functionality implemented to model the presence of external obstacles is presented before the chosen obstacle avoidance technique is described in Section 8.2.4.

8.2.1 Oceanic Currents Representation

As described in (Fossen 2011), ocean currents are predominantly horizontal circulation systems caused by forces acting upon the mean flow. These forces include gravity, wind friction, variations in the fluid's density, temperature and salinity as well as the Coriolis force. In order to accurately model these currents, the mathematical model of the RoboSalmon vehicle shown previously in Equation (3.12) has to be altered. The alteration is simple and involves changing the velocity vector within the Coriolis and hydrodynamic terms from $\boldsymbol{v}$ to $\boldsymbol{v}_r$ as shown below in Equation (8.1).

$$\mathbf{M}\dot{\boldsymbol{v}} + \boldsymbol{C}(\boldsymbol{v}_r)\boldsymbol{v}_r + \boldsymbol{D}(\boldsymbol{v}_r)\boldsymbol{v}_r + \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} \tag{8.1}$$

Here the vector $\boldsymbol{v}_r$ represents the relative velocity between the vehicle, $\boldsymbol{v}$ and the ocean current, $\boldsymbol{v}_c$ in the body-fixed axis as Equation (8.3) illustrates.

$$\boldsymbol{v}_r = \boldsymbol{v} - \boldsymbol{v}_c \tag{8.2}$$

Importantly, because only motion in the horizontal plane is taken into consideration within this work, the vector, $\boldsymbol{v}$ is equal to:

$$\boldsymbol{v} = \begin{bmatrix} u \\ v \\ r \end{bmatrix} \tag{8.3}$$

Moreover, as discussed in (Fossen 2011), if the current being modelled is assumed to be irrotational, Equations (8.2) and (8.3), when combined, produce the following simplified relative velocity vector:

$$\mathbf{v_r} = \begin{bmatrix} u - u_c \\ v - v_c \\ 0 \end{bmatrix} \tag{8.4}$$

Where the parameters, u_c and v_c represent the longitudinal and lateral velocities of the ocean current in the *body-fixed* axis and are calculated using the following two equations.

$$u_c = V_c \, cos(\beta_c - \psi) \tag{8.5}$$

$$v_c = V_c \, sin(\beta_c - \psi) \tag{8.6}$$

Where the parameters, β_c and ψ represent the angle of the ocean current and the vehicle in the *earth-fixed* frame of reference and V_c is calculated using the standard equation shown below.

$$V_c = \sqrt{u_c^2 + v_c^2} \tag{8.7}$$

Equations (8.1)-(8.7) have described the changes made to the mathematical model of the RoboSalmon vehicle in order to ensure a realistic representation of ocean currents are incorporated within it. Equations (8.5)-(8.6) demonstrate that the magnitude and direction of the current can be varied by altering the values selected for the parameters, V_c and β_c. It is important to note though that the values selected for V_c should always be smaller than the velocity of the vehicle. This is to ensure that the vehicle is still capable of manoeuvring in the required direction and the vehicles do not begin to drift with the current. As will be shown in Section 8.3, the values chosen for these parameters have been selected specifically to ensure that this is indeed the case.

8.2.2 Guidance System Compensation for Ocean Current Disturbances

Inspecting Equations (8.5) and (8.6), it is apparent that apart from the scenario where β_c is equal to ψ, incorporating an ocean current results in the value for the lateral velocity being non-zero when the vehicle is manoeuvring in a straight line. This is particularly important as it means there will now be a sideslip angle associated with the vehicle. As shown below in Figure 8.1, the sideslip angle is defined as the angle from the *body-fixed*, x-axis to the vehicles velocity vector, U. As the diagram demonstrates, the introduction of this sideslip angle will prevent the vehicle from being able to move in the direction of its required heading angle, ψ. Instead, the vehicle now manoeuvres in the direction denoted by the symbol, χ, i.e. its course angle.

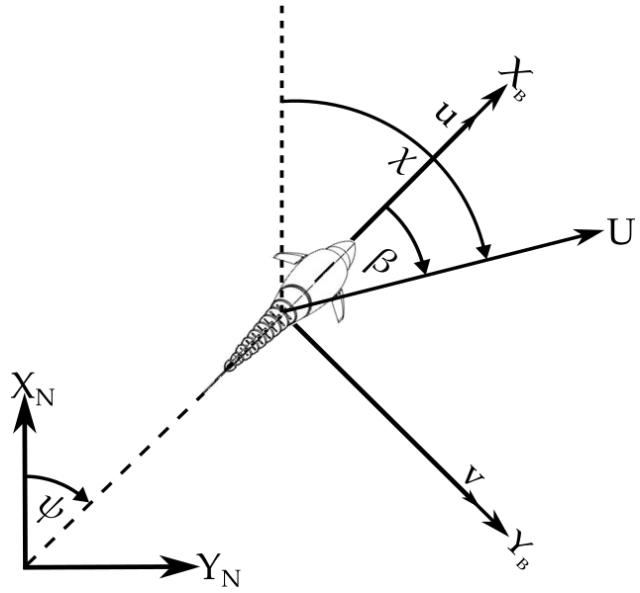


Figure 8.1 - Geometric relationship between course angle, χ, heading angle, ψ and sideslip angle, β.

Analysing Figure 8.1, it is apparent that when these three angles are combined, the course angle is equal to the sum of the heading angle and the sideslip angle.

$$\chi = \psi + \beta \tag{8.8}$$

Where the sideslip angle is calculated using the following equation.

$$\beta = sin^{-1}\left(\frac{v - v_c}{\sqrt{(u - u_c)^2 + (v - v_c)^2}}\right) \tag{8.9}$$

As Figure 8.1 demonstrates, if left uncompensated, the introduction of a sideslip angle will prevent each vehicle from being able to manoeuvre in the direction evaluated from the control laws presented throughout this thesis. To ensure that this is not the case, the guidance algorithm must be altered. This is achieved by rearranging Equation (8.8) to produce:

$$\psi_d = \chi - \beta \tag{8.10}$$

Where the course heading angle, χ represents the angle calculated using the control laws presented throughout this thesis. By implementing this modification, Equation (8.10) demonstrates that the guidance systems now instruct each vehicle to manoeuvre with a heading that takes into consideration the sideslip angle of the vehicle and as a result, align the vehicles velocity vector with its desired direction, χ. This is shown below in Figure 8.2.

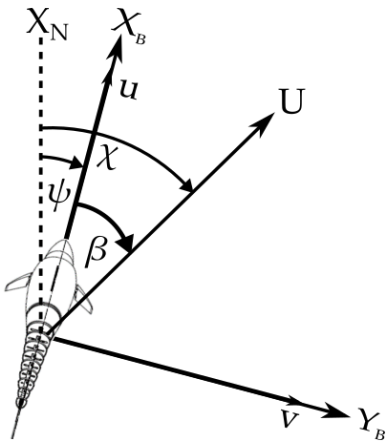


Figure 8.2- Altered vehicle heading angle taking into consideration sideslip angle.

Upon inspection, it is apparent from Figure 8.2 that after taking into consideration the sideslip angle, the heading angle of the vehicle, ψ is much smaller when compared with that shown in Figure 8.1. However, as required, the velocity vector of the vehicle, U is now aligned with the vehicle's required course direction, χ.

While Figure 8.2 demonstrates that the modification presented in Equation (8.10) ensures each vehicle is still capable of manoeuvring in the correct direction, these alterations have had to be included due to original Line-of-Sight calculation adopted and presented in Equation (3.25). Furthermore, as Equation (8.9) illustrates, these modifications require the vehicles to not only have an accurate representation of their own longitudinal and lateral velocities but also those of the oceanic currents. While this assumption has been made in this thesis and can be easily implemented within the mathematical model, in reality though, having accurate assessments of these parameters – particularly the oceanic current velocities – is difficult. Consequently, if applied in reality, it would be advantageous to replace the Line-of-Sight calculation presented in Equation (3.25) with a modern, state of the art Line-of-Sight algorithm such as the Integral Line-of-Sight technique. As discussed in (Caharija,2016) (Kelasidi,2017) , this technique automatically compensates for the sideslip angle introduced by oceanic currents without requiring the lateral and longitudinal components of the currents velocity.

8.2.3 Obstacle Detection Representation

Due to operating in a largely unknown environment, it is imperative that all vehicles have suitable obstacle avoidance capabilities to prevent collisions with underwater structures such as shipwrecks, cliffs, reefs and suspended mines (Braginsky & Guterman 2016). However, before the vehicle can make the necessary trajectory alterations they must firstly be able to sense, in real time, that an obstacle is positioned along its current trajectory. For

the majority of commercially available AUVs, this obstacle detection is achieved by using underwater imaging sonar technology (Zhao et al. 2008). Obstacle detection using this technique operates by emitting either a conical or fan shaped pulse in the direction of travel and analysing the intensity of the subsequent reflections to determine if an obstacle(s) is positioned somewhere within the confines of the emitted pulse. This obstacle detection process is presented diagrammatically below in Figure 8.3.

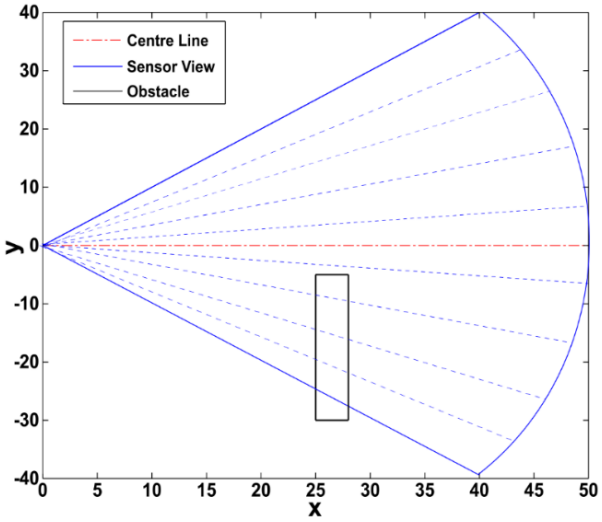


Figure 8.3 - Imaging sonar field-of-view model with obstacle.

In terms of implementing an accurate representation of this obstacle detection technique, it is apparent that obstacles can only be detected if they are within the field of view of the sonar sensor. To replicate this restriction within the mathematical model, each vehicle will only become aware of an obstacle when two specific conditions are satisfied. As shown below in Figure 8.4, these conditions are related to the *absolute* distance between the vehicle and the obstacle, d_{obs} and the associated relative angle, $\psi_{R_{obs}}$.

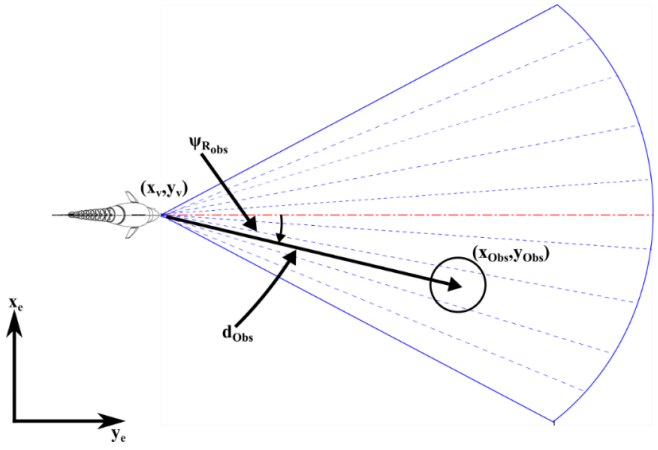


Figure 8.4 - Geometry associated with calculating the parameters, d_{obs} and $\psi_{R_{obs}}$.

Here, the angle, $\psi_{R_{Obs}}$ is calculated using Equations (8.11) and (8.12) below.

$$\psi_{R_{Obs}} = \psi_{LOS_{Obs}} - \psi_V \tag{8.11}$$

Here, ψ_V represents the heading angle of the vehicle in the *earth-fixed* reference frame and $\psi_{R_{Obs}}$ is the line-of-sight angle of the obstacle which is calculated using the following equation:

$$\psi_{LOS_{Obs}} = tan^{-1}\left(\frac{x_{obs} - x_v}{y_{obs} - y_v}\right) \tag{8.12}$$

As shown below in Equation (8.13), the distance between the vehicle and the obstacle is calculated using the standard equation for the distance between two points in 2-dimensional space.

$$\mathrm{d_{Obs}} = \sqrt{(x_v - x_{Obs})^2 + (y_v - y_{Obs})^2} \tag{8.13}$$

To implement an accurate representation of the obstacle detection technique presented in Figure 8.3, the parameters, $\psi_{R_{Obs}}$ and d_{Obs} should satisfy the following two conditional statements before the vehicle becomes aware of the obstacles.

$$\left|\psi_{R_{Obs}}\right| \leq \frac{\pi}{4} \tag{8.14}$$

$$\mathrm{d_{obs}} \leq 12\mathrm{m} \tag{8.15}$$

It is important to note that the parameter d_{abs} is calculated to be the absolute distance from the vehicles centre of gravity to the centre of the obstacle. Given that the obstacles have a radius of 6m and the short length of the RoboSalmon vehicle (0.9m), this means that each vehicle has a distance of approximately 5m in which to manoeuvre round the obstacle after it first becomes aware of its presence. This distance was deemed to be suitable due to the relatively slow speed of the vehicles and their fast turning capability.

Regardless, to incorporate this simplified representation of the obstacle detection technique, additional subsystems have had to be incorporated within the original AUV architecture presented previously in Chapter 4 (Figure 4.2). These alterations are shown below in Figure 8.5.

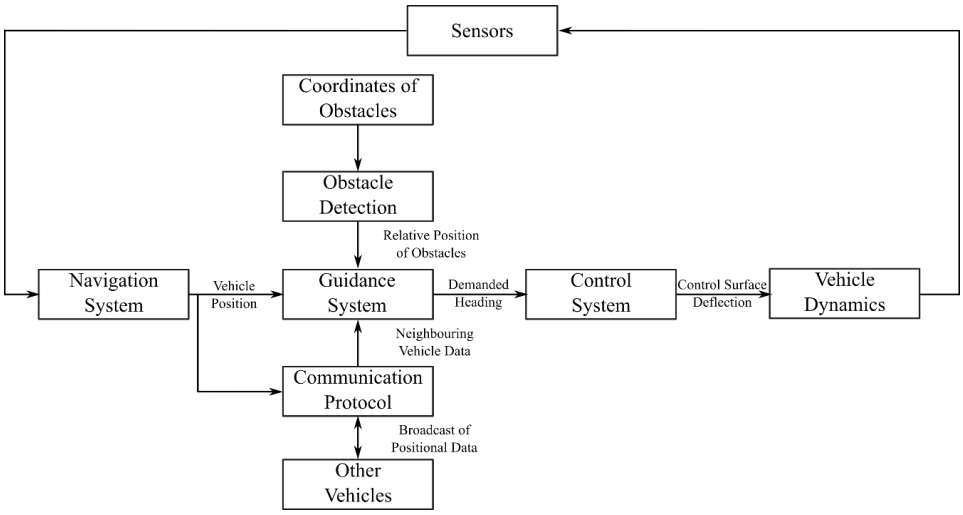


Figure 8.5- Altered AUV system architecture to incorporate obstacle detection functionality.

The diagram above demonstrates that the simplified representation of the obstacle detection technique is incorporated across two subsystems. The first subsystem contains a list of coordinates that represent the positions of each obstacle within the simulation. The second subsystem contains the actual *Obstacle Detection* functionality. This subsystem uses the coordinates held by the first subsystem to calculate the parameters presented in Equations (8.12) and (8.13). Thereafter the subsystems determine whether or not the conditional statements in Equations (8.14) and (8.15) have been satisfied. If so, the functionality proceeds to pass the coordinates of that particular obstacle to the vehicle's *Guidance System*. Conversely, if the conditional statements in Equations (8.14) and (8.15) are not satisfied, no data is passed to the *Guidance System*.

8.2.4 Obstacle Avoidance – Guidance System Alteration

To prevent the vehicles from colliding with any detected obstacle, a suitable obstacle avoidance technique must be implemented. For this study, a collision cone strategy has been implemented (Chakravarthy & Ghose 1998). This technique operates by providing each vehicle with a range of angles (collision cone) that if the vehicle is to manoeuvre with, a collision will become inevitable (Rafferty 2014). Therefore, to prevent collisions, each vehicle, if necessary, is instructed to alter its course angle to ensure that it remains out with the calculated collision cone. The associated geometry of this collision cone strategy is shown below in Figure 8.6.

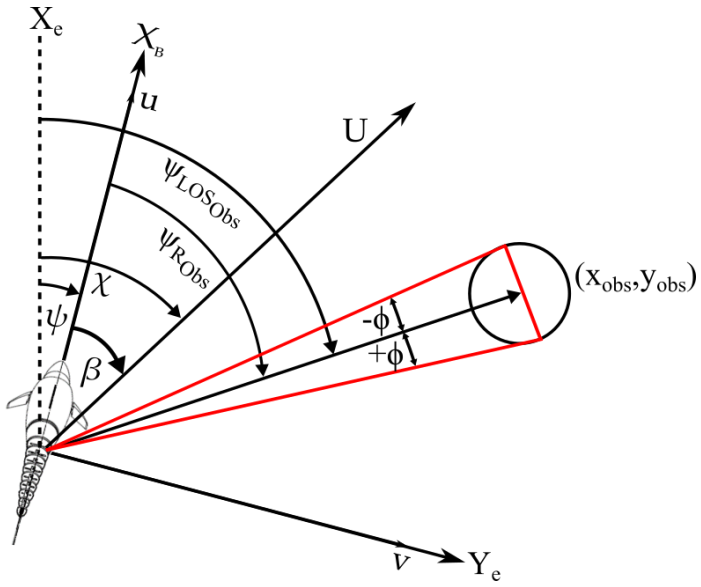


Figure 8.6- Geometry associated with collision cone (Red Triangle).

Using this geometry, it is apparent that the collision cone (red triangle) can be defined in the *Earth-fixed* reference frame using the two angles presented below in Equations (8.16) and (8.17).

$$\alpha_l = \psi_{LOS_{Obs}} - \phi \tag{8.16}$$

$$\alpha_u = \psi_{LOS_{Obs}} + \phi \tag{8.17}$$

Here the angle, ϕ can be calculated using Equation (8.18) below.

$$\phi = sin^{-1}\left(\frac{r_{obs}}{d_{obs}}\right) \tag{8.18}$$

Since every obstacle implemented in this work has been incorporated as a circle, the parameter r_{obs} in Equation (8.18) is set to the radius of the obstacle. Moreover, it is also important to note that every obstacle is assumed to have the same radius length, i.e. 6m. The angles, α_l and α_u define the range of course heading angles that the vehicle should avoid in order to prevent a collision with the obstacle. Furthermore, while the example presented above in Figure 8.6 considers only a single obstacle, the collision cone technique can be extended to take into consideration multiple obstacles as shown below in Figure 8.7.

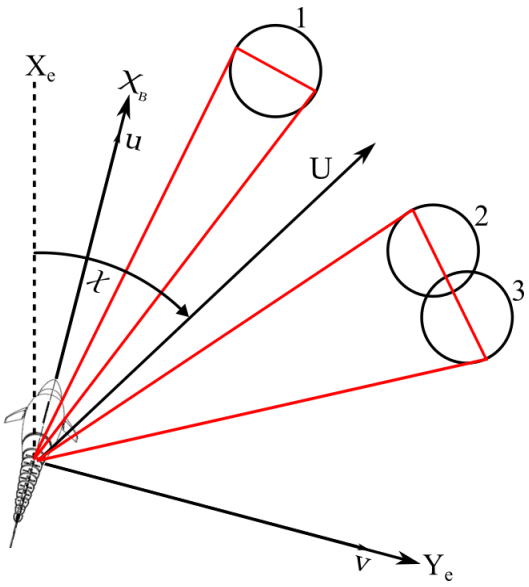


Figure 8.7- Implementation of collision cone technique for multiple obstacles (Diagram Not to Scale).

As the values presented in Table 8.1 illustrate, the collision cones associated with each obstacle presented in Figure 8.7 can be defined using the following sets of angles, i.e. α_1= [30º,40º], α_2= [50º,60º] and α_3= [58º,68º]. Importantly, because the circles representing the second and third obstacles intersect, the combined cone angle for these two obstacles can be rewritten as $\alpha_{2/3}$= [50º, 68º]. Overall, when combined, the total cone angle for these three obstacles can be written as the union of the different sets, i.e. $\alpha = [30º, 40º] \cup [50º, 68º]$.

Table 8.1 - Collision cone angles.

Obstacle	$\boldsymbol{\alpha_l}$	$\boldsymbol{\alpha_u}$
1	30º	40º
2	50º	60º
3	58º	68º

If the desired course angle calculated by the formation control algorithm is out with the confines of the union of angles represented by the parameter α, then there is no need for the vehicle to alter its course angle. Conversely, if the vehicles course angle is on a collision course with one of the obstacles, the vehicle's desired heading angle is altered to the angle closest to the original desired angle but importantly out with the confines of the collision cone. Lastly, as shown below in Figure 8.8, in order to ensure that this obstacle avoidance

manoeuvre is undertaken, the system architecture presented previously in Figure 8.5 has once again been modified.

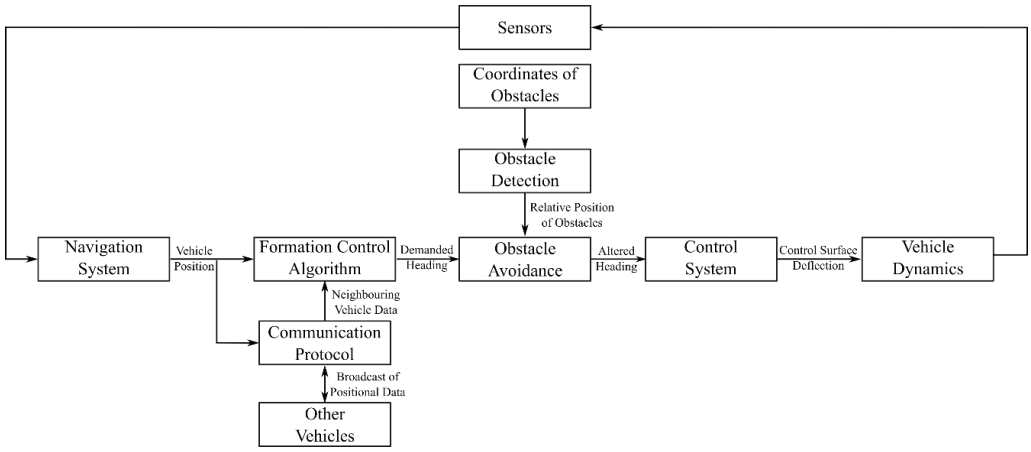


Figure 8.8- Altered AUV system architecture with obstacle avoidance incorporated.

This altered architecture shows that the *Guidance Sys*tem has been separated into its two main components, the formation control algorithm and the obstacle avoidance strategy. This has been undertaken deliberately to illustrate that the obstacle avoidance functionality is only evaluated once the formation control algorithm has calculated the vehicle's desired heading angle. Thereafter, Figure 8.8 demonstrates that this angle is passed to the *Obstacle Avoidance* subsystem where the decision making process shown below in Figure 8.9 is completed.

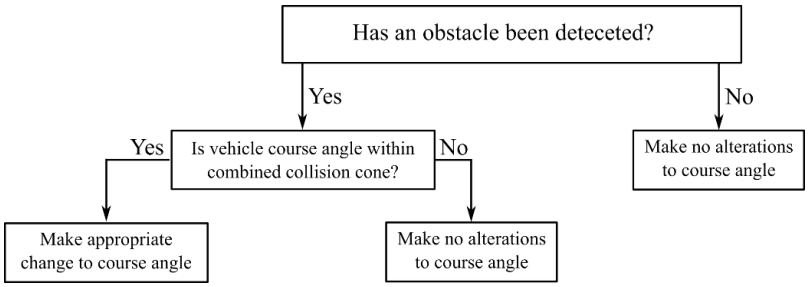


Figure 8.9 - Decision making process completed within obstacle avoidance subsystem.

By completing this decision making process and implementing, where required, the obstacle avoidance manoeuvre discussed above, it should be possible for each vehicle to avoid colliding with any external obstacles. The only situation that would result in the vehicles not being able to manoeuvre round the detected obstacles is when the restrictions imposed on the algorithm in Equations (8.14) and (8.15) result in the vehicles becoming trapped. An example of such a situation is shown below in Figure 8.10.

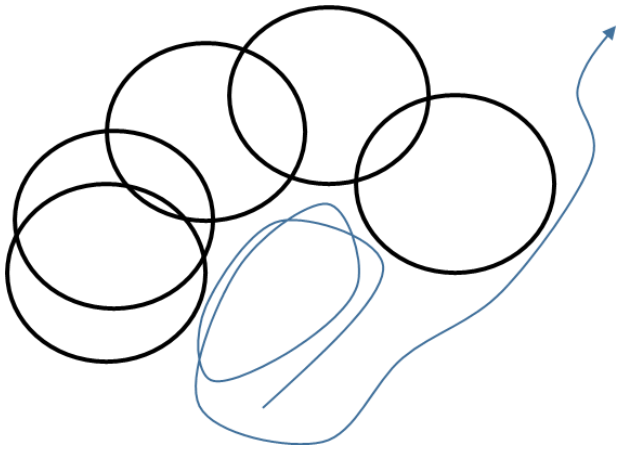


Figure 8.10 - Example scenario when obstacle avoidance technique results in a vehicle becoming trapped.

The diagram shown in Figure 8.10 demonstrates a possible scenario resulting from the implementation of the obstacle avoidance technique presented in this chapter. The diagram demonstrates that because the vehicle doesn't know the location of all the obstacles, it makes trajectory manoeuvres that result in it moving towards more obstacles instead of away from them. In the scenario in Figure 8.10, the vehicle is eventually able to manoeuvre past the obstacles but depending on the geometry of the vehicles position and that of the obstacles, it is possible that the vehicle would remain trapped and unable to manoeuvre past the obstacles. The only way to overcome this problem would be to provide the vehicle with the capability to be able to "see" more vehicles at any one time or conversely, for the vehicle to store the positions of known obstacles and use these within the avoidance technique presented in this section. However, for this thesis, each vehicle will only be aware of obstacles that satisfy the conditions of Equations (8.14) and (8.15).

8.3 Simulation Setup & Performance Metrics

To allow a direct comparison with the results presented in the previous chapter, it is necessary to ensure that the simulation setup is identical to those used in Chapter 7. To demonstrate that this is indeed the case, shown below in Table 8.2 is a summary of the values used for the various algorithmic parameters.

Compared to the simulation setup presented in Chapter 7, the only difference for the simulations undertaken in this chapter is the fact that the timeslot size is no longer varied but instead, kept constant at 24s. The reason for only testing this particular timeslot size is simply due to the fact that it represents the worst-case scenario in terms of communication throughput tested in this thesis. Subsequently, if incorporating these external disturbances

is to affect the ability of the trajectory prediction and waypoint consensus algorithms, it will be most profound when this timeslot size is implemented.

Table 8.2 - Variation in simulation parameters.

Parameter	Group 1	Group 2
Technique Implemented	Trajectory Prediction	Waypoint Consensus
Behavioural Zones Size (m)	[20,25,200]m & [6,11,200]m	
Timeslot Size ($T_{Timeslot}$)	24s	
Transmission & Propagation Delay (T_{Tot})	0.205s	
Reference Heading Angle (ψ_{ref})	$\frac{\psi}{4}$	

However, what Table 8.2 does not present is the values used for the parameters associated with the external disturbances, i.e. the magnitude of the current's velocity as well as its flow angle and also the size and number of stationary obstacles. The following two sections detail and justify the values used for these parameters in more detail.

8.3.1 Simulation Parameters – Ocean Currents

As presented in Section 8.2.1, an ocean current can be represented by the magnitude of its velocity vector, V_c, as well as its flow angle, β_c. While the value selected for the current's flow angle has no limitations, the magnitude of its velocity has to be within the limits shown below in Equation (8.19) (Fossen 2011).

$$V_{min} \leq V_c \leq V_{max} \tag{8.19}$$

Here the parameters, $V_{\min}$ and $V_{\max}$ represent the minimum and maximum attainable velocities of the vehicle. Consequently, for the simulations completed throughout this study, it was decided that the value selected for the parameter, V_c would be equal to 0.1m/s, i.e. half the nominal velocity of the RoboSalmon AUV. This particular value has been selected to ensure that regardless of the direction of the ocean current, the vehicle is still be able to manoeuvre in the direction instructed by the formation control algorithm.

For the current's direction, it has been decided that a constant angular value of 135º would be used. This angle has been chosen specifically as it means that as the vehicles are

manoeuvring to generate the required formation, vehicles at one side of the group are subjected to a tail current while the vehicles at the other side are subjected to a head current. As shown below in Figure 8.11, this results in the greatest possible velocity differential among the vehicles when generating the required parallel line formation.

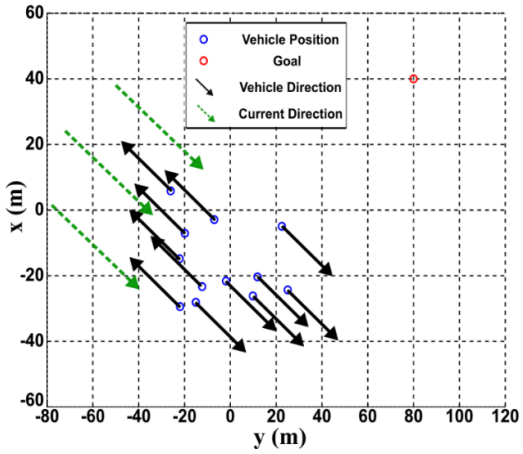


Figure 8.11- Diagrammatic representation of the selected ocean current direction compared with the initial vehicle heading direction.

8.3.2 Simulation Parameters – Obstacles

As discussed previously in Section 8.2.3, each obstacle implemented within is defined by the x and y coordinate of its centre position as well as the magnitude of its radius (6m). As Figure 8.5 demonstrates, these coordinates along with the criteria presented in Equations (8.14) and (8.15) are used within the *Guidance System* of each vehicle to determine whether or not an obstacle has been detected. As the main aim of this investigation is to determine what effect introducing these obstacles has on the performance of the algorithms, it was imperative that the coordinates selected positioned the obstacles in the path of a number of vehicles from within each simulation. To ensure that this is indeed the case, the trajectories obtained from the simulations completed in Chapter 7 were analysed to obtain a reference area within which a number of obstacles were positioned.

8.3.3 Summary of Simulation Test Cases

As shown below in Table 8.3, to test and compare the introduction of both ocean currents and external obstacles on the trajectory prediction and waypoint consensus algorithms, four sets of simulations were completed.

As throughout the work completed in this Thesis, the 200 simulations completed within each test case contain 100 simulations where the vehicles are initially using the *repulsive* control law and 100 simulations where the vehicle are initially using the *attractive* control

law. It is also apparent from Table 8.3 that for the two algorithms, i.e. trajectory prediction and waypoint consensus, the two external disturbances are tested separately. This is to ensure that the subsequent analysis could easily ascertain the specific effect each disturbance has on the respective algorithms.

Table 8.3 - Summary of simulation test cases undertaken.

Simulation Test Case Number	**1**	**2**	**3**	**4**
Algorithm Implemented	Trajectory Prediction	Trajectory Prediction	Waypoint Consensus	Waypoint Consensus
External Disturbance Implemented	Ocean Current	External Obstacle	Ocean Currents	External Obstacle
Number of Simulations Completed	200	200	200	200
Timeslot Size	24s	24s	24s	24s
Transmission Time & Propagation Delay	0.205s	0.205s	0.205s	0.205s
Reference Heading Angle	45º	45º	45º	45º

8.4 Results

The purpose of this section is to demonstrate what effect the incorporation of external disturbances has on the ability of the trajectory prediction and waypoint consensus algorithms to generate the required parallel line formation. To achieve this, Section 8.4.1 discusses the results obtained from the simulations implementing the trajectory prediction algorithm while Section 8.4.2 discusses the results obtained using the waypoint consensus algorithms.

8.4.1 Convergence Performance - Trajectory Prediction Algorithm

Shown below in Figures 8.12 (a) – (e) is the evolution of the percentage of vehicles satisfying the various convergence associated with generating the required parallel line formation with and without the inclusion of the external disturbances discussed this chapter.

Analysing the results associated with the inclusion of the stationary obstacles, it is apparent that the trajectory prediction algorithm is still capable of generating the required parallel line formation. This is due to the fact that in each individual figure, i.e. (a) – (d), the necessary convergence percentage has been satisfied. However, when comparing the

results presented below with those of Figure 7.9 where no external disturbances were incorporated, it is apparent that the time taken to achieve convergence has increased significantly.

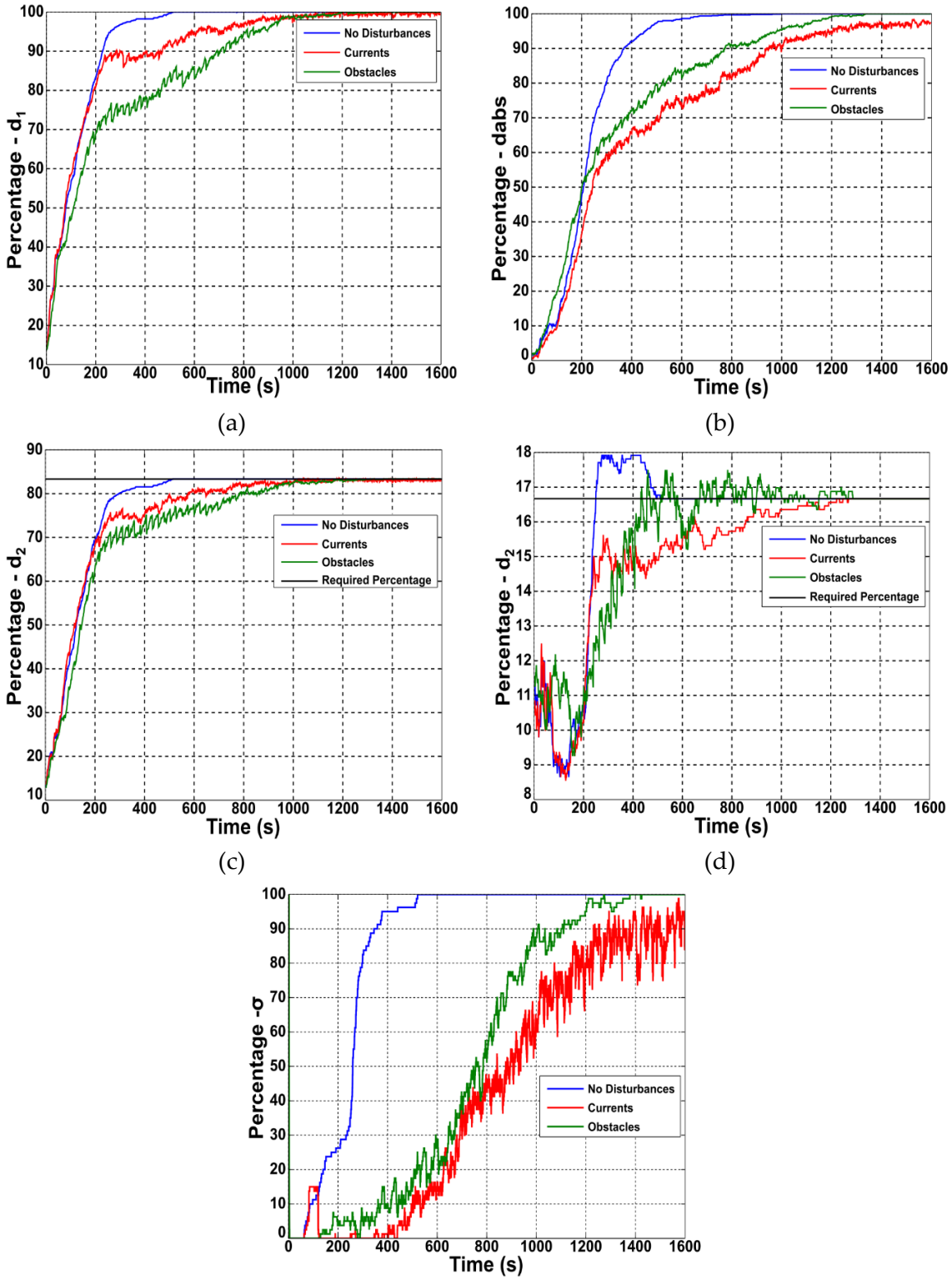
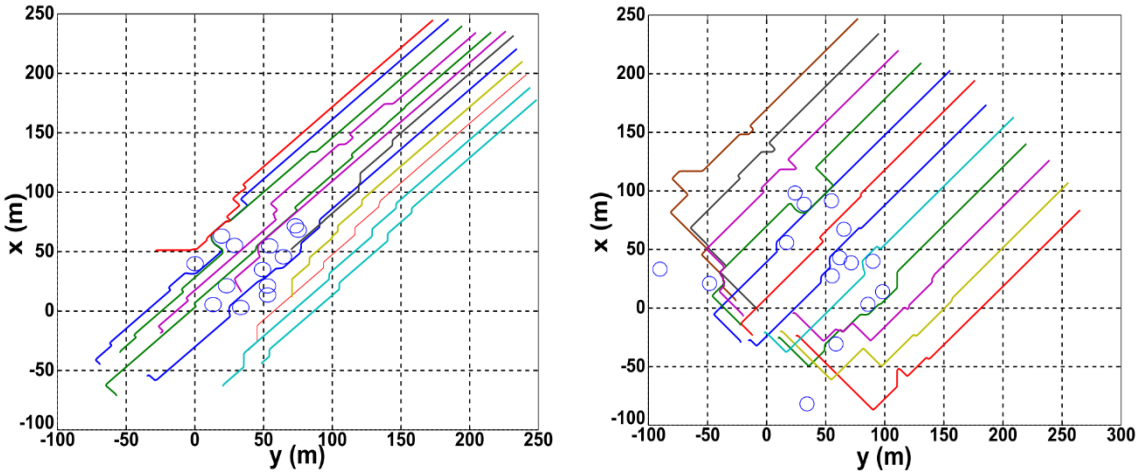


Figure 8.12- Evolution of the percentage of vehicles satisfying necessary convergence criteria when trajectory prediction technique is implemented and disturbances are incorporated.

While this increase in convergence time is significant, it is not unexpected and as shown below in Figure 8.13, can be explained by the fact that a number of vehicles are having to undertake the obstacle avoidance manoeuvres discussed in Section 8.2.4 before generating the required formation.

While a portion of this time can be attributed to the time required by the vehicles to manoeuvre past the obstacles, Figure 8.13 demonstrates that a significant portion of this time can be attributed to a number of vehicles having to complete corrective manoeuvres.



(a) (b)

Figure 8.13- Example of obstacle avoidance manoeuvres completed throughout simulations implementing the trajectory prediction algorithm and when the vehicles were initially using (a) *attractive* control law and (b) *repulsive* control law.

These corrective manoeuvres occur throughout the trajectories presented in Figures 8.13 (a) and (b) and are caused when a vehicles timeslot occurs while it is completing an obstacle avoidance manoeuvre and as such, the vehicle communicates a position vector that does not agree with the values obtained from the trajectory prediction algorithm of a neighbouring vehicle. Similarly to the results presented in Chapter 6, this error results in the algorithm of a neighbouring implementing the wrong control law which then has to be subsequently corrected when the next communication update is received.

For example, in Figure 8.13 (b) above, it is apparent that Vehicle 4 (fourth from the left) communicates its position while manoeuvring past the right hand side of the two obstacles in its path. Thereafter, because Vehicle 4 has moved *laterally* away from Vehicle 3, Vehicle 3, initiates the *attractive* control law and begins to reduce the *lateral* distance between the two vehicles back to within the confines of the orientation zone. However, once Vehicle 4 manoeuvres past the obstacles, it begins using the *repulsive* control law once more and

begins to manoeuvre back towards Vehicle 3. Vehicle 3, on the other hand, is unaware of this manoeuvre and continues to implement the *attractive* control law until it receives a positional update from Vehicle 4. At this point, Vehicle 3 makes the corrective manoeuvre and implements the *repulsive* control law one final time until the *lateral* distance between to the two vehicles is within the confines of the orientation zone. Furthermore, Figure 8.13 (b) demonstrates that the aforementioned incorrect implementation of the *attractive* control law by Vehicle 3 has a knock-on effect that results in both Vehicles 1 and 2 having to make additional heading angle changes.

Therefore, the introduction of the external obstacles results in an error occurring within the trajectory prediction algorithm which, similarly to the introduction of the TDMA protocol, results in a number of vehicles having to complete a number of additional corrective manoeuvres before the required parallel line formation can be generated.

While an increase in the number of heading manoeuvres completed is inevitable, the results discussed and presented above in Figure 8.13 illustrate that due to the design of the trajectory prediction algorithm, a significant percentage of these additional manoeuvres are unnecessary.

In terms of the effect of introducing ocean currents, the results presented in Figure 8.13 demonstrate that a significant number of vehicles are unable to continuously manoeuvre with the exact same heading direction. This can be attributed to the fact that as shown in Figures 8.12 (a), (b) and (c), there is a small, yet continuous fluctuation in the percentage of vehicles satisfying the criteria associated with the vehicles *lateral* and *absolute* nearest neighbour distances. Subsequently, as was the case when the TDMA protocol was incorporated in Chapter 6, this fluctuation means that a number of vehicles are periodically switching between implementing the different control laws. However, more importantly, due the incorporation of the ocean currents, every time a vehicle alters its velocity, it will also need to change its heading angle as per Equation (8.10) in order to ensure the vehicle manoeuvres in the required direction. Consequently, due to the convergence performance shown in Figure 8.12 (b), vehicles are continuously altering their velocity. As a result, their heading angle is continuously changing. Therefore, these two factors explain why the convergence performance associated with the standard deviation of the vehicle heading angle is significantly reduced.

As with the results associated with the introduction of the obstacles, the continuous fluctuation presented in Figure 8.12 (a), (b) and (c) can be attributed to the fact that the trajectory prediction algorithm no longer has a continuously accurate representation of the positioning of its nearest neighbours. Furthermore, as shown below in Figure 8.14, because

the ocean currents are implemented throughout the entirety of the simulations, the associated errors never converge to zero.

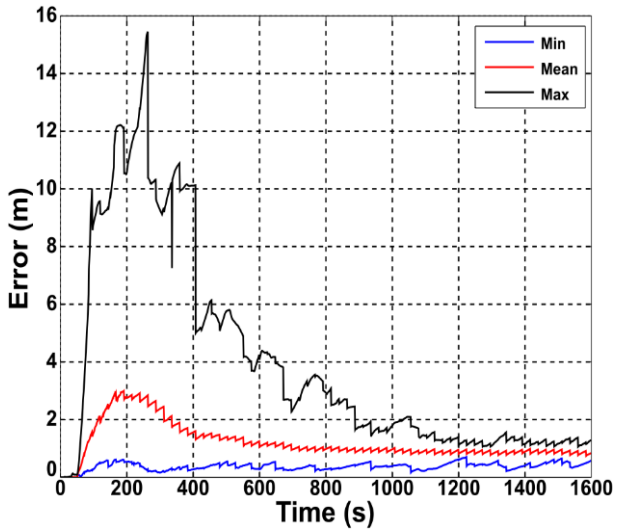


Figure 8.14 - Evolution of the minimum, mean and maximum errors associated with the trajectory prediction functionality when ocean currents are included.

As with the results presented in Figures 8.12-13 and similarly to those presented in Chapter 6, the errors produced from the trajectory prediction algorithm as a result of implementing ocean currents is likely to result in the trajectory prediction algorithm either incorrectly altering the vehicles velocity or implementing the wrong control law. Both of which, as discussed above, will result in the vehicle unnecessarily altering its heading angle. Finally, when combining the results presented in Figure 8.12 and 8.14 and taking into consideration the cyclical nature of the TDMA protocol, it is apparent that this selection of the wrong control law or vehicle velocity will continues as long as there is an external disturbance within the simulated environment. Finally, in terms of the algorithms performance, the number of manoeuvres completed by each vehicle, increased on average, by more than fivefold as a result implementing ocean currents. While a number of these additional manoeuvres are required, the results presented above demonstrate that due to the errors associated with the trajectory prediction algorithm, a significant percentage of these manoeuvre are unnecessary and are due to the limitations of the trajectory prediction algorithm.

8.4.2 Convergence Performance – Waypoint Consensus Algorithm

As with the results presented in Figure 8.12, shown below in Figure 8.15, is the convergence performance of the waypoint consensus algorithm with and without the external disturbances introduced in this chapter.

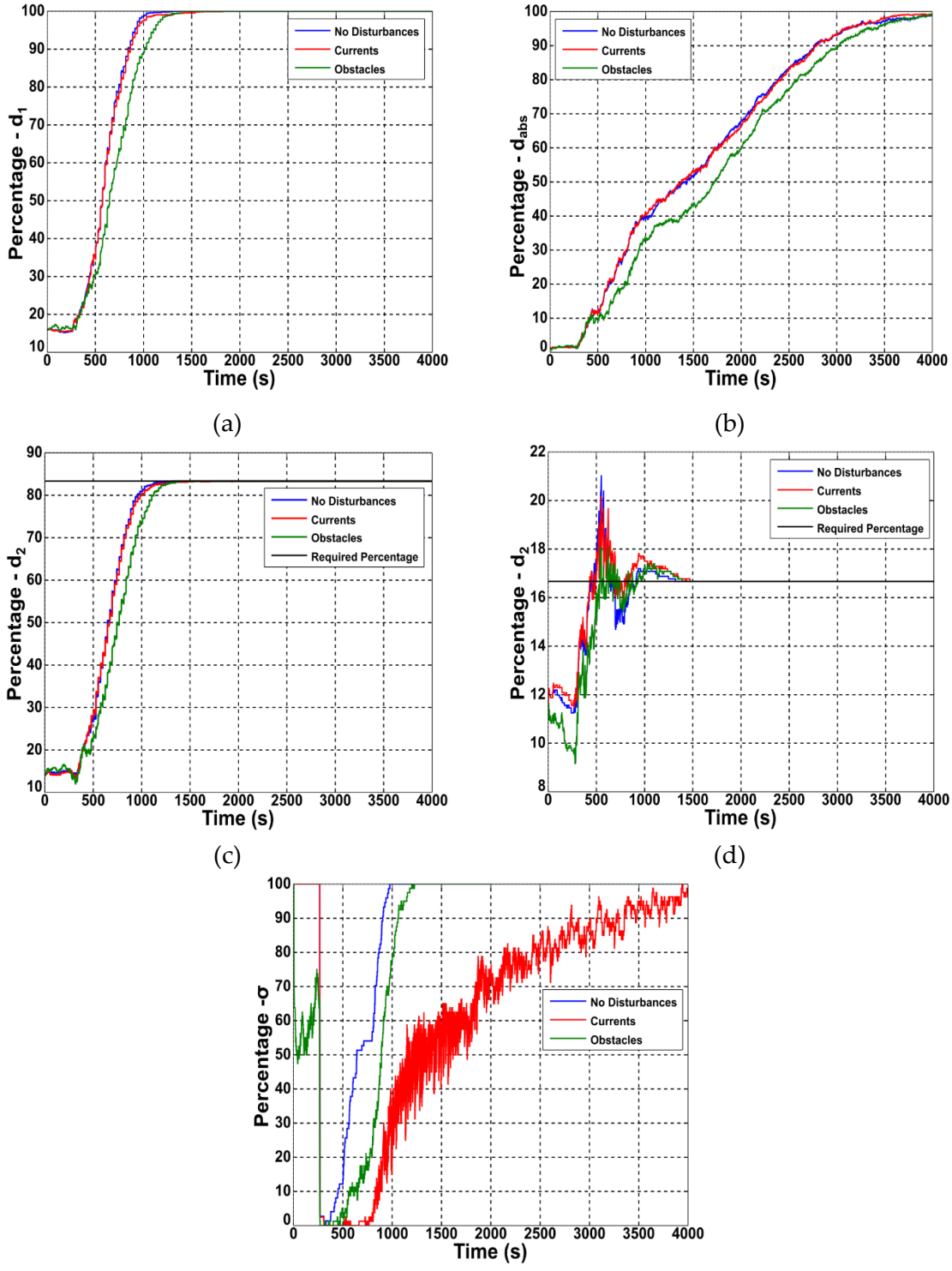


Figure 8.15 - Evolution of the percentage of vehicles satisfying necessary convergence criteria when the waypoint consensus algorithm is used and external disturbances are incorporated.

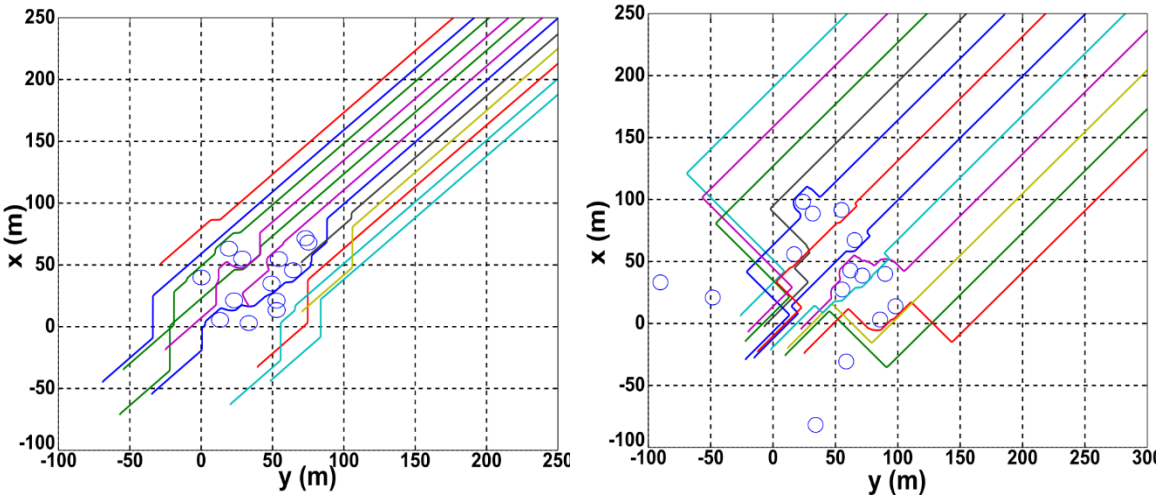
Analysing the results presented in Figures 8.15 (a) - (d), it is apparent that the convergence performance of the waypoint consensus algorithm with ocean currents incorporated varies very little from the results obtained without any external disturbances. However, when analysing Figure 8.15 (e), it is apparent that the results obtained with and without ocean

currents for the standard deviation of the vehicles heading angle vary drastically with the convergence time increasing by approximately 3000s. In order to understand why this is the case, it is important to emphasise that due to the incorporation of ocean currents, every time a vehicle alters its velocity, the sideslip angle of the vehicle also changes and as a result, the vehicles desired heading angle needs to be altered as per Equation (8.10). Therefore, it should be expected that the convergence criteria associated with the standard deviation of the vehicles heading angle continues to fluctuate as long as vehicles continue to alter their velocity. Subsequently, because the vehicles can only converge to an appropriate *absolute* nearest neighbour distance by altering their velocity, the convergence of the parameter, σ should occur at approximately the same rate as the convergence of the parameter, d_{abs}. This is indeed shown to be the case when the results presented in Figures 8.15 (b) and (e) are compared.

Moreover, because the vehicles now have to alter their heading angle every time a change in velocity is requested, the number of heading manoeuvres completed by each vehicle increased by a factor of four when compared with the results obtained with no currents included.

The results presented above demonstrate that the introduction of ocean currents has had little effect on the convergence performance of the waypoint consensus algorithm apart from the unavoidable increase in the time taken for the parameter, σ to converge. Likewise, the fourfold increase in the number of manoeuvres completed by each vehicle is also unavoidable and is caused by each vehicle having to take into consideration the sideslip angle of the vehicle each time it alters its velocity. The only way to reduce the number of heading manoeuvres completed by the vehicles would be to redesign the velocity control law presented in Chapter 4.

When the results associated with the incorporation of stationary obstacles are analysed, it is apparent that the algorithms convergence performance is once again affected very little apart from a relatively small increase in the convergence time. This, as discussed in the previous section is due to the additional time required by the vehicles to complete the obstacle avoidance manoeuvres presented in Section 8.2.4. Moreover, due to the fact that the waypoint consensus algorithm uses a single, stationary waypoint to determine when each vehicle implements the different control laws, it is not susceptible to incorrectly implementing the wrong control law as presented in the previous section. Subsequently, as shown below in Figure 8.16, the vehicles are not forced to complete additional and unnecessary corrective manoeuvres.



(a) (b)

Figure 8.16 - Example of obstacle avoidance manoeuvres completed when using the waypoint consensus algorithm and the vehicles are initially using (a) the attractive control law and (b) the repulsive control law.

8.4.3 Algorithm Comparison & Decision

Based on the results presented in the previous two sections, it is apparent that of the two algorithms, the waypoint consensus algorithm is significantly less affected by the inclusion of the external disturbances. This is primarily due to the fact that unlike the trajectory prediction algorithm, the waypoint consensus algorithm isn't reliant on a continuously accurate representation of the positioning of neighbouring vehicles. Subsequently, by design, it is incapable of selecting the wrong control laws as has shown to be the case for the trajectory prediction algorithm. Furthermore, the deviations in performance seen in the waypoint consensus algorithm are as expected with an increased convergence time due to the vehicles manoeuvring around the obstacles and more heading manoeuvres resulting from the vehicles having to take sideslip into consideration.

Conversely, while the results demonstrate that the trajectory prediction algorithm is still capable of generating the required formation, the results also illustrate its tendency to select the incorrect control law when access to up-to-date positional information is unavailable. Therefore, because of this undesirable characteristic, confidence in the trajectory prediction algorithm to operate as required in the unpredictable underwater environment cannot be guaranteed.

As a result, the waypoint consensus algorithm's design is more resilient and therefore, better suited to overcoming the nuances of the underwater environment. For these reasons,

the remaining work completed in this thesis will only use the waypoint consensus algorithm.

8.5 Testing Algorithm Robustness

As mentioned throughout this thesis, the simulations completed so far have only used a limited set of values for the various algorithmic parameters that can be varied within the formation control algorithm. As shown below in Table 8.4, these parameters include the reference heading angle, the size of the orientation zone, the current flow angle, the magnitude of the current's velocity as well as the size of the obstacles included. These parameters have purposefully been kept constant to ensure that any changes to the algorithm's performance can be attributed to and explained by the specific characteristic of the underwater environment incorporated, i.e. the incorporation of the TDMA protocol, the ocean currents as well as the stationary obstacles. However, to test the robustness of the waypoint consensus algorithm, it is important to demonstrate that the algorithm is still able to operate successfully regardless of the values used for these parameters.

Table 8.4 - Summary of algorithmic parameters.

Parameter	Symbol
Position of Recovery Area	(x_{des}, y_{des})
Reference Heading Angle	ψ_{ref}
Deployment Orientation Zone Size	(zoo_{ID}, zoo_{uD})
Threshold for Initiating Recovery Phase	d_{rec}
Recovery Orientation Zone Size	(zoo_{IR}, zoo_{uR})
Magnitude of Ocean Current	V_c
Direction of Ocean Current	β_c
Radius of Obstacle	r_{obs}

While it would be impractical to attempt to test every possibly combination of the parameters presented above, the following sections will now describe the simulation setup used to ensure that not only was the robustness of the algorithm tested sufficiently but it was done so within a scenario that is representative of an oceanic sampling mission.

8.5.1 Simulation Setup – Mission Overview

The first objective of the waypoint consensus algorithm is to allow a group of vehicles – initially deployed from a surface vehicle - to generate the parallel line formation necessary

for efficient oceanic sampling missions. Once the vehicles have completed their sampling objectives, the algorithm is then tasked with ensuring the vehicles subsequently converge to a predefined *recovery* area to be collected. So far, this process has been simplified by selecting coordinates for the *recovery* area to be such that the reference heading angle, ψ_{ref}, is always equal to 45º. However, as shown below in Figure 8.17, for the work completed in this study, the coordinates of the *recovery* area are varied so that every integer value of ψ_{ref} to the starboard side of the surface vessel is tested.

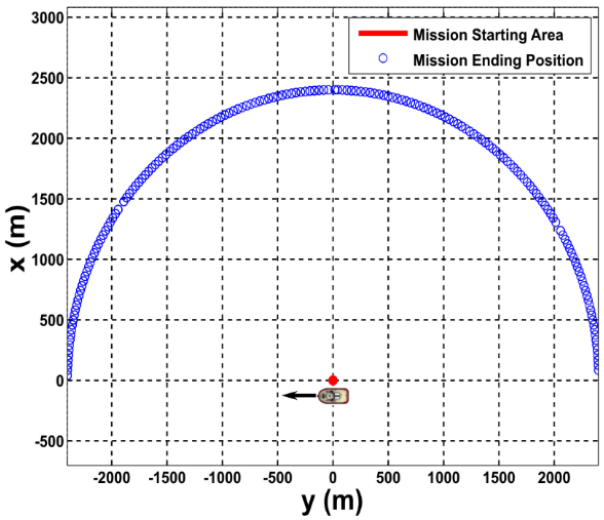


Figure 8.17 - Variation in the mission recovery positions used for the simulations completed within this study (Surface Vessel -Not to Scale).

The reason that only angular values to the starboard side of the surface vehicle are used is due to the assumption that because the vehicles have been deployed from the starboard side of the vehicle, it is unlikely that the mission objective will then instruct the vehicles to manoeuvre in a direction that is to the portside of the surface vessel.

Regardless, Figure 8.17 clearly illustrates that the coordinates selected for the *recovery* areas ensure that the algorithm is tested for every possible integer value for ψ_{ref} in the interval, $-90º \leq \psi_{ref} \leq 90º$. Consequently, in order to test each integer value of ψ_{ref}, 181 individual simulations have to be completed.

As well as defining the coordinates for the *recovery* area, Figure 8.17 illustrates that the initial positioning for each vehicle must also be defined. This process and the restrictions placed on the initial position of the vehicles are described below in Section 8.5.1.1.

8.5.1.1 Initial Vehicle Positioning

To ensure that each vehicle is at a safe distance from their nearest neighbour at the start of each simulation, the initial positioning of each vehicle was chosen to ensure that their *absolute* nearest neighbour distance would be at least 3m. As shown below in Figure 8.18, this results in the majority of vehicles having an initial *absolute* nearest neighbour distance that is between three and seven metres.

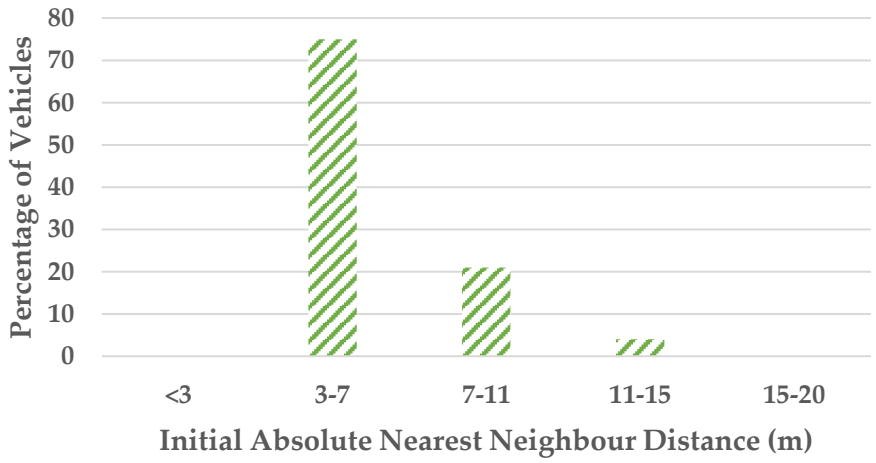


Figure 8.18 - Variation in the percentage of vehicles with different initial *absolute* nearest neighbour distances.

Intuitively, the associated initial *lateral* nearest neighbour distance for each vehicle will be by default, smaller. This is shown below in Figure 8.19 where the results demonstrate that approximately 80% of the vehicles have an initial *lateral* nearest neighbour distance (relative to the reference heading angle of that simulation) that is less than 3m.

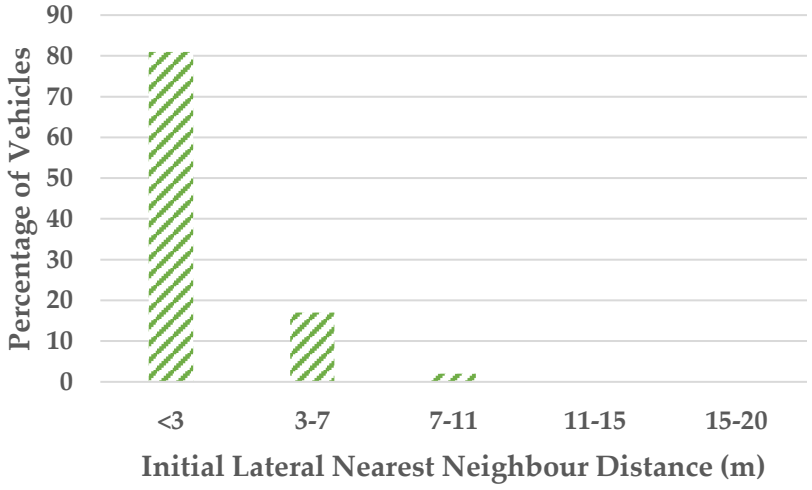


Figure 8.19- Variation in the percentage of vehicles with different initial *lateral* nearest neighbour distances.

8.5.1.2 Size of Orientation Zone during Deployment & Mapping Phase

The size of the orientation zone during the *Deployment* and *Mapping* phases dictates the *lateral* separation distance between neighbouring vehicles. Up to this point, only two sets of values have been used for this parameter, i.e. 6-11m and 20-25m. In this simulation study however, every possible integer value between 10 and 45 meters has been tested for this parameter. With the distance between the lower and upper boundaries remaining unchanged at 5m, the associated values for the upper boundary of the orientation zone range from 15m to 50m.

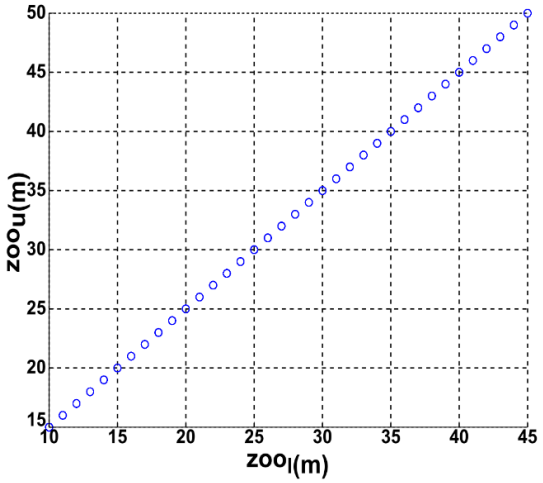


Figure 8.20 - Different orientation zone sizes used throughout simulation study.

The above figure demonstrates that for the simulations completed, there are thirty-six unique orientation zone sizes. This figure is smaller than the 181 simulations necessary to test every integer value for the vehicle's reference heading angle. As a result, each unique orientation zone size was used multiple times. The distribution of the orientation zone sizes used is shown below in Figure 8.21.

The results demonstrate that the percentage value is approximately the same for the various subgroups This is particularly important as it demonstrates that in terms of the size of the orientations zone, the algorithm has been tested evenly across the simulations completed. Finally, the reason that the lower boundary of the orientation zone is limited to 45m is due simply to time restrictions and the fact that the simulation run time increases every time the orientation zone size is increased.

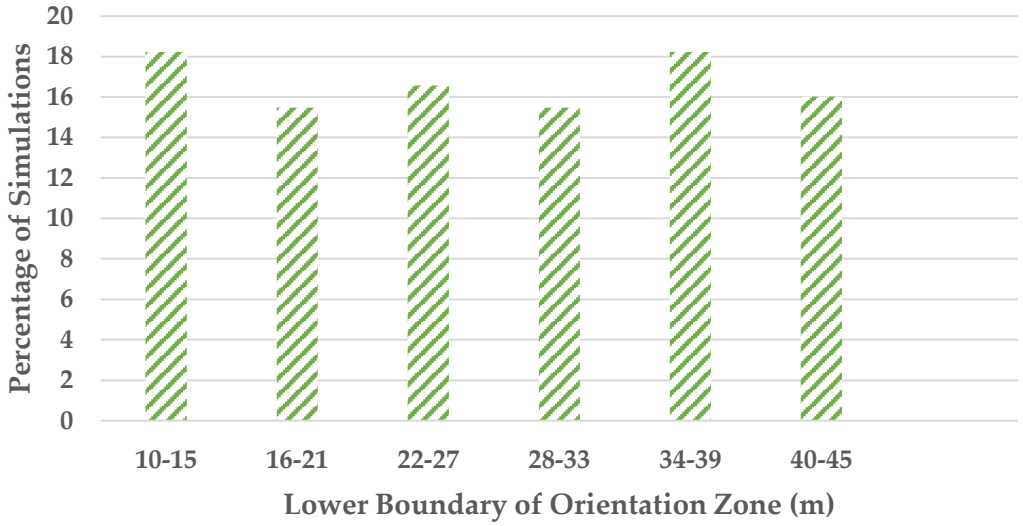


Figure 8.21 - Variation in orientation zone size used throughout simulation study.

8.5.1.3 Size of Orientation Zone during Recovery Phase

As discussed above, the purpose of incorporating a *Recovery* phase within this study is to demonstrate that the algorithm can successfully transition between the different control laws required throughout an oceanic sampling mission, i.e. *Deployment, Data Gathering & Recovery*,. In a real-life oceanic surveying mission, it is likely that the required inter-vehicle distance during this phase of the mission would be the same and would be evaluated based on the size of the vehicles used and what is deemed a suitable separation distance. Subsequently, for this study, the upper and lower boundaries of the *orientation* zone have been set to 6m and 11m respectively.

8.5.1.4 Initiation of Recovery Phase

The final parameter to be defined is the specific point during the mission when the *recovery* phase is initiated. In reality, this transition would occur when the distance between the location of the group's centre ($\bar{x}_{NN}, \bar{y}_{NN}$) and the vehicle's recovery position, (x_{des}, y_{des}) is below a certain threshold. However, for the work completed in this study, it has been decided that this transition would occur at the exact same time in each simulation. This decision was made to ensure the results could be processed and presented together in a similar manner to those shown in Chapter 5. Subsequently, for the simulations completed, each vehicle will transition into the *Recovery* phase at 3000s. This value has been chosen based on the results of Figure 7.11 where it has been shown that it takes less than 1500s for the vehicles to generate the required parallel line formation when the boundaries of the orientation zone are equal to 20m and 25m. Since the largest orientation zone used in this

study is exactly double these value, 3000s was deemed to be a logical value. Finally, once this transition occurs, the simulations continues for a further 3000s before terminating.

8.5.1.5 External Disturbances – Ocean Currents

As well as varying the algorithmic parameters discussed above, the simulations undertaken have also varied the parameters associated with the external disturbances. Table 8.5 demonstrates that the velocities selected for the ocean current are equal to 25%, 50% and 75% of the RoboSalmon vehicles nominal velocity. By using these three values, the results will be able to demonstrate whether or not an increase or decrease in the magnitude of the currents velocity affects the algorithms ability to operate as required.

Table 8.5- Variation in the values selected for the currents velocity.

Category	Magnitude of Current Velocity, V_c (m/s)
1	0.05
2	0.1
3	0.15

Furthermore, the values used for the direction of the ocean current are presented below in Table 8.6

Table 8.6 -- Variation in the values selected for the currents flow angle.

Category	Current Flow Angle, β_C
1	$\beta_c = \psi_{ref}$
2	$\beta_c = \psi_{ref} - 180^o$
3	$0 < \lvert\psi_{ref} - \beta_c\rvert < 90^o$

The value selected for the current flow angle in the first category has been made equal to the vehicle's reference heading angle. This has been deliberately chosen to test whether or not a tail current influences the algorithms ability to generate the required formation. Conversely, the value selected in the second category tests the performance of the algorithm when the vehicles are experiencing a head current. Finally, the third category selects a flow angle that results in each vehicle experiencing a current that moves laterally relative to the vehicles course angle.

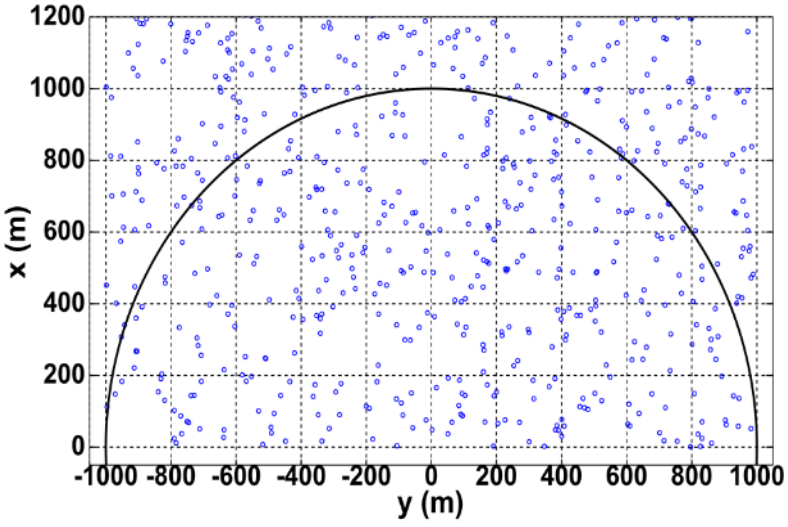
8.5.1.6 External Disturbances – Obstacles

In this study, four different obstacles sizes are tested: 4m, 6m, 8m and 10m. As with the parameters associated with the ocean currents, these obstacle sizes were chosen to demonstrate whether or not an increase or decrease in the obstacles size affected the algorithms ability to operate as required. As shown below in Table 8.7, these obstacles sizes are divided evenly across 181 simulations.

Table 8.7 - Variation in obstacle size across simulations.

Simulation Number	Obstacle Size (m)
1-46	4
47-92	6
93-138	8
139-181	10

As shown below in Figure 8.22, in order to ensure the obstacles were positioned in the path of the vehicles, a large number of obstacles were placed within the simulated environment. This was done specifically to guarantee that regardless of the vehicle's reference heading angle, there was a significant likelihood that a number of vehicles would have to manoeuvre past an obstacle.



(a)

Figure 8.22 - Simulation environment including obstacles.

8.5.2 Results

Section 8.5.1 above has described in detail the assumptions made, restrictions implemented and values used for the algorithmic parameters presented in Table 8.4. The aim of this section is to present the results obtained from these simulations and demonstrate whether or not the waypoint consensus algorithm is still able to operate as required despite having varied these parameters.

8.5.2.1 Vehicle Trajectories

To begin analysing the results, shown below in Figure 8.23 are the trajectories obtained from five of the simulations completed.

Overall, the trajectories demonstrate that the waypoint consensus algorithm has been able to generate the required parallel line formation during the *Deployment* phase of the mission. The trajectories also demonstrate that once the parallel line formation has been generated, the vehicles are subsequently capable of reducing their *lateral* nearest neighbour distance as required for the vehicle *Recovery* phase of the mission. These results suggest that overall, the waypoint consensus algorithm can operate successfully regardless of the algorithmic parameters selected.

However, upon closer inspection, it is apparent that apart from Figure 8.23 (a), none of the remaining trajectories have resulted in every vehicle being positioned directly alongside their nearest neighbour as per the criteria associated with the parameter, d_{abs}. For example, in Figure 8.23 (b), the three vehicles positioned at the right-hand side of the group are positioned significantly aft of the remaining members of the group. Likewise, in Figure 8.23 (c), the vehicles positioned at either side of the group are positioned significantly behind their nearest neighbours. This inability for every vehicle to converge to the required *absolute* nearest neighbour distance continues in Figure 8.23 (d) with the vehicles positioned to the extreme left of the group as well as throughout Figure 8.23 (e).

This variation in the convergence performance of the *absolute* and *lateral* nearest neighbour distances is consistent with the results presented previously and is primarily due to the design of the velocity control law implemented. As discussed in Chapter 7 (Section 7.2.2.6), each vehicle is only able to alter its velocity once per communication cycle and only for a predefined length of time, As shown above, this periodic change in the vehicles velocity results in the *absolute* distance between neighbouring vehicles converging very gradually.

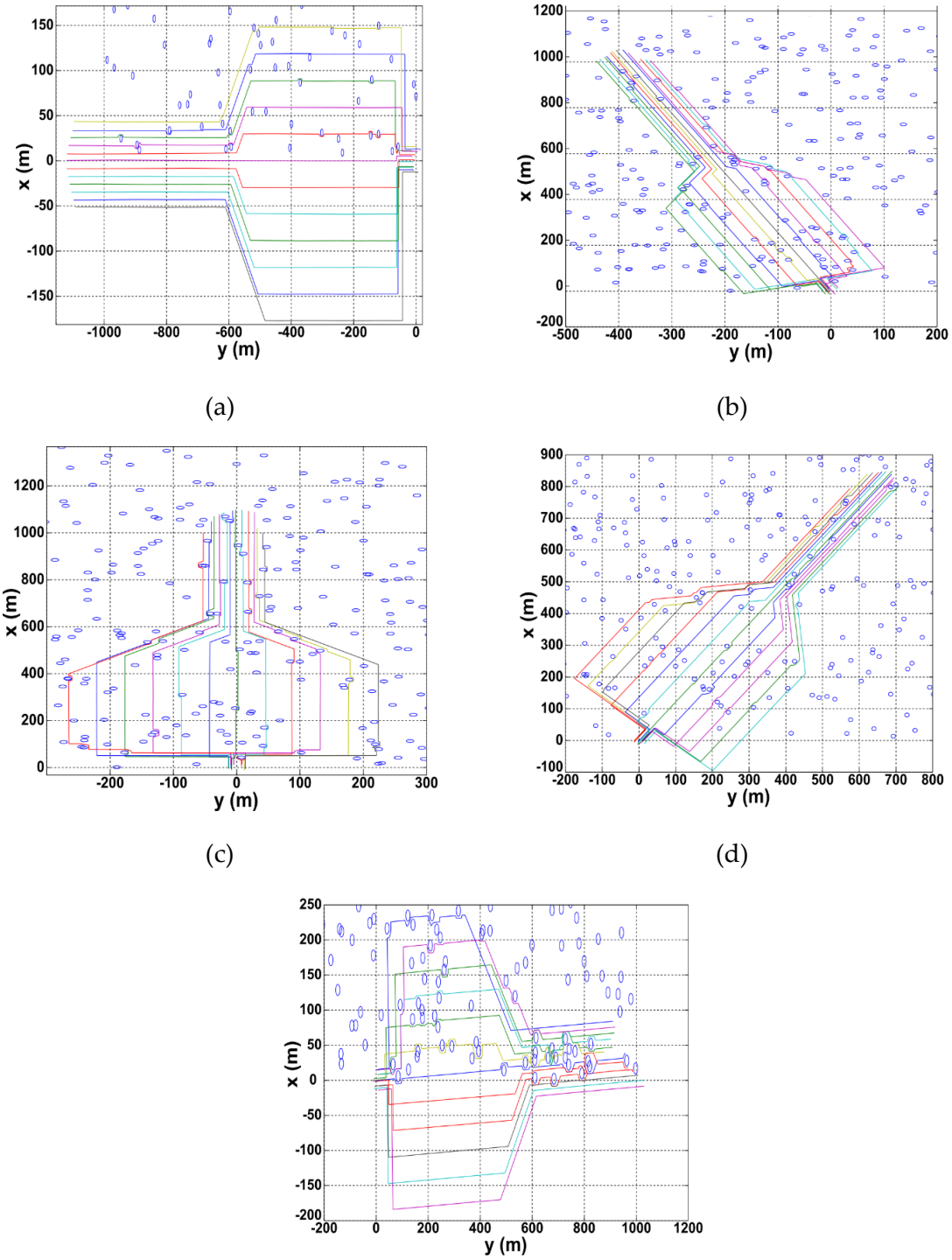


Figure 8.23 - Vehicle Trajectories obtained when reference heading angle is equal to (a) -90º (b) -21º (c) 0º (d) 39º and (e) 88º.

Figures 8.23 (a) –(e) also demonstrate that the obstacle avoidance technique implemented is operating as required with the vehicles - when necessary - manoeuvring out of the parallel line formation and subsequently around the stationary obstacle. Once the vehicle

has moved past the obstacle, Figures 8.23 (a) – (e) demonstrate that they are subsequently capable of re-joining the group in the required parallel line formation.

Therefore, the trajectories presented above suggest that the waypoint consensus algorithm is operating as expected despite varying the various algorithmic parameters presented in Table 8.4. However, as mentioned above, these trajectories only represent a small percentage of the 181 simulations completed. Consequently, the following sections will now analyse the results obtained across all simulations completed. In particular, these results will focus primarily on the effect of varying the parameters associated with the implementation of both stationary obstacles and oceanic currents as well as varying the size of the orientation zone.

8.5.2.2 Increasing Obstacle Size

The purpose of this section is to demonstrate whether or not the convergence performance of the waypoint consensus algorithm is influenced by the size of the obstacles incorporated within the simulation environment. To achieve this, shown below in Figure 8.24 is the evolution of the convergence performance of the parameters, d_1, d_2 and d_{abs} as the size of the obstacles are increased. Before analysing these results, it is important to note that the percentage values shown represent the percentage of vehicles from the simulations that contained obstacles of the same size. As shown above in Table 8.7, across the 181 simulations, each obstacles size was used in at least 45 of the simulations completed.

As expected, the results clearly demonstrate a sensitivity to the size of the obstacles incorporated with the overall percentage of vehicles satisfying the various convergence criteria decreasing as the size of the obstacles are increased. This is not surprising and is due to the fact that as discussed in Section 8.5.1.6, as the size of the obstacles were increased, the size of the simulation environment was kept constant. This means that the obstacles are covering an ever-increasing percentage of the simulation environment and as a result, the likelihood that more vehicles will have to complete more obstacle avoidance manoeuvres for longer periods of time are increased.

Nevertheless, the results demonstrate that regardless of obstacle size, the percentage of vehicles satisfying the relevant criteria continues to increase towards the required value throughout the simulations. This is particularly true for the parameters d_1, and d_2 where the convergence percentage (regardless of obstacle size) is generally within 10% of its required value. However, Figure 8.24 (d) illustrates a significant reduction in the percentage of vehicles satisfying the criteria associated with the parameter, d_{abs} with up to 50% of the

vehicles being unable to converge to a value within the specified limits. As discussed above, this is believed to be primarily due to the design of the velocity control law.

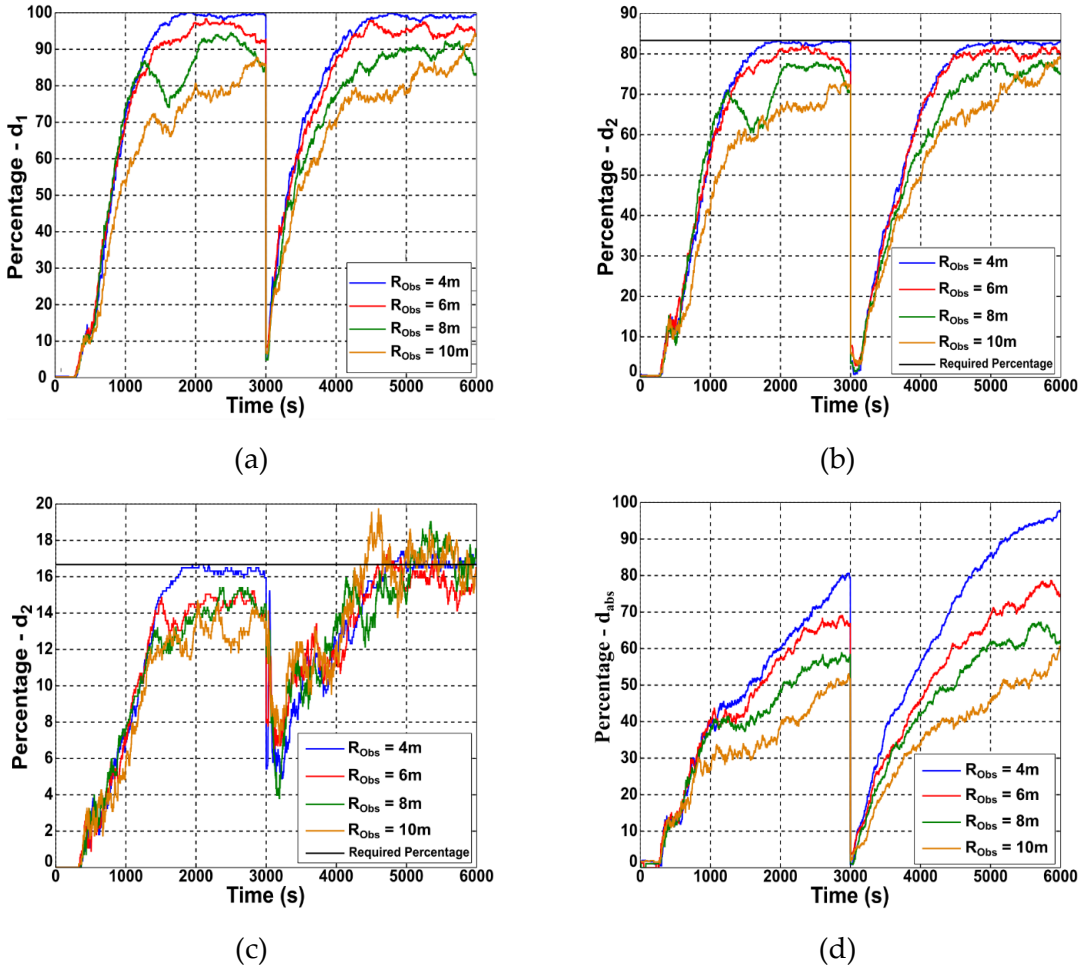


Figure 8.24 - Variation in the convergence performance of the parameters (a) d_1 (b) d_2 (c) d_2 and (d) d_{abs} as the size of the obstacles are increased.

However, as Figure 7.8 illustrates, another reason for this poor convergence is due to the fact that the waypoint consensus algorithm will not alter the vehicles velocity until the convergence criteria associated with d_1 and d_2 have been satisfied. Consequently, as Figures 8.24 (a) – (c) demonstrate, this will have been the case in several the simulations completed. Therefore, the inability of the vehicles to converge to the parallel line formation will inherently prevent the vehicles from positioning themselves directly beside their nearest neighbour.

However, it is important to note though that the behaviour described above is entirely as expected and demonstrates that the algorithm is capable of successfully undertaken obstacle avoidance manoeuvres This is proven by the fact that the evolution of the percentage values is continuously fluctuating across the simulations completed. This

behaviour is indicative of the vehicles overall satisfying the various convergence criteria but periodically not being able to due to undertaken obstacle avoidance manoeuvres.

8.5.2.3 Increasing Orientation Zone Size

In the previous section, the results illustrated that as the size of the obstacles incorporated increased, more vehicles were unable to continuously satisfy the various convergence criteria. As discussed, this is due to the vehicles having to periodically complete obstacle avoidance manoeuvres with the frequency of these manoeuvres increasing with obstacle size. However, as shown above in Table 8.4, a number of other algorithmic parameters were varied throughout the simulations completed. The purpose of this section therefore is to demonstrate what effect altering the size of the orientation zone had on the convergence performance of the algorithm.

As with the results presented in the previous section, this is achieved by analysing the variation in the convergence performance of the parameters, d_1, d_2, d_{abs} as well as the standard deviation of the vehicles heading angle, σ as the size of the orientation zone is increased. These results are shown below in Figure 8.25.

Overall, the only variation in the convergence performance across the different parameters is the time taken for the vehicles to begin satisfying the relevant criteria. The results clearly demonstrate that as the size of the orientation zones increases, the time taken for the vehicles to begin satisfying the relevant criteria increases too. Unsurprisingly, this is due to the vehicles having to travel a further distance before beginning to satisfy the relevant criteria. Furthermore, the results also demonstrate that regardless of the size of the orientation zone, each parameter converges to approximately the same percentage value. This can be attributed to the fact that throughout the simulation set up, the obstacle sizes were divided evenly across the different simulations (Table 8.7) and therefore their influence (as discussed in Section 8.5.2.2) should be approximately the same for each grouping shown in Figure 8.25.

Importantly, Figure 8.25 (b) demonstrates that as the size of the orientation zone is increased, the number of vehicles able to satisfy the relevant criteria decreases. Once again, this result is not surprising and is due two related factors. The first of which is the aforementioned inherently slow convergence rate of the velocity control law. The second and most important factor is that as the orientation zone size is increased, each vehicle has to travel an increasingly further distance before the relevant criteria associated with the parameters, d_1 and d_2 can be satisfied. Consequently, when these two factors are combined,

it should be expected that fewer vehicles will be able to satisfy the criteria associated with the parameter, d_{abs} throughout the same time frame.

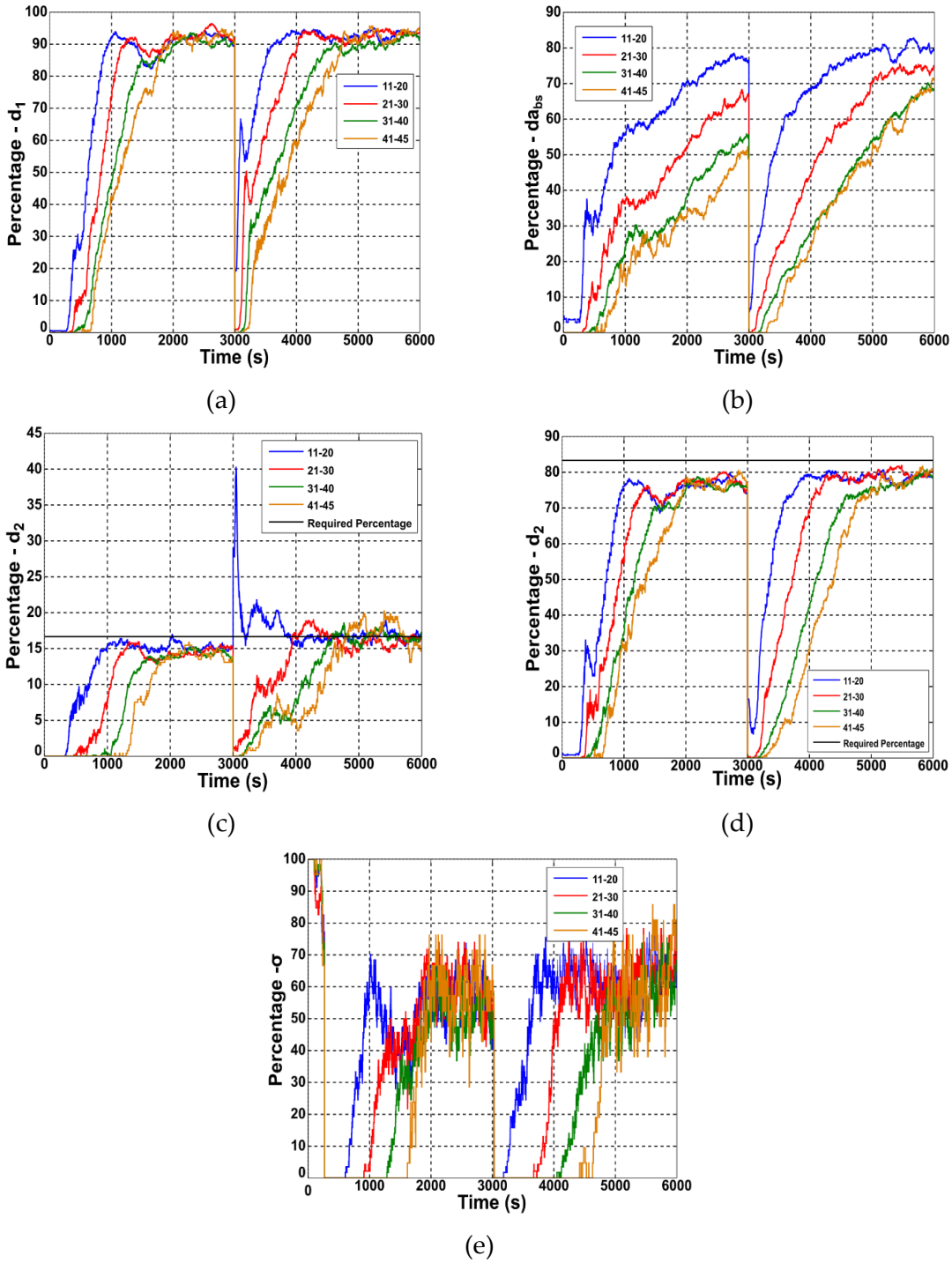


Figure 8.25 - Variation in the convergence performance of the parameters (a) d_1 (b) d_{abs} (c) d_2 (d) d_2 and (e) σ as the orientation zone size is increased.

Nevertheless, overall, the results presented in Figure 8.25 (b) illustrate a continuous increase in the percentage of vehicles satisfying the criteria associated with the parameter, d_{abs} throughout each mission stage. This result once again suggests that the waypoint consensus algorithm operated as expected throughout the simulations completed.

8.5.2.4 Varying Current Magnitude & Direction

The final parameters varied throughout the simulations completed were the velocity, V_c and direction, β_c of the ocean current. Subsequently, shown below in Figure 8.26 is the variation in the convergence performance of the parameters, d_1, d_2, d_{abs} and σ as the magnitude of the currents velocity, V_c is increased.

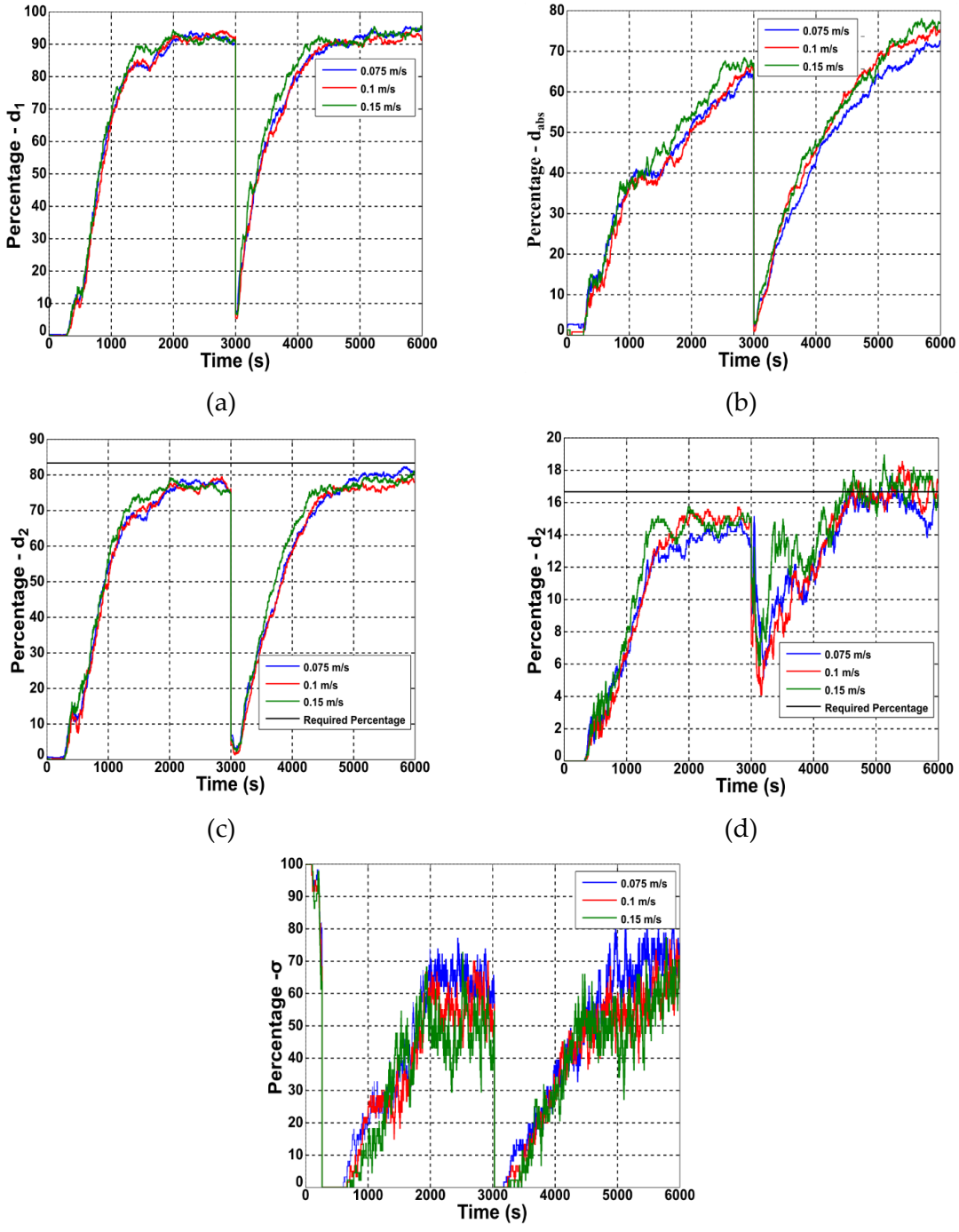


Figure 8.26 - Variation in the convergence performance of the parameters (a) d_1 (b) d_{abs} (c) d_2 (d) d_2s and (e) σ as the velocity of the ocean current is increased.

Similarly, shown below in Figure 8.27, is the variation in the convergence performance of these parameters, as the value used for the currents angle, β_c is varied.

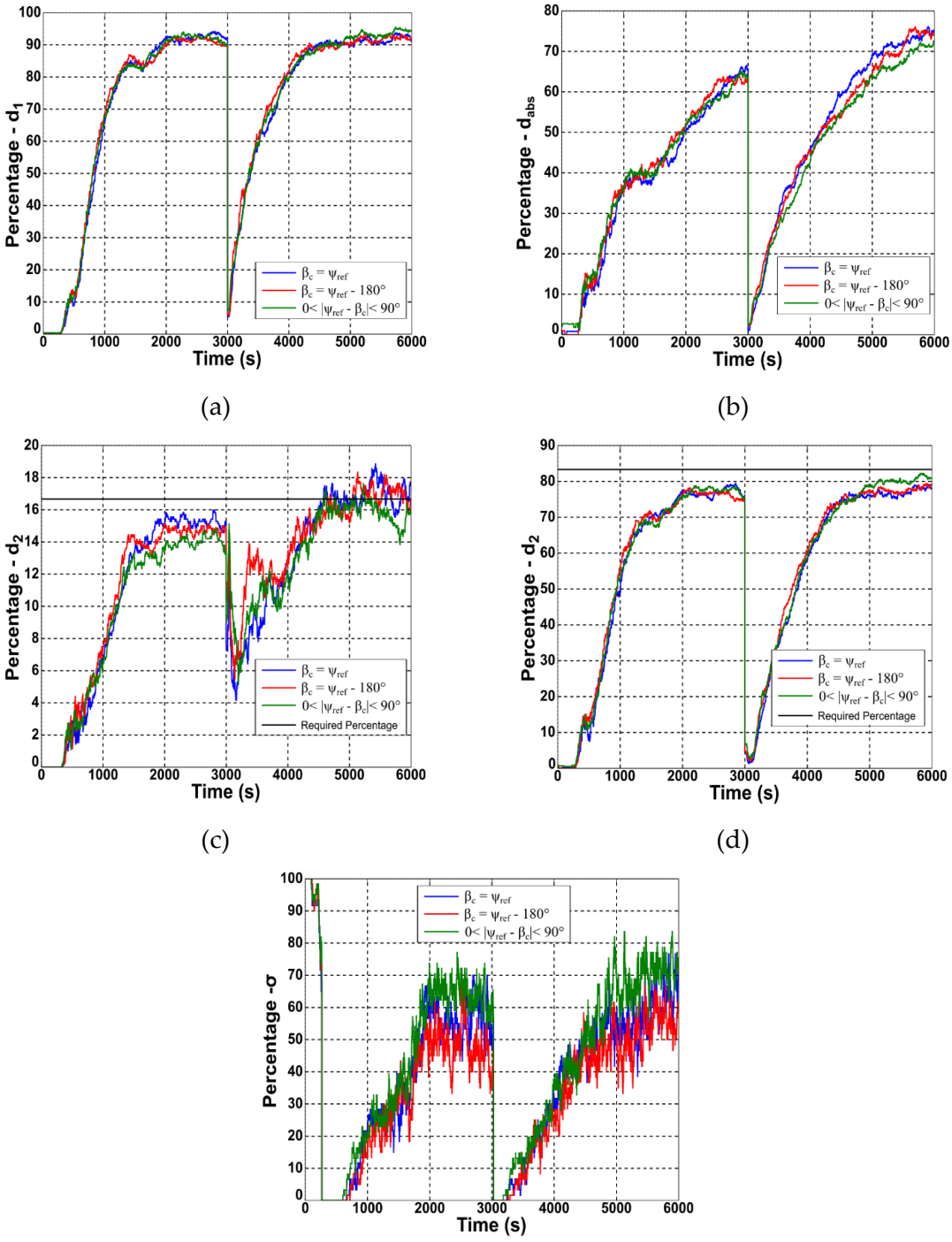


Figure 8.27- Variation in the convergence performance of the parameters (a) d_1 (b) d_{abs} (c) d_2 (d) d_2 and (e) σ as the angle of the ocean current is varied.

Unlike the results presented in the previous section, the results presented above demonstrate little variation in the convergence performance as the ocean currents angle varies. This is mainly due to the fact that as discussed in Chapter 8, each vehicle alters its

heading angle to compensate for the sideslip angle caused by the incorporation of ocean currents.

Therefore, while the changes made to each vehicle's heading angle will have to be greater and more often to compensate for the ocean current, overall, it should and does have little influence on the convergence performance of the algorithm. Based on the above finding, it is subsequently unsurprising to find that varying the direction of the ocean current also had little effect on the convergence performance of the algorithm as shown in Figure 8.27

8.6 Summary

The primary purpose of this chapter was to investigate what effect introducing external disturbances had on the ability of the trajectory prediction and waypoint consensus algorithms to coordinate the vehicles into the required parallel line formation. The subsequent investigation demonstrated that of the two algorithms, the waypoint consensus algorithm was more resilient, and as a result, better suited to coordinating vehicles in the unpredictable underwater environment. This is mainly due to the fact that unlike the trajectory prediction algorithm, the waypoint consensus algorithm doesn't rely on a continuously accurate representation of each vehicles positioning in order to operate successfully. Therefore, it was decided that for the remainder of this thesis, only the waypoint consensus algorithm would be used.

After the waypoint consensus algorithm had been selected as the preferred algorithm, the remainder of this chapter focussed on demonstrating what effect, if any, varying certain algorithmic parameters had on the algorithms ability to operate as required. These tests were undertaken to thoroughly test the robustness of the algorithm and demonstrate that it was capable of undertaken realistic oceanic surveying missions. While the results did demonstrate deviations in the algorithms convergence performance, these deviations were entirely as anticipated and due to vehicles having to periodically (and successfully) undertake obstacle avoidance manoeuvres as well as compensate for the oceanic current implemented.

With the robustness and comparative efficiency of the waypoint consensus algorithm having now been tested, the following chapter will now present the results obtained from simulations that replicated different realistic oceanic surveying scenarios.

Chapter 9

Simulation of Realistic Oceanographic Surveying Missions

9.1 Introduction

The previous chapter demonstrated that of the two algorithms presented in this thesis, the waypoint consensus algorithm's design is more resilient to the external disturbances likely to be found in the underwater environment. As well as the above, Chapter 8 also tested the robustness of the algorithm by varying certain algorithmic parameters as well as those associated with the external disturbances included in the simulations. The subsequent results demonstrated that regardless of the parameters used, the convergence performance of the algorithm was as anticipated and as a result, confidence in the ability of the algorithm to operate as required increased.

While these additional tests illustrated the algorithms ability to operate as required, the scenario length of the simulations was relatively short at less than two hours. Furthermore, as Figure 8.22 illustrates, the reference heading angle of the group, ψ_{ref} was kept constant throughout the entirety of the various simulations completed. However, in reality, oceanic surveying missions would not only be significantly longer, but the vehicles would also be required to manoeuvre to a number of different locations throughout a single mission.

The purpose of this chapter therefore is to demonstrate whether or not the algorithm is firstly capable of coordinating the vehicles for realistic scenario lengths and there afterwards, demonstrate whether it is also able to manoeuvre the vehicles to different locations while maintaining the required parallel line formation. To demonstrate this, the "figure-of-eight" trajectory presented previously in Chapter 3 will once again be employed as will the 'lawnmower' trajectory pattern shown in Figure 1.2.

To present this work, the chapter is structured as follows. Section 9.2 provides a brief overview of the algorithmic parameters used within in the simulations to ensure both the figure of eight and lawnmower trajectory patterns can be attempted. This section will also discuss the external disturbances incorporated within the simulations completed. This section will then conclude by providing an overview of the simulations completed. Thereafter, Section 9.3 analyses the results obtained from these simulations before Section 9.7 summarises the work completed in this chapter.

9.2 Realistic Oceanic Sampling Missions

As presented in Table 8.4, the following six parameters must be defined before the waypoint consensus algorithm can be tested within a realistic simulation environment:

- Position of Vehicle Recovery Area
- Deployment Orientation Zone Size
- Recovery Orientation Zone Size
- Magnitude of Ocean Current
- Direction of Ocean Current
- Radius & Number of Obstacles

The following two sections will now describe the values used for these that will allow the algorithm to coordinate the vehicles firstly through a figure of eight trajectory and there afterwards, a lawnmower search pattern.

9.2.1 Scenario Setup – Algorithmic Parameters

As shown below in Figure 9.1, the figure of eight trajectory can be generated by instructing the vehicles to manoeuvre through four desired locations (waypoints) that are equal to the four corners of a square.

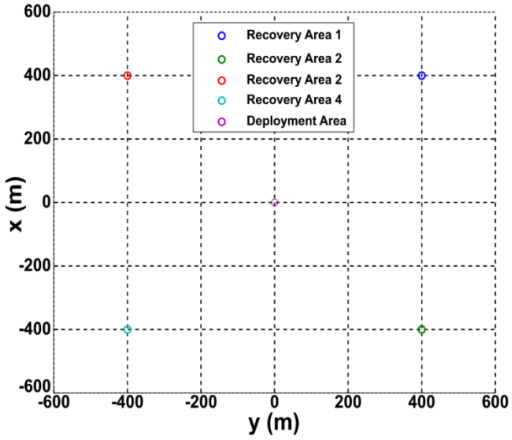


Figure 9.1 - Different recovery areas used throughout figure of eight scenario.

In order for the waypoint consensus algorithm to undertake the figure of eight trajectory, the following parameters must be defined for each phase of the mission:

- Coordinates of the groups desired location.
- Orientation Zone Size.

Once these parameters have been defined, the waypoint consensus algorithm will use Equation (5.13) to define the groups reference heading angle, ψ_{ref} for each mission phase. There afterwards, Equations (7.1)-(7.4) will define the individual waypoints for each vehicle. When this process is completed, the individual waypoints used during each phase of the mission will be similar to those shown below in Figure 9.2.

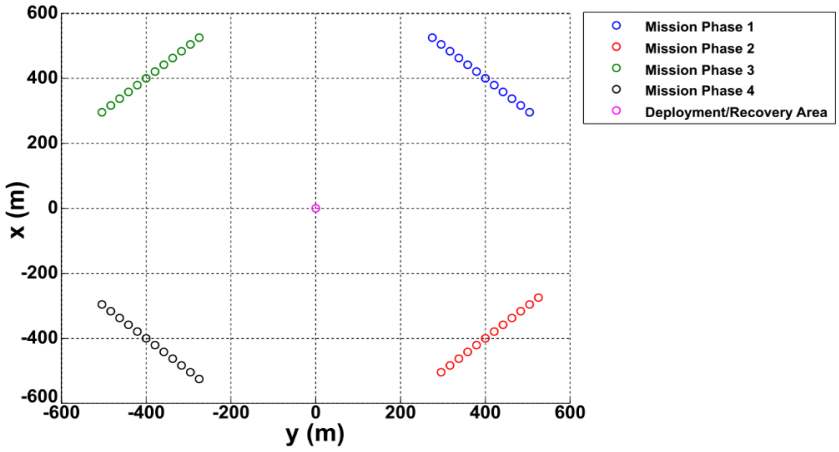


Figure 9.2 - Individual waypoints calculated using Waypoint Consensus Algorithm for "figure of eight" scenario.

Using a similar process to that described above, the waypoint consensus algorithm will generate the following individual waypoints for the lawnmower trajectory scenario.

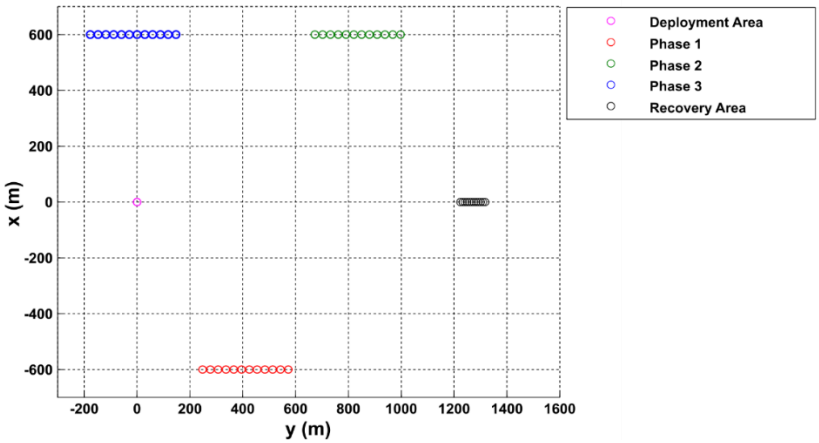


Figure 9.3 - Individual Waypoints calculated using Waypoint Consensus Algorithm for "lawnmower trajectory" scenario.

In this instance though, the reference heading angle of the group for each phase of the mission will need to be user defined and equal to the values shown below in Table 9.1.

Table 9.1- Representative reference heading angle values used during each phase of the "lawnmower" trajectory scenario.

Mission Phase	Reference Heading Angle
1	0°
2	180°
3	0°
4	180°

This is to ensure that when transitioning between the different phases, the vehicles will manoeuvre with a heading angle that is perpendicular to their next desired location and as a result, the vehicles should generate the required lawnmower pattern.

As well as defining the various desired locations, the required vehicle separation distance must also be defined. As discussed throughout this thesis, this distance is controlled through the selection of the orientation zone size. With the results presented at the end of Chapter 8 illustrating the algorithms ability to achieve various inter-vehicle separation distances, Table 9.2 below demonstrates that only two zone sizes were used for the work completed in this chapter.

Table 9.2 - Orientation zone sizes used throughout scenarios.

Mission Phase	Orientation Zone Size
Deployment	[27-32]m
Recovery	[6-11]m

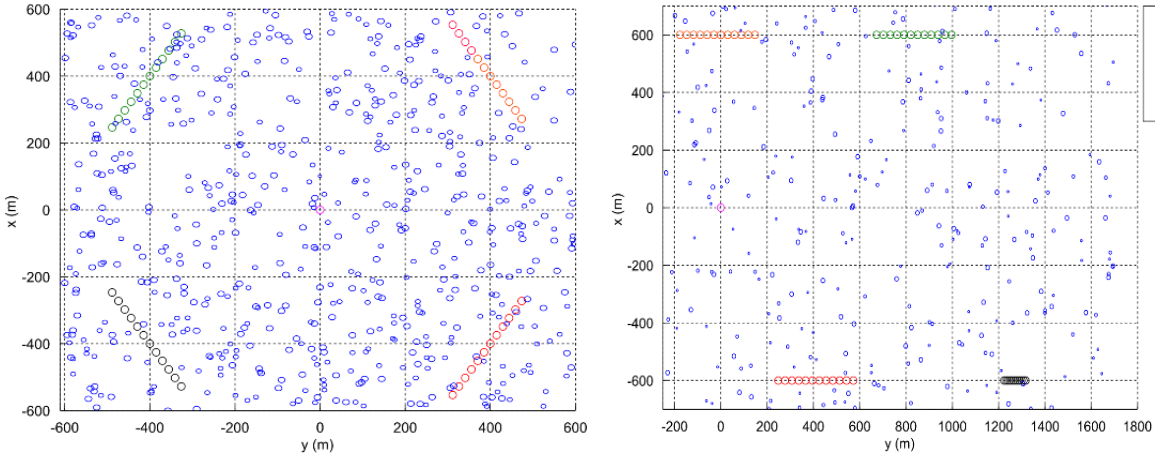
As with the simulations completed throughout this thesis, the zone sizes presented in Table 9.2 will ensure that each of the three behavioural control laws – *repulsion, orientation & attraction* will be used throughout the simulations completed in this study.

9.2.2 Scenario Setup – External Disturbances

In the previous section, the values adopted for the various algorithmic parameters for both the "figure of eight" and "lawnmower" trajectories were presented. This section will now

define the values used for the various parameters associated with the implementation of the two external disturbances, i.e. oceanic currents and external disturbances.

As with their inclusion in Chapter 8, the external disturbances included in these simulations are positioned within a specific area. As shown below in Figure 9.4, this area has been chosen to ensure that at any point, it is likely that at least one vehicle from the group will be required to complete an obstacle avoidance manoeuvre. Furthermore, unlike the simulations completed in Chapter 8, Figure 9.4 illustrates that the size of the obstacles contained within this area vary.



(a) (b)

Figure 9.4 - Simulation Environment with obstacles included for (a) figure of eight scenario and (b) lawnmower pattern.

This differs from those completed in Chapter 8 were the obstacle sizes were the same for each individual simulation but overall, they varied across the 181 simulations completed. Once again, this alteration has been included to further test the robustness of the waypoint consensus algorithm. Unsurprisingly, due to the larger simulation environment, the number of obstacles included within each scenario also increased.

For the inclusion of the oceanic currents, the values used for its velocity and direction are presented below in Table 9.3.

Table 9.3 - Direction and magnitude of oceanic current.

Scenario	Current Velocity (m/s)	Current Direction
Figure of Eight	0.15	135°
Lawnmower	0.15	135°

As with the simulations completed in Chapter 8, the magnitude and direction of the oceanic current has been kept constant throughout the scenarios completed. However, due to the fact that the groups reference heading angle will vary throughout the missions completed in this study, the relative angle between each vehicle's direction of travel and the oceanic current will vary far more than the simulations completed in Chapter 8. As with varying the size of the obstacles included, this will once again further test the robustness of the waypoint consensus algorithm.

9.2.3 Simulation Overview

The previous two sections have described the values chosen for both the algorithmic parameters as well the external disturbances for the simulations completed in this chapter. As shown below in Table 9.4, these parameters are used throughout six separate simulations completed as part of the investigation studying the feasibility of undertaking realistic oceanic scenarios with the waypoint consensus algorithm presented previously in Chapter 7.

Table 9.4 - Summary of simulations completed.

Simulation Number	Scenario	External Disturbance
1	Figure of Eight	None
2	Figure of Eight	Obstacles
3	Figure of Eight	Obstacles & Currents
4	Lawnmower	None
5	Lawnmower	Obstacles
6	Lawnmower	Obstacles & Currents

Table 9.4 illustrates that the "figure of eight" and "lawnmower" scenarios are each repeated three times. This has been undertaken to illustrate once again that the waypoint consensus is able to operate as required despite the introduction of the disturbances associated with the underwater environment.

9.3 Results

The purpose of the simulations completed in this chapter is to demonstrate whether or not the waypoint consensus algorithm is able to undertake realistic oceanic sampling missions. As the previous section has detailed, this has been undertaken by completing two sets of

simulations that firstly test the algorithms ability to manoeuvre the vehicles through a figure of eight trajectory and there afterwards, through a lawnmower trajectory pattern.

With the main aim of this chapter being to investigate the algorithms ability to manoeuvre the vehicles through the aforementioned trajectory patterns, the analysis presented below will be restricted to a comparison of the resulting trajectories obtained from the two sets of simulations completed. This restriction is due to the fact that apart from demonstrating the algorithms ability to change the groups desired location, no further insight will be obtained by undertaken the detailed analysis presented previously in Chapters 7 and 8.

9.3.1 Figure of Eight Analysis

Shown below in Figure 9.5 are the trajectories obtained from the three simulations completed as part of the figure of eight scenario. These simulations were completed to demonstrate the algorithms ability to generate the required pattern with and without the inclusion of the external disturbances discussed throughout this work.

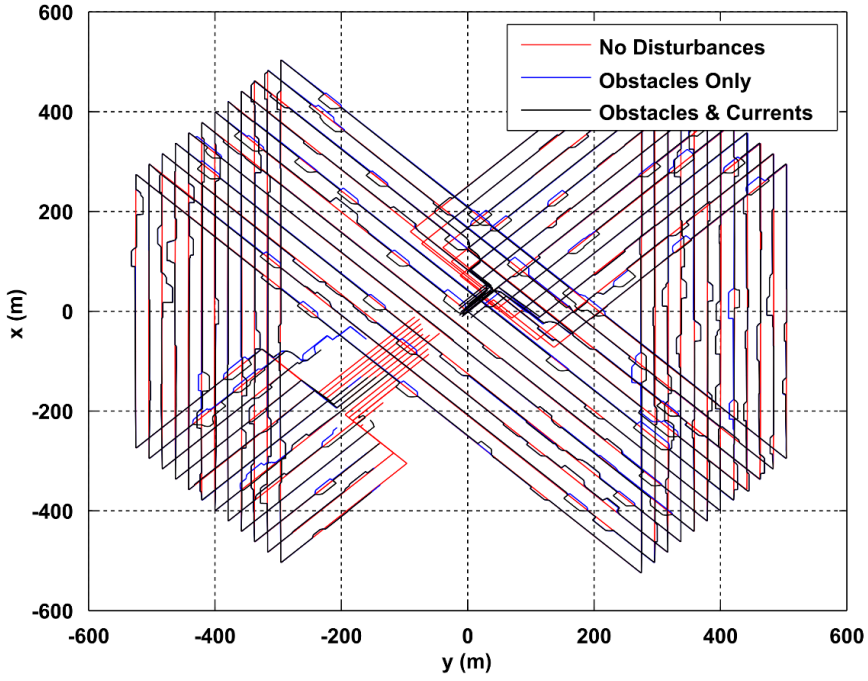


Figure 9.5 - Trajectories obtained from figure of eight simulations

The trajectories presented above clearly demonstrate that the waypoint consensus algorithm has successfully coordinated the vehicles to generate the figure of eight pattern while simultaneously maintaining the required parallel line formation. The results demonstrate that while there are differences in the trajectories obtained with and without the inclusion of external disturbances, the required trajectory is nevertheless generated. These results clearly illustrate the ability of the algorithm to successfully coordinate the

vehicles to different locations throughout a single mission regardless of the local operating conditions.

However, although the trajectories presented above demonstrate that the algorithm has been able to generate and maintain the required parallel line formation, not all of the vehicles have been unable to manoeuvre directly alongside their nearest neighbour. As has been the case with the waypoint consensus algorithm, this is due to the design of the velocity control law used within the algorithm and the fact that each vehicle can only alter its velocity once every communication cycle, i.e. approximately once every five minutes.

9.3.2 Lawnmower Trajectory Analysis

As with the results presented in the previous section, shown below in Figure 9.6 are the trajectories obtained from the lawnmower scenarios with and without the inclusion of the different external disturbances.

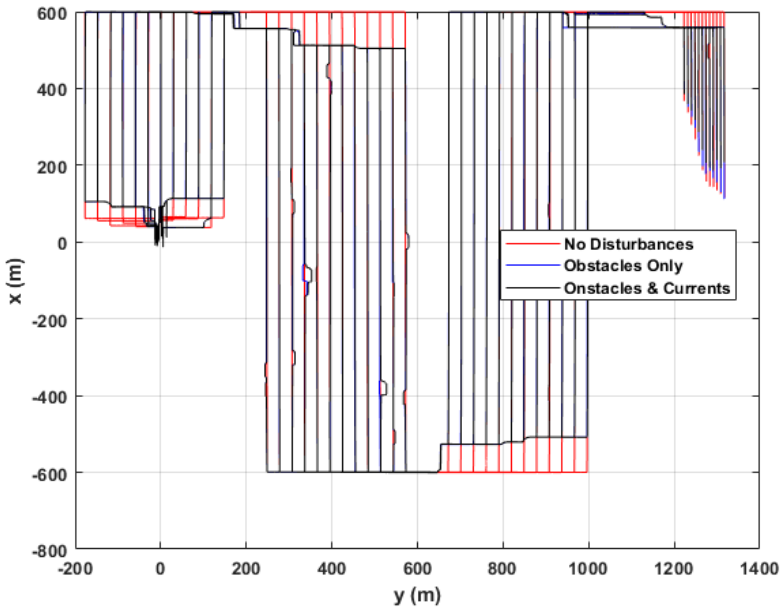


Figure 9.6- Trajectories obtained from lawnmower pattern simulations.

Similarly to the results presented in Section 9.3.1, the trajectories presented above once again illustrate the ability of the algorithm to successfully manoeuvre the vehicles to different locations while ensuring (where possible) the vehicles maintain the required parallel line formation. Also apparent from Figure 9.6 is the variation in the trajectories obtained as a result of the external obstacles been included in the simulations. Again, although noticeable, these variations are as expected and once again demonstrate the robustness of the algorithm to these disturbances.

However, it is also important to note that like Figure 9.5, Figure 9.6 also illustrates the inability of the waypoint consensus algorithm to ensure that each vehicle can manoeuvre directly alongside their nearest neighbour. As discussed above, this is due to the design of the velocity control law within the waypoint consensus algorithm.

Nevertheless, as required, the results presented above in Figures 9.5 and 9.6 demonstrate that the waypoint consensus algorithm is able to successfully coordinate the vehicles to a number of different locations throughout a single scenario. Furthermore, the results also demonstrate that the algorithm will ensure the vehicles maintain the required parallel line when possible.

9.4 Summary

The aim of this chapter was to demonstrate whether or not the waypoint consensus algorithm was capable of successfully coordinating the vehicles to a number of different locations during the course of a single mission. To achieve this, two sets of simulations were undertaken. The first set required the algorithm to manoeuvre the vehicles through a figure of eight pattern while the second group of simulations tested the algorithms ability to undertake the lawnmower trajectory pattern. Both sets of simulations tested the algorithms ability in this regard with and without the inclusion of the external disturbances presented in Chapter 8.

Overall, the results illustrated that the algorithm is indeed capable of successfully completing both the figure of eight and lawnmower trajectories .Moreover, the results also show that the algorithm ensures the vehicles either maintain or regenerate the required parallel line formation while manoeuvring between the different locations. Importantly, these conclusions are also applicable to the simulations where the various external disturbances were taken into consideration. Therefore, apart from the known inability of the algorithm to ensure each vehicle manoeuvres directly alongside their nearest neighbours, the results presented in this chapter illustrate, as required, the algorithms ability to successfully manoeuvre the vehicles to a number of different locations during a single missions. In doing so, these results further illustrate not only the algorithms ability to successfully coordinate multiple vehicles in the restrictive underwater communication channel but also its robustness to the disturbances experienced in this unpredictable environment.

Chapter 10

Conclusions & Future Work

10.1 Conclusions

The overall aim of the research presented in this thesis has been to develop a formation control algorithm capable of successfully coordinating a group of BAUVs to undertake oceanic surveying missions despite the constraints of the underwater milieu. The motivation for undertaking this study and establishing whether or not such an algorithm could be created was due to the significant improvements in mission efficiency (i.e. larger area coverage) that would be achieved in doing so.

Importantly, the formation control algorithm created would not only need to generate the required parallel line formation regardless of the vehicles initial positioning but also guarantee that the same vehicles could then safely navigate to a surface vessel for recovery. In the initial stages of this research, these three mission phases, i.e. *Deployment, Mapping & Recovery* were found to be analogous to the behavioural mechanisms of fish partaking in schooling behaviour. As a result, and continuing the biologically inspired theme introduced by the RoboSalmon BAUV, the initial formation control algorithm presented in this thesis replicated these behavioural mechanisms. Throughout this thesis, this initial biomimetic algorithm was continuously developed in an attempt to improve its suitability to operate as required despite the nuances of the underwater environment. However, before the algorithm could be developed and tested, a suitable mathematical model of the RoboSalmon vehicle had to be created.

In Chapter 3, the pre-existing high-fidelity mathematical model of the RoboSalmon vehicle has been presented and its unsuitability to efficiently model multiple vehicles simultaneously demonstrated. The results demonstrated that the execution time for a simulation involving multiple vehicles was excessively long and would be impractical for analysing the proposed formation control algorithm. To reduce the execution time, two techniques were considered - Look Up Tables (LUTs) and Artificial Neural Networks (ANNs). These replaced the complex and processor intensive functionality associated with

modelling the motion of RoboSalmon's actuated tail section. By replacing this complex functionality with simpler mathematical basis functions (ANNs) and indexing operations (LUTs), the results presented in Chapter 3 firstly demonstrated the ability of these simpler models to maintain the accuracy of the original. Thereafter, the results illustrated the ability of these simpler models to reduce the simulation execution time of the model by 90% when using the LUTs and 85% when an ANN was implemented. Unsurprisingly, this significant reduction in simulation execution time meant that multiple vehicles could be simulated simultaneously in a drastically more time efficient manner. While both techniques operated as envisioned, the results demonstrated that the LUT method was not only faster but also more accurate. As a result, it was this particular mathematical model that was used throughout the remainder of this study to model the dynamics of the RoboSalmon vehicle and hence, develop the formation control algorithm.

The development of the formation control algorithm began in Chapter 4 by briefly describing the behavioural mechanisms used by fish within schooling structures. There afterwards, a detailed description of how these behaviours were transformed into a suitable formation control algorithm was presented. This initial algorithm contained three heading control laws that would manoeuvre each vehicle in either a *repulsive, orientating or attractive* manner dependent on the *absolute* distance between each vehicle and their nearest neighbours. In addition to these heading control laws, a simple velocity control law was also incorporated. While the results presented in Chapter 4 demonstrated the ability of the algorithm to generate a stable group formation, they also demonstrated that the performance of the various heading control laws varied dependent on the number of nearest neighbours taken into consideration. The results also illustrated that the algorithm was incapable of coordinating the vehicles into the parallel line formation required for efficient oceanic surveying. This chapter concluded by providing a number of recommended alterations that if incorporated, should improve the performance of the algorithm.

Chapter 5 began by detailing the alterations made to the algorithm based on the recommendations made at the end of Chapter 4. These alterations included using the *lateral* distance (instead of the *absolute* distance*)* when deciding which of the three heading control laws to implement. Furthermore, a reference heading angle was introduced in Chapter 5 that not only provided the algorithm with the ability to manoeuvre the vehicles to any desired location but also provide a single heading angle that every vehicle could manoeuvre with when using the *orientating* control law. Minor changes were also made to the *repulsive* and *attractive* control laws to ensure that as well as generating the required

parallel line formation as efficiently as possible, the vehicles could also converge to a recovery area at the end of the mission. In order to demonstrate whether or not the changes made were successful, the modified formation control algorithm was tested using the exact same simulations as undertaken in Chapter 4. The subsequent results clearly demonstrated that the modifications made operated as envisioned with the *repulsive* control law ensuring the generation of the parallel line formation, the *orientating* control law maintaining this formation and the *attractive* control law allowing the vehicles to subsequently converge to a smaller recovery area.

With Chapter 5 illustrating the ability of the modified formation control algorithm to operate as required, the main aim of Chapter 6 was to detail the implementation of a realistic representation of the underwater communication channel and analyse what effect its introduction had on the efficiency of the algorithm. To achieve this, the beginning of Chapter 6 detailed the functionality implemented to ensure an accurate representation of the well-known communication protocol, TDMA was incorporated within the simulations. By incorporating this particular protocol, each vehicle would now only have access to up-to-date positional information (of neighbouring vehicles) on a periodic basis. The period between these updates was called the Communication Cycle Length and the remainder of Chapter 6 detailed the results obtained from an extensive simulation study analysing what effect increasing this parameter's value had on the algorithms ability to coordinate the vehicles as required. The results demonstrated that as the communication updates became less frequent, the algorithm was surprisingly still capable of coordinating the vehicles as required. As detailed extensively in Chapter 6, this continued ability was due to a combination of the geometry associated with evaluating the *lateral* nearest neighbour distance, the periodic nature of the TDMA protocol and the design of the *attractive* control law. However, unsurprisingly, the results also demonstrated that as the Communication Cycle Length was increased, the efficiency of the algorithm steadily deteriorated. This deterioration resulted in an approximate 900% increase in the convergence time of the algorithm when realistic values for the Communication Cycle Length were incorporated. Furthermore, due to the differing geometry associated with evaluating the *absolute* nearest neighbour distance, the results also demonstrated that convergence of this parameter could no longer be guaranteed.

With Chapter 6 illustrating a significant decline in the algorithms efficiency, the aim of Chapter 7 was to detail the implementation of two methodologies designed specifically to allow the algorithm to operate successfully and as efficiently as possible regardless of the underwater communication channel. The first methodology used a prediction algorithm to

provide each vehicle with an estimate of its nearest neighbour's position in the interim period between subsequent communication updates. This prediction functionality contained not only an identical copy of the RoboSalmon mathematical model presented in Chapter 3 but also an identical version of the modified formation control algorithm described in Chapter 5. In contrast, the second methodology contained zero predictive functionality but instead used a waypoint consensus algorithm to provide each vehicle with a unique waypoint to navigate towards. By implementing this consensus algorithm, it was envisioned that the required parallel line formation could be generated without the individual vehicles requiring up-to-date positional information of its neighbouring vehicles. The subsequent simulations undertaken to test these two methodologies were deliberately chosen to be identical to those completed in Chapter 6. This was to ensure a fair comparison of the results obtained with and without these two methodologies implemented. Overall, the results suggested that the predictive methodology provided the better solution with the formation control algorithm satisfying all necessary convergence criteria as if the communication among the vehicles was instantaneous, i.e. the results were nearly identical to those presented in Chapter 5. Conversely, the results obtained from the waypoint consensus algorithm demonstrated that while the various convergence criteria could be satisfied, the convergence time increased proportionality with the Communication Cycle Length. This resulted in the predictive methodology achieving the required formation 2-3.5 times faster than the waypoint consensus algorithm.

While the results presented in Chapter 7 suggested that the predictive methodology provided the best solution, the aim of Chapter 8 was to illustrate whether or not this was still the case when additional environmental disturbances such as stationary obstacles and ocean currents were incorporated. As a result, Chapter 8 began by detailing the additional functionality implemented in order to ensure a realistic representation of these external disturbances were incorporated within the simulations. Thereafter, the same simulations as completed in Chapter 7 were undertaken but this time with the aforementioned external disturbances incorporated. The results shown in Chapter 8 demonstrate quite clearly that the performance of the waypoint consensus algorithm is less affected by the inclusion of these external disturbances. The reason that this is the case is due to the fact that the waypoint consensus algorithm does not rely on an accurate representation of its nearest neighbours positioning and instead, only needs to accurately track its chosen (stationary) waypoint. In contrast, the predictive functionality, because of the introduction of these additional external disturbances (which it cannot predict) is no longer capable of always providing each vehicle with an accurate representation of the positioning of neighbouring vehicles. As with the results presented in Chapter 6, this resulted in the algorithm altering

the heading angle and velocity of each vehicle based on incorrect information. Unsurprisingly, this resulted in these vehicles then having to undertake subsequent corrective manoeuvres when a communication update from its nearest neighbour(s) was received. Therefore, based on the results presented throughout Chapter 8, it was decided that overall, the waypoint consensus algorithm provides the most robust solution in terms of overcoming the constraints and unpredictable nature of the underwater environment.

In order to allow a fair comparison of the results obtained throughout the formation control algorithms development, i.e. Chapters 4-8, a number of user defined algorithmic parameters had so far remained unchanged. These parameters included the required heading angle of the group, the size of the three behavioural zones, the initial vehicle positions, the number and size of the obstacles included within the simulation environment as well as the magnitude and direction of the ocean currents. Furthermore, while the three behavioural controls laws had been thoroughly tested, a full oceanic surveying mission that transitioned through the *Deployment, Mapping* and *Recovery* phases had yet to be undertaken. Subsequently, the purpose of the final simulations presented in Chapter 8 was to demonstrate the ability of the algorithm to operate as required regardless of the values used for these various algorithmic parameters. To achieve this, the various values used for these parameters and within what thresholds they were tested was subsequently presented and tested. The subsequent results obtained from these simulations demonstrated that the formation control algorithm using the waypoint consensus methodology was indeed able to operate as required regardless of both the external disturbance included as well as the values selected for the various user-defined algorithmic parameters.

The purpose of Chapter 9 was to build on the results obtained in Chapter 8 and demonstrate that the formation control algorithm was not only capable of forming the required parallel line formation but also capable of manoeuvring the group to a number of different locations throughout a single mission. To achieve this, two scenarios that involved the vehicles manoeuvring through a figure of eight pattern and then a lawnmower pattern were created. These scenarios were also simulated with and without the presence of the external disturbances used in Chapter 8. As anticipated, the results demonstrated the ability of the algorithm to manoeuvre the vehicles through the different patterns while maintaining the required parallel line formation were possible. Furthermore, similarly to the results presented in Chapter 8, the inclusion of the external disturbances had only the expected effect on the algorithms performance to operate as required.

Overall, the work presented throughout this thesis has demonstrated that the original aims and objectives of this thesis as described in Chapter 1 have been satisfied. This is shown

below where the pertinent conclusions, contributions and recommendations from this work are summarised:

- The ability of system identification techniques such as ANNs and LUTs to drastically improve the simulation execution time of a mathematical model without noticeably altering its accuracy.
- While imitating the behavioural mechanisms of fish allows the formation of a stable group structure, the simplicity of these behaviours prevents the generation of the required parallel line formation.
- The ability of this imitating formation control algorithm also resulted in varying algorithmic behaviour depending on the values used for certain algorithmic parameters.
- By making a number of changes to the algorithm based on the behavioural mechanisms of fish, this work has demonstrated the ability to create a formation control algorithm capable of generating a group formation capable of undertaken realistic oceanic mapping missions.
- This work has also demonstrated that the algorithm, albeit with increasing difficulty is still capable of generating the required formation despite the introduction of a realistic representation of the underwater communication channel by the inclusions of the TDMA communication protocol.
- This work has demonstrated that introducing predictive functionality to overcome the limitations associated with the communication channel only operates effectively in an undisturbed environment that contains no external obstacles.
- Most importantly, the results demonstrate that a formation control algorithm capable of achieving group consensus and requiring as little communication as possible provides the most robust solution to overcoming the various limitations associated with the underwater environment.

In concluding, the work presented in this thesis has presented the design and development of a formation control algorithm capable of successfully coordinating a group of BAUVs in the unpredictable and restrictive nature of the underwater environment. In particular, this thesis has demonstrated that due to these particular characteristics, the implementation of predictive methodologies to track the positioning of neighbouring vehicles and hence, overcome the nuances associated with the underwater environment, does not present a robust solution. Instead, the results indicate that in order to create a formation control algorithm that is resilient to the complexities of operating in the underwater environment, the algorithm should be designed specifically not to rely on or require up-to-date positional

information of neighbouring vehicles. This has been achieved by the implementation of a waypoint consensus algorithm. As the results demonstrate, this algorithm only requires each vehicle to receive a single communication update from each neighbouring vehicle before the algorithm has the necessary information to ensure the generation of the required parallel line formation.

10.2 Future Work

The work presented in this thesis has detailed the development of an algorithm capable of generating a particular formation regardless of the limitations of the local operating environment. Going forward, there are numerous areas of future work for the research completed in this thesis. In the immediate future, alterations could and should be made to the velocity control law in order to improve its efficiency and allow vehicles to manoeuvre alongside one another in a timelier manner. Further testing could also be completed to demonstrate what effect – if any - increasing the group size has on the algorithms ability to operate as required. Furthermore, investigations into analysing what effect different deployment strategies would have on the algorithm's ability/efficiency to operate as required should also be completed. For example, at present, every vehicle is already positioned in the water when the algorithm is activated. However, a different deployment strategy could be to deploy the vehicles from the surface ship one by one and have the algorithm activated as soon as the first vehicle enters the water. As well as testing different deployment strategies, the applicability of the algorithm to different types of underwater vehicle could also be investigated. Therefore, a potential area of future work would be to implement the formation control algorithm within a simulation environment containing a different type of underwater vehicle, i.e. an AUV, AUG or Hybrid AUV.

As well as the above, it would also be advantageous to investigate the feasibility of introducing some form fault detection and reconfiguration functionality into the algorithm. This functionality should aim to allow the vehicles to firstly identify that one (or more) of the vehicles within the group has a fault and thereafter achieve consensus on a suitable reconfiguration strategy. This reconfiguration should aim to manoeuvre the vehicles into a formation that minimises the loss in the area mapped resulting from the loss of a vehicle or vehicles.

One final area of potential future work would be investigating the feasibility of implementing an adaptive sampling functionality into the algorithm. In doing so, this would provide the vehicles with the autonomy to alter their trajectory in order to obtain the most useful data possible throughout a mission. For example, in a search and recovery

mission, if a number of areas of interest are discovered at one side of the group, the vehicles positioned at the other side of the group should alter their trajectories and reconfigure the group's formation in order to maximise vehicle coverage at known sites of interest. An alternative application of this adaptive functionality would be to allow the vehicles to use their on board sensors to measure the concentration of a particular substance and alter the groups formation based on the associated data obtained from the sensors from across the group. This would allow groups of AUVs to be used to monitor and provide accurate assessments of the evolution of events such as oil spills.

The areas of development presented above have been suggested in such an order that if they were to be applied sequentially, the efficiency of the baseline formation control algorithm would be improved first. Thereafter, the successful implementation of fault detection and reconfiguration functionality as well as the introduction of adaptive sampling techniques would drastically improve the autonomy of the overall system and as a result, further improve the efficiency and usability of the algorithm.

A: RoboSalmon Mathematical Model

This appendix presents the derivation of the mathematical model used to represent the dynamics of the RoboSalmon vehicle. The mathematical model of RoboSalmon can be represented using the following Equation:

$$\mathbf{M}\dot{\boldsymbol{v}} + \boldsymbol{C}(\boldsymbol{v})\boldsymbol{v} + \boldsymbol{D}(\boldsymbol{v})\boldsymbol{v} + \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} \quad \text{(A1.1)}$$

Here **M** is the mass/inertia matrix, **C** is the Coriolis matrix, **D** is the damping matrix, **ν** is the state vector containing the Body-Fixed velocities, **τ** is the input force/moment vector, **g** is the gravitational force/moment vector which is zero due to the assumption of neutral buoyancy, **η** represents the Earth-Fixed dynamic variables and **J** is the Euler kinematic transformation matrix. Therefore, the purposes of this appendix is to describe the functionality contained within the individual matrices of the above equation.

A1 – Inertial Matrix

The inertia matrix, **M** in Equation (A1.1) consists of two components: the rigid body inertia and the added mass inertia due to the vehicle operating underwater. Consequently, the matrix, **M** can be defined as the sum of these two components as shown below in Equation (A1.2).

$$\mathbf{M} = \mathbf{M_{RB}} + \mathbf{M_A} \quad \text{(A1.2)}$$

Where $\mathbf{M_{RB}}$ represents the rigid-body inertia properties of the vehicle and $\mathbf{M_A}$ represents the properties of the added mass inertia. Therefore, as Equation (A1.1) demonstrates that the overall inertia matrix, **M** is multiplied by the accelerations of the vehicle $\dot{\boldsymbol{v}}$, the components of the rigid body inertia matrix $\mathbf{M_{RB}}$ can be defined as shown below in Equation (A1.3).

$$M_{RB} = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_x & 0 & 0 \\ 0 & 0 & 0 & 0 & I_y & 0 \\ 0 & 0 & 0 & 0 & 0 & I_z \end{bmatrix} \tag{A1.3}$$

As discussed above, the second term in Equation (A1.2) is used to model the added mass effect caused by the RoboSalmon vehicle operating in the underwater environment. As discussed in (Fossen,1994), the matrix, $\mathbf{M_A}$ is taken from the equation used to evaluate kinetic energy of a fluid and its contents are defined below in Equation (A1.4)

$$\mathbf{M_A} = -\begin{bmatrix} X_{\dot{u}} & X_{\dot{v}} & X_{\dot{w}} & X_{\dot{p}} & X_{\dot{q}} & X_{\dot{r}} \\ Y_{\dot{u}} & Y_{\dot{v}} & Y_{\dot{w}} & Y_{\dot{p}} & Y_{\dot{q}} & Y_{\dot{r}} \\ Z_{\dot{u}} & Z_{\dot{v}} & Z_{\dot{w}} & Z_{\dot{p}} & Z_{\dot{q}} & Z_{\dot{r}} \\ K_{\dot{u}} & K_{\dot{v}} & K_{\dot{w}} & K_{\dot{p}} & K_{\dot{q}} & K_{\dot{r}} \\ M_{\dot{u}} & M_{\dot{v}} & M_{\dot{w}} & M_{\dot{p}} & M_{\dot{q}} & M_{\dot{r}} \\ N_{\dot{u}} & N_{\dot{v}} & N_{\dot{w}} & N_{\dot{p}} & N_{\dot{q}} & N_{\dot{r}} \end{bmatrix} \tag{A1.4}$$

Where the SNAME notation is used to explain the definition of each of the coefficients (added mass derivatives). For example, $X_{\dot{u}}$ represents the rate of change of the force in the x direction with respect to the acceleration in the same axis.

However, a number of assumptions can be made that simplifies not only the definition of the individual components of Equation (A1.4) but also the overall matrix. The first assumption is that due to the low speed of the RoboSalmon vehicle, the individual components of Equation (A1.4) can be considered to be constant. Secondly, if the vehicle has multiple planes of symmetry, the off-diagonal components of Equation (A1.4) can be removed. As discussed in (Naddi, 2009), the RoboSalmon vehicle is assumed to take the shape of a prolate ellipsoid which has multiple planes of symmetry and as a result, the off-diagonal terms in Equation (A1.4) can be removed. Therefore, the matrix in Equation (A1.4) can be significantly simplified to the form shown below in Equation (A1.5) with the individual values used for each of the components also presented below in Table A1.1.

$$\mathbf{M_A} = \begin{bmatrix} -X_{\dot{u}} & 0 & 0 & 0 & 0 & 0 \\ 0 & -Y_{\dot{v}} & 0 & 0 & 0 & 0 \\ 0 & 0 & -Z_{\dot{w}} & 0 & 0 & 0 \\ 0 & 0 & 0 & -K_{\dot{p}} & 0 & 0 \\ 0 & 0 & 0 & 0 & -M_{\dot{q}} & 0 \\ 0 & 0 & 0 & 0 & 0 & -N_{\dot{r}} \end{bmatrix} \tag{A1.5}$$

Table A1.1 - Added Mass Derivative Values

Added Mass Derivative	Value
$X_{\dot{u}}$	-0.3495
$Y_{\dot{v}}$	-3.5554
$Z_{\dot{w}}$	-3.7321
$K_{\dot{p}}$	0
$M_{\dot{q}}$	-0.1123
$N_{\dot{r}}$	-0.1123

Therefore, the complete inertia matrix, **M** consisting of both the rigid body and added mass inertia can be defined by combining Equations (A1.3) and (A1.4) as shown below in Equation (A1.6).

$$\mathbf{M} = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 & 0 & 0 & 0 \\ 0 & m - Y_{\dot{v}} & 0 & 0 & 0 & 0 \\ 0 & 0 & m - Z_{\dot{w}} & 0 & 0 & 0 \\ 0 & 0 & 0 & I_x - K_{\dot{p}} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_y - M_{\dot{q}} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_z - N_{\dot{r}} \end{bmatrix} \quad \text{(A1.6)}$$

A2 - Coriolis and Centripetal Matrix

As with the inertia matrix, the Coriolis and Centripetal matrix, **C** can also be divided into two components consisting of a rigid body component $\mathbf{C}_{RB}$ and an added mass component $\mathbf{C}_{A}$. Again, the matrix representing the rigid body component can be obtained by ascertaining the coefficients of the velocity components of the equations presented in Equation (A1.1). Doing this, produces the matrix shown below in Equation (A1.7).

$$\mathbf{C_{RB}}(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & 0 & 0 & mw & -mv \\ 0 & 0 & 0 & -mw & 0 & mu \\ 0 & 0 & 0 & mv & -mu & 0 \\ 0 & 0 & 0 & 0 & I_z r & -I_y q \\ 0 & 0 & 0 & -I_z r & 0 & I_x p \\ 0 & 0 & 0 & I_y q & -I_x p & 0 \end{bmatrix} \quad \text{(A1.7)}$$

Similarly, as presented in (Fossen,1994), the added mass terms for the Coriolis and centripetal terms can be defined as shown below in Equation (A1.8).

$$\mathbf{C_A}(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & 0 & 0 & -\mathrm{Z}_{\dot{w}}w & \mathrm{Y}_{\dot{v}}v \\ 0 & 0 & 0 & \mathrm{Z}_{\dot{w}}w & 0 & -\mathrm{X}_{\dot{u}}u \\ 0 & 0 & 0 & -\mathrm{Y}_{\dot{v}}v & \mathrm{X}_{\dot{u}}u & 0 \\ 0 & -\mathrm{Z}_{\dot{w}}w & \mathrm{Y}_{\dot{v}}v & 0 & -\mathrm{N}_{\dot{r}}r & M_{\dot{q}}q \\ \mathrm{Z}_{\dot{w}}w & 0 & -\mathrm{X}_{\dot{u}}u & \mathrm{N}_{\dot{r}}r & 0 & -K_{\dot{p}}p \\ -\mathrm{Y}_{\dot{v}}v & \mathrm{X}_{\dot{u}}u & 0 & -M_{\dot{q}}q & K_{\dot{p}}p & 0 \end{bmatrix} \quad \text{(A1.8)}$$

As a result, the complete Coriolis and centripetal matrix, **C** is presented below in Equation (A1.9).

$$\mathbf{C}(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & 0 & 0 & mw - \mathrm{Z}_{\dot{w}}w & \mathrm{Y}_{\dot{v}}v - mv \\ 0 & 0 & 0 & \mathrm{Z}_{\dot{w}}w - mw & 0 & mu - \mathrm{X}_{\dot{u}}u \\ 0 & 0 & 0 & mv - \mathrm{Y}_{\dot{v}}v & \mathrm{X}_{\dot{u}}u - mu & 0 \\ 0 & -\mathrm{Z}_{\dot{w}}w & \mathrm{Y}_{\dot{v}}v & 0 & -\mathrm{N}_{\dot{r}}r & M_{\dot{q}}q \\ \mathrm{Z}_{\dot{w}}w & 0 & -\mathrm{X}_{\dot{u}}u & \mathrm{N}_{\dot{r}}r & 0 & -K_{\dot{p}}p \\ -\mathrm{Y}_{\dot{v}}v & \mathrm{X}_{\dot{u}}u & 0 & I_y q - M_{\dot{q}}q & K_{\dot{p}}p - I_x p & 0 \end{bmatrix} \quad \text{(A1.9)}$$

A3 – Hydrodynamic Damping Matrix

In Equation (A1.1), the vector, **D** is used to represent the hydrodynamic damping properties of the RoboSalmon vehicle. Hydrodynamic damping is the term used in marine vehicle engineering to encapsulate the different forms of drag associated with a vehicle. This hydrodynamic damping is caused by a number of different factors and as a result, the overall hydrodynamic damping matrix, **D** can be written as the sum of multiple components as shown below in Equation (A1.10):

$$\mathbf{D}(\boldsymbol{\nu_r}) = \mathbf{D_P}(\boldsymbol{\nu_r}) + \mathbf{D_S}(\boldsymbol{\nu_r}) + \mathbf{D_W}(\boldsymbol{\nu_r}) + \mathbf{D_M}(\boldsymbol{\nu_r}) \quad \text{(A1.10)}$$

Where $\mathbf{D_P}$ represents the radiation-induced potential damping, $\mathbf{D_S}$ is the linear skin friction, $\mathbf{D_W}$ is the wave drift damping and $\mathbf{D_M}$ is the damping due to vortex shredding. However, of these four components, only two – the linear skin friction and the damping due to vortex shredding - are considered necessary to accurately model the drag forces acting on an underwater vehicle.

To approximate the drag caused by the aforementioned vortex shredding, the standard equation for drag as shown below is used:

$$\mathrm{D} = -\frac{1}{2}\rho \mathrm{C}_D A |V_r(\beta)| V_r(\beta) \quad \text{(A1.11)}$$

Where V_r represents the relative velocity between the vehicle, $\boldsymbol{v}$ and the ocean current, $\boldsymbol{v}_c$ and permits the sideslip angle of the vehicle to be taken into consideration, A is the projected cross sectional area of the vehicle, C_D is the drag coefficient and ρ is the density of water. The drag coefficient is dependent on the shape of the vehicle being modelled. The shape of the RoboSalmon vehicle mirrors that of a prolate ellipsoid and as a result, the drag coefficient is estimated using the following equation (Naddi,2015):

$$C_D = 0.44\left(\frac{b}{a}\right) + 4C_f\left(\frac{a}{b}\right) + 4C_f\left(\frac{a}{b}\right)^{\frac{1}{2}} \tag{A1.12}$$

Where C_f is the friction drag coefficient, a is the radius of the ellipsoid along the x-axis and b is the radius of the ellipsoid along the y-z plane. Consequently, the ratio of these two values will vary depending on which axis the drag coefficient is being evaluated for. Therefore, using the above two equations and the various combinations of the a and b parameters, an estimate for the drag forces acting on the vehicle along the x, y and z axis can be obtained.

However, the methods discussed above don't take into consideration the contributions of either the caudal fin or indeed, the pectoral fins. To take these contributions into consideration, Equation (A1.11) is used again but this time with separate drag coefficients that represent the shape of both the caudal fin and pectoral fins.

Finally, the above explanation of the functionality used to estimate the vehicle's drag has so far assumed that the vehicle is a rigid-body when in actual fact, the tail section of the vehicle is constantly moving in an undulatory manner. As a result of this undulatory motion, the projected cross sectional area of the tail section will vary and as shown in Equation (A1.11), this will result in the drag of the vehicle – particularly that of the tail section also varying. In an attempt to model this undulation of the vehicle's drag, the equation presented in Equation (A1.11) has been altered to that shown below in Equation (A1.13).

$$D_x = -\frac{1}{2}\rho C_D A|V_r(\beta)|V_r(\beta)\cos(90-\theta) \tag{A1.13}$$

Where θ represents the angle between the body-fixed horizontal axis and the lateral displacement of the caudal fin as shown below in Figure A1.

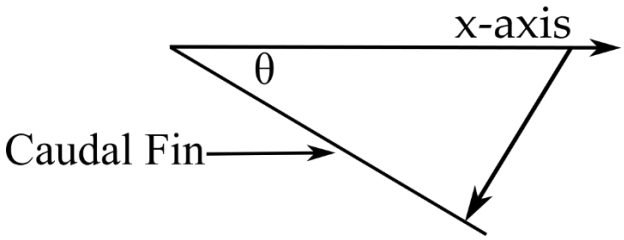


Figure A.1 - Geometry of caudal fin angle.

Therefore, it is apparent from Figure A.1 that when the position of the caudal fin reaches its peak value during a single undulation, the angle, θ is at its maximum value and consequently, the estimate for the drag in the x-direction as estimated using Equation (A1.13) will also reach its peak value as expected. Similarly, to estimate the drag in the y-direction, the same equation presented in Equation (A1.13) is used with the exception that the cosine of the angle is replaced with the sine of the angle,θ as shown below in Equation (A1.14).

$$\mathrm{D_y} = -\frac{1}{2}\rho \mathrm{C}_D(\beta) A |V_r(\beta)| V_r(\beta) \sin(90-\theta) \tag{A1.14}$$

A3 – Restoring Forces & Moments

In marine vehicle engineering, the restoring forces and moments describe the interaction between the gravitational and buoyancy forces acting on the vehicle. As with all physical systems, the gravitational forces acting on the vehicle do so through the vehicle's centre of gravity and similarly, the buoyancy forces act through the vehicle's centre of buoyancy.

The vector of equations used to represent these forces and moments are shown below in Equation (A1.15) and have been taken directly Fossen (2011).

$$\boldsymbol{g}(\eta) = \begin{bmatrix} (W-B)\sin\theta \\ -(W-B)\cos\theta\sin\phi \\ -(W-B)\cos\theta\cos\phi \\ -(y_G W - y_B B)\cos\theta\cos\phi + (z_G W - z_B B)\cos\theta\sin\phi \\ (z_G W - z_B B)\sin\theta + (x_G W - x_B B)\cos\theta\cos\phi \\ -(x_G W - x_B B)\cos\theta\cos\phi - (y_G W - y_B B)\sin\theta \end{bmatrix} \tag{A1.15}$$

Here, W and B are the gravitational and buoyancy forces acting on the vehicle, θ is the pitch angle of vehicle in the Earth-fixed axis, ϕ is the roll angle of the vehicle also in the earth-fixed axis,x_G, y_G and z_G represent the position of the vehicle's centre of gravity along the x, y and z axis and finally, x_B, y_B and z_B represent the position of the centre of buoyancy along the same x, y and z axis.

The equations presented in (A1.15) are the complete set of equations required to model the restoring forces and moments acting on any vehicle operating in the oceanic environment. However, as discussed in (Watts, 2008; Fossen, 2011 & Naddi, 2015), a number of simplifications can be made to the above equations. The first of which is to assume that the RoboSalmon vehicle is neutrally buoyant which means that the gravitational force, W is equal to the buoyancy force, B. The second assumption is to position the centre of buoyancy directly above the centre of gravity and as a result, $x_B = x_G$ and $y_B = y_G$. Applying these assumptions means the equations presented in Equation (A1.15) can be significantly simplified to that which is shown below in Equation (A1.16).

$$\boldsymbol{g}(\eta) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ (z_G W - z_B B)\cos\theta \sin\phi \\ (z_G W - z_B B)\sin\theta \\ 0 \end{bmatrix} \tag{A1.16}$$

Achieving neutral buoyancy through the design of the vehicle is almost impossible. Nevertheless, the assumption of neutral buoyancy is deemed to be acceptable due to the fact that the investigations completed using the RoboSalmon vehicle occurred in the horizontal plane only and as a result, the motion of the vehicle in the vertical plane is not considered paramount to accurately replicating the motion of the physical system.

B: Backpropagation Algorithm

This algorithm operates by using a criterion function to assign an error value to each neuron in the output layer of the network. As the name of the algorithm suggests, these errors are then back-propagated throughout the entire network until every neuron has an error associated with it, which roughly represents its contribution to the overall network error. At this point, the algorithm evaluates the derivative of these errors with respect to the weights of the network and the optimisation technique then uses these derivatives to update each weight within the network in an attempt to minimise the networks overall error.

The particular criterion function used in this work is known as the squared error function and is shown below in Equation (B.1):

$$\mathrm{E}(\mathrm{w},\mathrm{k}) = \frac{1}{2}\sum_{\mathrm{i}=1}^{\mathrm{l}}(\mathrm{y}(\mathrm{i}) - \mathrm{y_m}(\mathrm{i}))^2 \tag{B.1}$$

Where w and k represent the values of the interconnecting weights within the neural network, y represents the desired value from the l^{th} output of the network and $\mathrm{y_m}$ represents the actual output from the network.

As discussed above, the derivative of the above function with respect to the appropriate weights is then used to update the values of the interconnecting weights in an attempt to reduce the overall error of the network. The equations used to represent this process are presented below:

$$k_{li}^{+} = k_{li} - \frac{\partial E_l}{\partial k_{li}} \tag{B.2}$$

$$w_{li}^{+} = w_{li} - \frac{\partial E_i}{\partial w_{ij}} \tag{B.3}$$

Here, k_{li}^{+} and w_{li}^{+} represent the new values for the interconnecting weights, k_{li} and w_{li} that should iteratively reduce the overall network error.

Equations (B.1), (B.2) & (B.3) and the associated discussion have presented a brief overview of the back propagation algorithm used in this work to train the ANN. A full derivation of this method is presented below. Gradient descent is a first-order iterative optimisation algorithm which operates by trying to estimate the local minima of a function. As discussed above for the work completed in this thesis, this function is equal to the criterion function and evaluates the squared error between the desired and actual response of the network and is represented mathematically in Equation (B.1)

To implement the gradient descent technique as part of the backpropagation algorithm, the derivative of Equation (B.1) has to be evaluated with respect to the weights connecting the hidden layer with the output layer, k. To achieve this, the chain rule is adopted to produce the following set of equations:

$$\frac{\partial E}{\partial k_{li}} = \frac{\partial E}{\partial y_m}\frac{\partial y_m}{\partial u_o}\frac{\partial u_o}{\partial k_{li}} \tag{B.4}$$

Where the first term on the right hand side represents the derivative of Equation (B.1) which produces the following equation:

$$\frac{\partial E_l}{\partial y_{m_,}} = -\left(y_l - y_{m_l}\right) \tag{B.5}$$

The above equation represents the derivative of the error associated with the lth output from the network, E_l with respect to the output obtained from network for the same output, $y_{m_,}$.

Furthermore, the second derivative on the right hand-side of Equation (B.4) represents the derivative of the activation function used within the neurons of the output layer and is of the following form:

$$\frac{\partial y_{m_l}}{\partial u_{o_l}} = 1 - tanh^2\left(u_{o_l}\right) \tag{B.6}$$

Finally, the last derivative of Equation (B.4) evaluates the rate of change of the input to each neuron in the output layer with respect to the interconnecting weights represented by the indices l and i. Since the input to each neuron in the output layer can be represented using Equation (B.5) shown above, its derivative with respect to each of the interconnecting weights, k_{li} can be represented simply as the output of the ith neuron in the hidden layer, y_{h_i} as presented below in Equations (B.7) and (B.8).

$$u_{o_l} = \left(\sum_{i=1}^{q} k_{li} f_i \left(\sum_{j=1}^{m} w_{ij} x_j \right) \right) \tag{B.7}$$

$$y_{h_i} = \frac{\partial u_{o_i}}{\partial k_{li}} = \left(f_i \left(\sum_{j=1}^{m} w_{ij} x_j \right) \right) \tag{B.8}$$

Therefore, combining the three derivative equations presented above, the rate of change of the error parameter, E with respect to the interconnecting weight k_{li} can be evaluated. Subsequently, this parameter can be used to evaluate a new value for the interconnecting weights between the hidden output layers using the equation shown below:

$$k_{li}^{+} = k_{li} - \frac{\partial E_l}{\partial k_{li}} \tag{B.9}$$

Where k_{li}^{+} represents the new value for the interconnecting weight, k_{li} which reduces the value for the overall error parameter, E. However, the development of Equations (B.4-B.9) only accounts for the evaluation of the interconnecting weights between the hidden layer and the output layer.

Consequently, as the name of the algorithms suggests, the errors evaluated above are propagated back through the remaining layers of the network to allow the new, improved values for the weights, w_{ij} representing the connections between the input and hidden layers to be evaluated. To evaluate these parameters, first the derivative of the error associated with each output neuron with respect to the output value from each neuron within the hidden layer is evaluated. This can be represented mathematically as shown below in Equation (B.10):

$$\frac{\partial E_{Tot}}{\partial y_{h_i}} = \sum_{p=1}^{l} \frac{\partial E_p}{\partial y_{h_i}} \tag{B.10}$$

Where y_{h_i} represents the output from the i^{th} neuron of the hidden layer, E_p represents the error evaluated for each neuron in the output layer and i represents the particular neuron within the hidden layer for which the above derivative is being evaluated for. Again, using the chain rule, Equation (B.10) can be rewritten into the following form:

$$\frac{\partial E_i}{\partial y_{h_i}} = \sum_{l=1}^{l=200} \frac{\partial E_l}{\partial y_{m_l}} \frac{\partial y_{m_l}}{\partial u_{o_l}} \frac{\partial u_{o_l}}{\partial y_{h_i}} \tag{B.11}$$

Where the first two terms on the right-hand side of the above equation are identical to the first two terms in Equation (B.4). The last term however, represents the derivative of the input to the neurons in the output layer with respect to the outputs of the neurons within the hidden layer. Again, since the input to the output layer of neurons, u_{o_l} is equal to the equation shown above (B.7), its derivative with respect to the outputs from the i^{th} neuron in the hidden layer is equal to the interconnecting weight represented by k_{li}.

Since Equation (B.8) represents the derivative of the total error with respect to the outputs obtained for the hidden neuron, y_{h_i} it is now possible to evaluate the derivative of the total error with respect to the interconnecting weights between the input and hidden layers using the following equation.

$$\frac{\partial E_i}{\partial w_{ij}} = \frac{\partial E_i}{\partial y_{h_i}} \frac{\partial y_{h_i}}{\partial u_{h_i}} \frac{\partial u_{h_i}}{\partial w_{ij}} \tag{B.12}$$

Where again, because the activation function used within the neurons of the hidden layer are identical to those used in the output layer, the second term of Equation (B.12) is identical to Equation (B.6) with the exception of u_o being replaced with u_h. Finally, since the input to each of the neurons in the hidden layer, represented by u_{h_i} in the above equation, is equal to the formula shown below in Equation (B.13), its derivative with respect to the associated connecting weights, w_{ij} is simply equal to the input parameter identified by x_j.

$$u_{h_i} = \left(\left(\sum_{j=1}^{m} w_{ij} x_j \right) \right) \tag{B.13}$$

Finally, the derivative obtained from Equation (B1.10) is used to evaluate new values for the weights connecting the input layer with the hidden layer using the equation shown below in Equation (B.14).

$$w_{li}^{+} = w_{li} - \frac{\partial E_i}{\partial w_{ij}} \tag{B.14}$$

The above derivation represents the steps taken to evaluate new values for the interconnecting weights of the neural network that will reduce the error calculated by the criterion function.

Bibliography

Adkins, D. & Yan, Y.Y., 2006. CFD Simulation of Fish-like Body Moving in Viscous Liquid. *Journal of Bionic Engineering*, 3, pp.147–153.

Akyildiz, I.F., Pompili, D. & Melodia, T., 2004. Challenges for efficient communication in underwater acoustic sensor networks. *ACM SIGBED Review*, 1, pp.3–8.

Akyildiz, I.F., Pompili, D. & Melodia, T., 2006. State-of-the-Art in Protocol Research for Underwater Acoustic Sensor Networks. In *1st ACM International Workshiop on Underwater Networks*. ACM, pp. 7–16. Available at: https://dl.acm.org/citation.cfm?id=1161043.

Alt, C. Von, 2003. Autonomous underwater vehicles. In *Autonomous Underwater Lagrangian Platforms and …*. pp. 1–5. Available at: http://auvac.org/uploads/publication_pdf/AUVs WHOI 2003.pdf.

Anderson, B.D.., Baris Fidan, C.. & Hendrickx, J.., 2008. Rigid Graph Control Autonomous Formations. *IEEE Control Systems Magazine*, (December), pp.48–63.

Anderson, J.M. & Chhabra, N.K., 2002. Maneuvering and stability performance of a robotic tuna. *Integrative and Comparative Biology*, 42(1), pp.118–126.

Andrews, J.G., Weber, S. & Haenggi, M., 2007. Ad Hoc Networks: To Spread or Not to Spread? [Ad Hoc and Sensor Networks]. *IEEE Communications Magazine*, 45(12), pp.84–91.

Aoki, I., 1981. A Simulation Study on the Schooling Mechanism in Fish. *Bulletin of Japanese Society of Scientific Fisheries*, 48(8), pp.1081–1088.

Aoki, I.., 1982. Simulation Study on the Schooling Mechanisms in Fish. *Bulletin of Japanese Society of Scientific Fisheries*, 48(1), pp.1081–1088.

Arnold, M., 2007. Multi-Rate Time Integration for Large Scale Multibody System Models. *IUTAM Symposium on Multiscale Problems in Multibody System*, pp.1–10. Available at: http://link.springer.com/10.1007/978-1-4020-5981-0_1.

Ayaz, M., 2012. Reliable data deliveries using packet optimization in multi-hop underwater sensor networks. *Journal of King Saud University - Computer and Information Sciences*, 24(1), pp.41–48. Available at: http://dx.doi.org/10.1016/j.jksuci.2011.09.001.

Bai, E.W. & Giri, F., 2010. Introduction to Block-Oriented Nonlinear Systems. In *Lecture Notes in Control and Information Sciences*. pp. 3–11.

Balch, T. & Arkin, R.C., 1998. Behavior-based formation control for multirobot teams. *IEEE Transactions on Robotics and Automation*, 14(6), pp.926–939.

Bar-Cohen, Y., 2011. Biomimicking Marine Mechanisms and Organizational Principles.

Marine Technology Society Journal, 45(4), pp.14–15.

Barrett, D., 1996. *Propulsive Efficiency of a Flexible Hull Underwater Vehicle*. Massachusetts Institute of Technology. Available at: https://dspace.mit.edu/handle/1721.1/10559.

Bellingham, J.G. & Rajan, K., 2007. Robotics in Remote and Hostile Environments. *Science*, 318(5853), pp.1098–1102. Available at: http://www.sciencemag.org/cgi/doi/10.1126/science.1146230.

Berry, M.J. & Linoff, G., 1997. *Data Mining Techniques*, New York: Wiley & Sons.

Bennet, D.J. & McInnes, C.R.,Verifiable control of a swarm of unmanned aerial vehicles. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*. 223(7), 939–953.

Bennet,D.J. & McInnes,C.R., 2009. Distributed control of multi-robot systems using bifurcating potential fields. *Journal of Robotics and Autonomous Systems.* 58(3), pp.256-264.

Bennet,D.J, 2010. Pattern Formation in Swarming Systems using Bifurcating Potential Fields. University of Strathclyde. Available at: http://digitool.lib.strath.ac.uk/webclient/StreamGate?folder_id=0&dvs=1550910517684~157&usePid1=true&usePid2=true

Bibuli, M., Caccia, M. & Lapierre, L., 2011. Development of a Two-Joint Robotic Fish for Real-World Exploration. *Journal of Field Robotics*, 28(1), pp.70–79.

Bishop, C., 1995. Statistical Pattern Recognition. In *Neural Networks for Pattern Recognition*. New York: Oxford University Press Inc., New York, pp. 1–28.

Blidberg, D.R., 2001. The developement of Autonomous Underwater Vehicles (AUV): a brief summary. In *IEEE Conference on Robotics and Automation (ICRA)*. Seoul, South Korea: IEEE.

Blum, A., 1992. *Neural Networks in C++*, New York: Wiley.

Bohlin, T., 2006. Prospects and Problems. In *Practucak Grey-box Process Identification*. London: Springer-Verlag. Available at: https://link.springer.com/book/10.1007%2F1-84628-403-1.

Boston Engineering, 2014. BIOSwimmer: Changing the Underwater Inspection Game. Available at: http://auvac.org/uploads/platform_pdf/BIOSwimmer.pdf.

Braginsky, B. & Guterman, H., 2016. Obstacle Avoidance Approaches for Autonomous Underwater Vehicle : Simulation and Experimental Results. , 41(4), pp.882–892.

Brignone, L., Alves, J. & Opderbecke, J., 2009. GREX sea trials: First experiences in multiple underwater vehicle coordination based on acoustic communication. In *OCEANS 2009 - Europe*. Bremen.

Brooks, R.J. & Tobias, A.M., 1996. Choosing the best model: Level of detail, complexity, and model performance. *Mathematical and Computer Modelling*, 24(4), pp.1–14.

Bryne, T.H., 2017. *Nonlinear Observer Design for Aided Inertial Navigation of Ships*. Norwegian University of Science and Technology. Available at:

http://www.fossen.biz/home/PhD/thesis/Torleiv Bryne 2017.pdf.

Burlutskiy, N., Touahmi, Y. & Lee, B.H., 2012. Power efficient formation configuration for centralized leader-follower AUVs control. *Journal of Marine Science and Technology (Japan)*, 17(3), pp.315–329.

Burrowes, G.E., Brown, J. & Khan, J.Y., 2007. Adaptive Space Time - Time Division Multiple Access (AST-TDMA) Protocol for an Underwater Swarm of AUV ' s. In *OCEANS Conference*. Bergen: IEEE, pp. 41–48.

Button, R.W. Kamp, J., Curtin, T.. & Drydon, J., 2009. *A Survey of Missions for Unmanned Undersea Vehicles*, Available at: http://www.rand.org/pubs/research_briefs/RB9539.html.

Cai, G., Chen, B.M. & Lee, T.H., 2011. Coordinate Systems and Transformations. In *Unmanned Rotorcraft Systems*. Springer, pp. 23–34. Available at: http://www.springer.com/engineering/control/book/978-0-85729-634-4%5Cnhttp://link.springer.com/10.1007/978-0-85729-635-1.

Caharija, W. et al., 2016. Integral Line-of-Sight Guidance and Control of Underactuated Marine Vehicles: Theory, Simulations, and Experiments. In *IEEE Transactions on Control Systems Technology*, 24(5),pp 1623-1642.

Caiti, A., Calabrò, V. & Munafò, A., 2012. AUV team cooperation: Emerging behaviours and networking modalities. In *IFAC Proceedings Volumes on Manoeuvring and Control of Marine Craft*. Arenzano, Italy, pp. 342–347.

Cella, U.M., Johnstone, R. & Shuley, N., 2009. Electromagnetic Wave Wireless Communication in Shallow Water Coastal Environment : Theoretical Analysis and Experimental Results. In ACM, ed. *In. Proc. 4th ACM Int Workshop ib Underwater Networks (WUWNet)*. Berkely, CA, pp. 1–9.

Centre, N.O., 2017. Sampling and monitoring the ocean using AUVs. *NOC Website*. Available at: http://www.noc.ac.uk/news/sampling-monitoring-ocean-using-auvs-nelson-mandela-university [Accessed October 8, 2018].

Chakravarthy, A. & Ghose, D., 1998. Obstacle Avoidance in a Dynamic Environment : A Collision Cone Approach. *IEEE Transactions on Systems, Man and Cybernetics—Part A: Systems and Humans,*, 28(5), pp.562–574.

Chance, T.S., Kleiner, A.A., Northcutt, J.G., 2000. The Autonomous Underwater Vehicle (AUV): A Cost-Effective Alternative to Deep-Towed Technology. *Integrated Coastal Zone Management Magazine*, pp.65–69. Available at: http://www.cctechnol.com/Files/Articles/TheAutonomousUnderwaterVehicle.pdf.

Chao, H., Cao, Y. & Chen, Y., 2010. Autopilots for Small Unmanned Aerial Vehicles : A Survey. *International Journal of Control, Automation and Systems*, 8, pp.36–44.

Chen, B., Pandey, P. & Pompili, D., 2012. An adaptive sampling solution using autonomous underwater vehicles. In *ACM International Conference on Underwater Networks Systems (WUWNet)*. Available at: http://www.cac.rutgers.edu/sites/all/files/Biblio Papers/2012_MCMC_AdaptSampl.pdf.

Chen, B., Pompili, D. & Parashar, M., 2010. Bio-inspired communications for coordination among autonomous underwater vehicles. In *33rd IEEE Sarnoff Symposium 2010, Conference Proceedings*. Princeton, NJ, pp. 1–60.

Chen, K. et al., 2014. A Survey on MAC Protocols for Underwater Wireless Sensor Networks. *IEEE Communications Surveys & Tutorials*, 16(3), pp.1433–1447. Available at: http://ieeexplore.ieee.org/abstract/document/6757189/.

Chen, Y. et al., 2005. Formation Control : A Review and A New Consideration. In *International Conference on Intelligent Robots and Systems (IROS)*. Edmonton, Canada.

Cho, J.L. (Jamie L., 1998. *Electronic Subsystems of a Free-Swimming Robotic Fish*. Available at: http://dspace.mit.edu/handle/1721.1/46192.

Clapham, R.J., 2015. *Developing High Performance Linear Carangiform Swimming*. University of Essex.

Clapham, R.J. & Hu, H., 2014a. iSplash-I: High Performance Swimming Motion of a Carangiform Robotic Fish with Full-Body Coordination. In *IEEE International Conference on Robotics and Automation (ICRA)*. Hong Kong, China: IEEE, pp. 322–327. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6906629.

Clapham, R.J. & Hu, H., 2014b. iSplash-II: Realizing Fast Carangiform Swimming to Outperform a Real Fish. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*. Chicago,IL: IEEE, pp. 1080–1086. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6942692.

Claus, B. & Bachmayer, R., 2012. Hybrid glider : motivation , design and evaluation. In *Further Advances in Unmanned Marine Vehicles*. London, UK: IET, pp. 283–306.

Clough, B., 2002. Metrics, schmetrics! How the heck do you determine a UAV's autonomy anyway? In *Performance Metrics for Intelligent Systems (PerMIS) Conference*. Gaithersburg, MD., pp. 313–319. Available at: http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA515926.

Conry, M. et al., 2013. BIOSwimmer: Enabling technology for port security. *2013 IEEE International Conference on Technologies for Homeland Security, HST 2013*, pp.364–368.

Copley, J., 2014. Just How Little Do We Know about the Ocean Floor. Available at: http://www.scientificamerican.com/article/just-how-little-do-we-know-about-the-ocean-floor/.

Cruz, D. et al., 2007. Decentralized cooperative control: A multivehicle platform for research in networked embedded systems. *IEEE Control Systems Magazine*, 27(June), pp.58–78.

Cruz, N.A. & Matos, A.C., 2010. Adaptive sampling of thermoclines with Autonomous Underwater Vehicles. *Oceans*, pp.1–6. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5663903.

Cruz, N.A. & Matos, A.C., 2010. Reactive AUV motion for thermocline tracking.

OCEANS'10 IEEE Sydney, OCEANSSYD 2010, pp.1–6. Available at: http://ieeexplore.ieee.org/document/5603883/.

Cui, R. et al., 2010. Leader-follower formation control of underactuated autonomous underwater vehicles. *Ocean Engineering*, 37(17–18), pp.1491–1502. Available at: http://dx.doi.org/10.1016/j.oceaneng.2010.07.006.

Cui, R. et al., 2009. Leader-follower formation control of underactuated AUVS with leader position measurement. In *Proceedings - IEEE International Conference on Robotics and Automation*. pp. 979–984.

Le Cun, Y. et al., 1998. Efficient BackProp.pdf. In *Neural Networks: Tricks of the Trade*. pp. 9–50. Available at: http://yann.lecun.com/exdb/publis/pdf/lecun-98b.pdf.

Cybenko, G., 1989. Approximation by Superpositions of a Sigmoidal Function. *Mathematical Control Signal Systems*, 2(3), pp.303–314. Available at: https://link.springer.com/article/10.1007/BF02551274.

Das, B., Subudhi, B. & Pati, B.B., 2016. Cooperative formation control of autonomous underwater vehicles: An overview. *International Journal of Automation and Computing*, 13(3), pp.199–225.

Davis, R., Eriksen, C. & Jones, C., 2002. Autonomous buoyancy-driven underwater gliders. In *The Technology and Applications of Autonomous Underwater Vehicles*. pp. 37–49. Available at: http://www.ifremer.fr/lpo/gliders/donnees_tt/references/techno/4Gliders.pdf.

Diamant, R. & Lampe, L., 2011. Spatial reuse time-division multiple access for broadcast Ad hoc underwater acoustic communication networks. *IEEE Journal of Oceanic Engineering*, 36(2), pp.172–185.

Douglas-Westwood, 2016. DouglasWestwoodReference.pdf. Available at: http://www.douglas-westwood.com/report/oil-and-gas/world-auv-market-forecast-2014-2018/ [Accessed April 20, 2016].

Drapher, C.S. et al., 1965. *Apollo Guidance and Navigation*, Available at: http://www.ibiblio.org/apollo/hrst/archive/1713.pdf.

Drones, N., 2010. Special Purpose Underwater Research Vehicle.pdf. *Naval Drones*. Available at: http://www.navaldrones.com/SPURV.html [Accessed January 11, 2017].

ECA Group, 2018. A-18-S. *Vehicle Specification*, pp.1–2. Available at: http://eca-media.ecagroup.com/player/pdf?key=51e7d99ae7658defb85b0b0a9a52236d [Accessed May 31, 2018].

Edwards, D.B. et al., 2004. A leader-follower algorithm for multiple AUV formations. In *2004 IEEE/OES Autonomous Underwater Vehicles (IEEE Cat. No.04CH37578)*. pp. 40–46.

Ellerby, D.J., 2010. How efficient is a fish? *The Journal of experimental biology*, 213(22), pp.3765–3767.

Eriksen, C.C. et al., 2001. Seaglider: A long-range autonomous underwater vehicle for oceanographic research. *IEEE Journal of Oceanic Engineering*, 26(4), pp.424–436.

Etkin, B. & Reid, L.D., 1996. General Equations of Phase Equilibrium. In *Dynamics of Flight - Stability and Control*. John Wiley & Sons, Ltd, pp. 93–127.

Evo Logics, 2014. S2CR 48/78 Product Information. Available at: http://www.evologics.de/files/DataSheets/EvoLogics_S2CR_4878_Product_Information.pdf.

Faltinsen, O.M., 2006. *Maneuvring Hydrodynamics of High-Speed Marine Vehicles.pdf*, Cambridge: Cambridge University Press.

Farr, N. et al., 2010. An integrated, underwater optical / acoustic communications system. In *OCEANS 2010 IEEE - Sydney*. Sydney: IEEE. Available at: http://ieeexplore.ieee.org/document/5603510/.

Farrell, J.A. et al., 2005. Chemical Plume Tracing via an Autonomous Underwater Vehicle. *IEEE Journal of Oceanic Engineering*, 30(2), pp.428–442. Available at: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1522521.

Farrell, J. a. et al., 2003. Chemical plume tracing experimental results with a REMUS AUV. *Oceans 2003. Celebrating the Past ... Teaming Toward the Future (IEEE Cat. No.03CH37492)*, 2, pp.962–968.

Ferguson, J. et al., 1999. Theseus AUV-two record breaking missions. *Sea Technology*, 40(2), pp.65–70. Available at: https://www.researchgate.net/publication/297546152_Theseus_AUV_-_Two_record_breaking_missions_-_Designed_to_lay_fiber-optic_cable_from_a_site_near_the_shore_of_Ellesmere_Island_to_a_scientific_acoustic_array_in_the_Arctic_Ocean.

Finn, A. & Scheding, S., 2010. Developments and Challenges for Autonomous Unmanned Vehicles. In *Intelligent Systems Reference Library*. pp. 5–25.

Fiorelli, E. et al., 2006. Multi-AUV control and adaptive sampling in Monterey Bay. *IEEE Journal of Oceanic Engineering*, 31(4), pp.935–948.

Fossen, T.I., 1994. Guidance and control of ocean vehicles.

Fossen, T.I., 2011. *Marine Craft Hydrodynamics and Motion Control*, John Wiley & Sons, Ltd.

Franklin, G., Powell, J., Emama-Naeini, A., 1991. *Feedback Control of Dynamic Systems.Pdf* 2nd Edtion. A.-W. P. Company, ed.,

Furlong, M.., Stevenson, P. & Mcphail, S.D., 2007. A Concept Design for an Ultra-Long Range Survey Class AUV. In *OCEANS 2007 - Europe*. IEEE, pp. 1–6. Available at: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4302453.

Furlong, M.E. et al., 2012. Autosub Long Range: A long range deep diving AUV for ocean monitoring. In *2012 IEEE/OES Autonomous Underwater Vehicles, AUV 2012*. IEEE, pp. 1–7. Available at: http://ieeexplore.ieee.org/document/6380737/.

Gao, T. et al., 2016. Mechanics Hull shape optimization for autonomous underwater vehicles using CFD. *Engineerin Applications of Computational Fluid Mechanics*, 10(1), pp.599–607. Available at: https://doi.org/10.1080/19942060.2016.1224735.

Gear, C.W. & Wells, D.R., 1984. Multirate linear multistep methods. *Bit*, 24(4), pp.484–502.

Generaly Dynamics, 2018. Bluefin-21 AUV. *Vehicle Specification*. Available at: https://gdmissionsystems.com/products/underwater-vehicles/bluefin-21-autonomous-underwater-vehicle [Accessed May 31, 2018].

German, C.R. et al., 2012. A long term vision for long-range ship-free deep ocean operations: Persistent presence through coordination of Autonomous Surface Vehicles and Autonomous Underwater Vehicles. *2012 IEEE/OES Autonomous Underwater Vehicles, AUV 2012*.

Ghabcheloo, R. et al., 2006. Coordinated path-following control of multiple AUVs in the Presence of Communication failures and time delays. In *IEEE International Conference on Decision Control*. San Diego, Cal, pp. 4346–4350.

Griffiths, G. McPhail, S.., 2011. AUVs for Depth and Distance: Autosub 6000 and Autosub Long Range. *Sea Technology*, pp.27–90. Available at: http://www.sea-technology.com/features/2011/1211/depth_distance.php.

Griffiths, G. et al., 2004. Energy storage for long endurance AUVs. In *ATUV Conf.,iMarEST*. Available at: http://eprints.soton.ac.uk/15783/.

Groves, P.D., 2013. Inertial Sensors. In Artech House, ed. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. pp. 137–161. Available at: https://ebookcentral.proquest.com/lib/gla/detail.action?docID=1531533.

Gurney, K., 1997. Neural Network - an overview. In *An Introduction to Neural Networks*. Taylor & Francis Group, pp. 1–6.

Hagen, P.E., Midtgaard, Ø. & Hasvold, Ø., 2007. Making AUVs truly autonomous. In *Oceans 2007*. IEEE. Available at: http://ieeexplore.ieee.org/document/4449405/#full-text-section.

Hauth, J., 2008. *Grey-Box Modelling for Nonlinear Systems*. Technische Universitate Kaiserlautern. Available at: https://kluedo.ub.uni-kl.de/frontdoor/deliver/index/docId/2045/file/diss.pdf.

Healey, A.J Marco, D.., 1992. Slow Speed Flight Control of Autonomous Underwater Vehicles Experimental Results with Nps AUV II.pdf. In *International Society of Offshore and Polar Engineers*. Available at: https://www.onepetro.org/conference-paper/ISOPE-I-92-172.

Healey, A.J. & Lienard, D., 1993. Multivariable SlidingKMode Control for Autonomous Diving and Steering of Unmanned Underwater Vehicles. , 18(3).

Heaton, J., 2009. *Programming Neural Networks with Encog3 in Java*,

Hegrenaes, O. & Hallingstad, O., 2011. Model-Aided INS With Sea Current Estimation for. , 36(2), pp.316–337.

Heidemann, J., Stojanovic, M. & Zorzi, M., 2012. Underwater sensor networks: applications, advances and challenges. *Philosophical Transations of the Royal Society A*, (370), pp.158–175.

Herkewitz, W., 2013. Ocean Drones Plumb New Depths. Available at: https://www.nytimes.com/2013/11/12/science/earth/ocean-drones-plumb-new-depths.html.

Hirata, K. & Kawai, S., 2001. Prototype Fish Robot, UPB-2001. *Prototype Fish Robot, UPF-2001*. Available at: http://www.nmri.go.jp/eng/khirata/fish/experiment/upf2001/index_e.html [Accessed August 30, 2016].

Hirata, K., Takimoto, T. & Tamura, K., 2000. Study on turning performance of a fish robot. *First International Symposium on Aqua Bio-Mechanisms*, pp.287–292. Available at: http://www.nmri.go.jp/eng/khirata/list/fish/isamec2000.pdf.

Horváth, G. & Sergey, A., 2003. Neural Networks in System Identification. In *Neural Networks for Instrumentation, Measurement and Related Industrial Applications*. pp. 43–78.

Hu, H., 2006. Biologically inspired design of autonomous robotic fish at Essex. In *IEEE SMC UK-RI Chapter Conference on Advances in Cybernetic Systems*. Sheffield,UK, pp. 1–8.

Huang, H.-M., Messina, E. & Albus, J., 2005. Autonomy Levels for Unmanned Systems (ALEFUS) Framework: An Update. In *SPIE Defense and Security Symposium*. Orlando, FL. Available at: http://ws680.nist.gov/publication/get_pdf.cfm?pub_id=822672.

Huth, A. & Wissel, C., 1991. The Simulation of the Movement of Fish Schools. *Journal Theoretical Biology*, 156(1), pp.365–385.

Jadbabaie, A., Lin, J. & Morse, A.S., 2003. Coordination of Groups of Mobile Autonomous Agents Using Nearest Neighbor Rules. , 48(6), pp.988–1001.

Jakuba, M. V et al., 2008. Long-Baseline Acoustic Navigation for Under-Ice Autonomous Underwater. *Journal of Field Robotics*, 25, pp.861–879.

Jia, Q. & Li, G., 2007. Formation Control and Obstacle Avoidance Algorithm of Multiple Autonomous Underwater Vehicles (AUVs) Based on Potential Function and Behavior Rules. In *Proceedings of the IEEE International Conference on Automation and Logistics*. Jinan, China, pp. 569–573. Available at: https://ieeexplore.ieee.org/document/4338629/.

Jin, G.J.G. et al., 2001. Modeling MR-dampers: a nonlinear blackbox approach. *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)*, 1, pp.1–5.

Kalwa, J., 2010. Final results of the European project GREX: Coordination and control of cooperating marine robots. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 7(PART 1), pp.181–186.

Kang, X.K.X., Xu, H.X.H. & Feng, X.F.X., 2009. Fuzzy logic based behavior fusion for multi-AUV formation keeping in uncertain ocean environment. In *OCEANS 2009*. Biloxi, MA,: IEEE, pp. 1–7. Available at: http://ieeexplore.ieee.org/document/5422361/.

Kat, C.P. & Els, P.S., 2012. Validation Metric based on Relative Error. *Mathematical and*

Computer Modelling of Dynamical Systems: Methods, Tools and Applications in Engineering and Related Sciences, 18(5), pp.487–520.

Kelasidi,E. et al., 2017. Integral Line-of-Sight Guidance for Path Following Control of Underwater Snake Robots: Theory and Experiments. In I*EEE Transactions on robotics, 33 (3), pp. 610-628.*

Khatib,O., 1986. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotic Research*, 5(1), pp. 90-98.

Kilfoyle, D.B. & Baggeroer, a B., 2000. The state of the art in underwater acoustic telemetry. *Oceanic Engineering, IEEE Journal of*, 25(1), pp.4–27. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=820733.

Kim,D.H., Wang,H., Shin,S., 2006. Decentralised control of autonomous swarm systems using artifical potential functions: Analytical design guidelines. *Journal of Intelligent and Robotic Systems.,* 45(4) pp. 369:394.

Krogh, A., 1995. Neural Network Ensembles , Cross Validation , and Active Learning. In *Advances in Neural Information Processing Systems*. MIT Press, pp. 231–238.

Kumph, J.M., 2000. *Maneuvering of a Robotic Pike*. Massachusetts Institute of Technology. Available at: http://hdl.handle.net/1721.1/8968.

Lawton, J.R.T., Beard, R.W. & Young, B.J., 2003. A decentralized approach to formation maneuvers. In *IEEE Transactions on Robotics and Automation*. pp. 933–941.

Leonard, J.J. & Bahr, A., 2016. Autonomous Underwater Vehicle Navigation. In *Springer Handbook of Ocean Engineering*. Springer, pp. 341–357. Available at: https://link.springer.com/book/10.1007/978-3-319-16649-0.

Leonard, N.E. et al., 2007. Collective Motion, Sensor Networks, and Ocean Sampling Collective Motion, Sensor Networks, and Ocean Sampling. *Proceedings of the IEEE*, 95(1), pp.48–74.

Leonard, N.E. et al., 2010. Coordinated Control of an Underwater Glider Fleet in an Adaptive Ocean Sampling Field Experiment in Monterey Bay. *Journal of Field Robotics*, 27(6), pp.718–740.

Lewis, M.A. & Tan, K.-H., 1997. High Precision Formation Control of Mobile Robots Using Virtual Structures. *Auton. Robots*, 4(4), pp.387–403. Available at: http://dblp.uni-trier.de/db/journals/arobots/arobots4.html#LewisT97.

Li, H., Shi, Y. & Yan, W., 2016. Receding Horizon Formation Control of Nonholonomic Autonomous Underwater Vehicles. In *Proceedings of the 35th Chinese Control Conference*. Chengdu, China, pp. 6062–6067. Available at: https://ieeexplore.ieee.org/document/7554308/.

Li, N.H.M. & Liu, H.H.T., 2008. Multiple UAVs formation flight experiments using Virtual Structure and Motion Synchronization. In *American Control Conference*. Seattle, WA, pp. 1–19.

Lighthill, M.J., 1971. Large-Amplitude Elongated-Body Theory of Fish Locomotion. In *Proceedings of the Royal Society B:Biological Sciences*. pp. 125–138.

Liu, J., Dukes, I. & Hu, H., 2005. Novel mechatronics design for a robotic fish. *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pp.2077–2082.

Liu, J. & Hu, H., 2010. Biological Inspiration: From Carangiform Fish to Multi-Joint Robotic Fish. *Journal of Bionic Engineering*, 7(1), pp.35–48. Available at: http://dx.doi.org/10.1016/S1672-6529(09)60184-0.

Ljung, L., 2001. Black-box models from input-output measurements. In IEEE, ed. *IEEE Instrumentation and Measurement Technology Conference*. Budapest: IEEE.

Ljung, L., 1999. *System Identification: Theory for the users, 2nd ed.*, Prentice-Hall.

Lovelock, J.E., 1990. Hands Up for the Gaia hypothesis. *Nature*, 344.

Maritime, K., 2017a. Autonomous Underwater Vehicle: The HUGIN Family. , p.11. Available at: https://www.km.kongsberg.com/ks/web/nokbg0397.nsf/AllWeb/A6A2CC361D3B9653C1256D71003E97D5/$file/HUGIN_Family_brochure_r2_lr.pdf?OpenElement.

Maritime, K., 2017b. Autonomous Underwater Vehicle - REMUS 6000. Available at: https://www.km.kongsberg.com/ks/web/nokbg0240.nsf/AllWeb/481519DA1B0207CDC12574B0002A8451?OpenDocument [Accessed January 23, 2017].

Maritime, K., 2018. AUV Systems. *Company Webpage*, pp.1–16. Available at: https://www.km.kongsberg.com/ks/web/nokbg0397.nsf/AllWeb/981F09C091062D2AC1257C94002AE05A/$file/commercial-auv-brochure.pdf?OpenElement [Accessed May 21, 2018].

Marshall, T., 2016. Autosubs are Go! *Planet Earth*, pp.12–15. Available at: http://www.nerc.ac.uk/latest/publications/planetearth/archive/planet-earth-sum16/.

Masters, T., 1993. *Practical Neural Network Recipes in C++*, Morgan Kaufmann.

Mathworks, 2006a. Matlab atan2 Function.pdf. Available at: https://uk.mathworks.com/help/matlab/ref/atan2.html [Accessed December 14, 2017].

Mathworks, 2006b. Standard Deviation. Available at: https://uk.mathworks.com/help/matlab/ref/std.html [Accessed May 2, 2018].

Matsumaru, R., Nagami, K. & Takeya, K., 2012. Reconstruction of the Aceh Region following the 2004 Indian Ocean tsunami disaster: A transportation perspective. *IATSS Research*, 36(1), pp.11–19. Available at: http://dx.doi.org/10.1016/j.iatssr.2012.07.001.

Mazlan, A.N.A., 2015. *A Fully Actuated Tail Propulsion System for a Biomimetic Autonomous Underwater Vehicle*. University of Glasgow. Available at: http://theses.gla.ac.uk/id/eprint/6343.

Mazlan, A.N.A. & McGookin, E., 2012. Modelling and Control of a Biomimetic Autonomous Underwater Vehicle. In *International Conference on Control, Automation, Robotics & Visions*. Guangzhou, China, pp. 88–93.

McColgan, J. & McGookin, E., 2017. Effect of Communication Delays on the Successful

Coordination of a Group of Biomimetic AUVs. In *OCeans 2017 - Europe*. Aberdeen: IEEE.

McGookin, E. & Watts, C., 2012. A biomimetic underwater vehicle design concept. In *Further Advances in Unmanned Marine Vehicles*. London, UK, pp. 331–357. Available at: http://eprints.gla.ac.uk/68370/.

McGookin, E.W., 1997. *Optimisation of Sliding Mode Controllers for Marine Applications : A study of methods and implementation issues*. University of Glasgow. Available at: http://theses.gla.ac.uk/3980/.

Millan, P. et al., 2014. Formation control of autonomous underwater vehicles subject to communication delays. *IEEE Transactions on Control Systems Technology*, 22(2), pp.770–777.

Mintchev, S. et al., 2014. Mechatronic design of a miniature underwater robot for swarm operations. *Proceedings - IEEE International Conference on Robotics and Automation*, pp.2938–2943.

Moline, M.A., Woodruff, D.L. & Evans, N.R., 2007. Optical Delineation of Benthic Habitat Using an Autonomous Underwater Vehicle. In *Journal of Field Robotics*. pp. 461–471.

Moline, M.A., Woodruff, D.L. & Evans, N.R., 2007. Optical Delineation of Benthic Habitat Using an Autonomous Underwater Vehicles. *Journal of Field Robotics*, 24(6), pp.461–471. Available at: http://onlinelibrary.wiley.com/doi/10.1002/rob.20176/epdf.

Morgansen, K.A., Triplett, B.I. & Klein, D.J., 2007. Geometric methods for modeling and control of free-swimming fin-actuated underwater vehicles. *IEEE Transactions on Robotics*, 23(6), pp.1184–1199.

Muqattash, A., Krunz, M. & Ryan, W.E., 2003. Solving the near – far problem in CDMA-based ad hoc networks q. *Ad Hoc Networks*, 1, pp.435–453.

Murray-Smith, D., 2012a. Appendix A1 : models of an unmanned underwater vehicle (UUV) the basic nonlinear model of the vehicle. In *Modelling and Simulation of Integrated Systems in Engineering: Issues of Methodology, Quality, Testing and Application*. pp. 321–331.

Murray-Smith, D., 2012b. The Principles of System Modelling. In *Modelling and Simulation of Integrated Systems in Engineering: Issues of Methodology, Quality, Testing and Application*. Woodhead Publishing Ltd, pp. 1–18.

Murray, R.M., 2007. Recent Research in Cooperative Control of Multivehicle Systems. *Journal of Dynamic Systems, Measurement, and Control*, 129(September 2007), p.571.

NASA, 1994. Magellan Mission to Venus. *NASA Facts*, pp.1–4. Available at: https://www.jpl.nasa.gov/news/fact_sheets/mgn.pdf [Accessed May 18, 2018].

Nicholson, J.W. Healey, A.., 2008. The Present State of Autonomous Underwater Vehicles (AUVs) Application and Technologies. *Marine Technology Society Journal*, 45(1), pp.44–51.

Nielsen, H.A. & Madsen, H., 2006. Modelling the heat consumption in district heating systems using a grey-box approach. *Energy and Buildings*, 38(1), pp.63–71.

Niku, S.B., 2001. Introduction to Robotics Analysis, Systems, Applications.

Norgaard, M., 1997. *Neural Network Based System Identification Toolbox*, The Matworks Company. Available at: http://www.iau.dtu.dk/research/control/nnlib/manual1.pdf.

Oh, K.K., Park, M.C. & Ahn, H.S., 2015. A survey of multi-agent formation control. *Automatica*, 53, pp.424–440. Available at: http://dx.doi.org/10.1016/j.automatica.2014.10.022.

Olfati-Saber, R., Fax, J.A. & Murray, R.M., 2007. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1), pp.215–233.

Orgen,P., Fiorelli,E., Leonardo,N.E., 2004. Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment. *IEEE Transations on Automatic Control.* 49(8), pp. 1292-1302.

Paley, D. a., Zhang, F. & Leonard, N.E., 2008. Cooperative control for ocean sampling: The glider coordinated control system. *IEEE Transactions on Control Systems Technology*, 16(4), pp.735–744.

Partan, J., Kurose, J. & Neil, B., 2007. A Survey of Practical Issues in Underwater Networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 11(4), pp.23–33.

Partridge, B.L., Pitcher, T.J. & Gables, C., 1980. of Comparative The Sensory Basis of Fish Schools : Relative Roles of Lateral Line and Vision. *Structure*, 325(1980), pp.315–325.

Paull, L. et al., 2014. AUV navigation and localization: A review. *IEEE Journal of Oceanic Engineering*, 39(1), pp.131–149.

Petillo, S., 2015. *Autonomous & adaptive oceanographic front tracking on board autonomous underwater vehicles*.

Petillo, S. & Schmidt, H., 2014. Exploiting adaptive and collaborative AUV autonomy for detection and characterization of internal waves. *IEEE Journal of Oceanic Engineering*, 39(1), pp.150–164.

Petillo, S., Schmidt, H. & Balasuriya, A., 2012. Constructing a distributed AUV network for underwater plume-tracking operations. *International Journal of Distributed Sensor Networks*, 2012.

Qian, N., 1999. On the momentum term in gradient descent learning algorithms. *Neural Networks*, 12(1), pp.145–151.

Raby, A. et al., 2015. Implications of the 2011 Great East Japan Tsunami on sea defence design. *International Journal of Disaster Risk Reduction*, 14, pp.332–346. Available at: http://dx.doi.org/10.1016/j.ijdrr.2015.08.009.

Rafferty, K.J., 2014. A Comparison Study of Search Heuristics for an Autonomous Multi-Vehicle Air-Sea Rescue System. Available at: http://theses.gla.ac.uk/5292/.

Ramp, S.R. et al., 2009. Preparing to predict: The Second Autonomous Ocean Sampling Network (AOSN-II) experiment in the Monterey Bay. *Deep-Sea Research Part II: Topical Studies in Oceanography*, 56(3–5), pp.68–86.

Redfield, S., 2013. Cooperation Between Underwater Vehicles. In *Marine Robot Autonomy*. Springer, pp. 257–286. Available at: http://link.springer.com/10.1007/978-1-4614-5659-9.

Ren,W. & Cao,Y. 2010. Distributed coordination of multi-agent networks: emergent problems, models and issues. Springer.

Ren, Wei, Beard,R.W, Atkins, E.M., 2007. Information Consensus in Multivehicle Control. *IEEE Control Systems Magazine*, (April), pp.71–82.

Ren, W. & Beard, R.W., 2004. Consensus of Information Under Dynamically Changing Interaction Topologies. , pp.4939–4944.

Ren, W. & Beard, R.W., 2005. Consensus Seeking in Multiagent Systems Under Dynamically Changing Interaction Topologies. , 50(5), pp.655–661.

Ren, W. & Beard, R.W., 2008. Overview of Consensus Algorithms in Cooperative Control. In *Distributed Consensus in Multi-vehicle Cooperative Control*. Springer, pp. 3–22.

Roper, D.T. et al., 2011. A review of developments towards biologically inspired propulsion systems for autonomous underwater vehicles. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, 225(2), pp.77–96.

Rowell, D., 2002. State-Space Representation of LTI Systems. *Analysis and Design of Feedback Control Systems*, pp.1–18.

Rumson, A., 2018. Mapping the Deep Ocean with Multiple AUVs. *Hydro International*. Available at: https://www.hydro-international.com/content/article/mapping-the-deep-ocean-with-multiple-auvs [Accessed May 21, 2018].

Salumäe,T. & Kruusmaa, M.,2013. Flow-relative control of an underwater robot. **469.** *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences.* http://doi.org/10.1098/rspa.2012.0671

Sargent, R.G., 1999. Validation and verification of simulation models. *Winter Simulation Conference WSC Proceedings*, pp.17–28.

Sargent, R.G., 2010. Verification and Validation of Simulation mdels. In *Proceedings of the 2010 Winter Simulation Conferenc*. IEEE, pp. 135–150. Available at: http://www.informs-sim.org/wsc10papers/014.pdf.

Sargent, R.G., 2013. Verification and validation of simulation models. *Journal of Simulation*, 7, pp.12–24. Available at: http://download.springer.com/static/pdf/383/art%253A10.1057%252Fjos.2012.20.pdf?originUrl=http%3A%2F%2Flink.springer.com%2Farticle%2F10.1057%2Fjos.2012.20&token2=exp=1494404050~acl=%2Fstatic%2Fpdf%2F383%2Fart%25253A10.1057%25252Fjos.2012.20.pdf%3ForiginUr.

Schmickl, T. et al., 2011. CoCoRo - The self-aware underwater swarm. In *Proceedings - 2011 5th IEEE Conference on Self-Adaptive and Self-Organizing Systems Workshops, SASOW 2011*. pp. 120–126.

Sfakiotakis, M., Lane, D.M. & Davies, J.B.C., 1999. Review of fish swimming modes for

aquatic locomotion. *IEEE Journal of Oceanic Engineering*, 24(2), pp.237–252.

Sfeir,J., Saad, M., Saliah-Hassane,H., 2011. An improved Artificial Potential Field approach to real-time mobile robot path planning in an unknown environment. *IEEE Internal Symposium on Robotic and Sensors Environments.*

Sjöberg, J. et al., 1995. Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, 31(12), pp.1691–1724.

Smith, R.N. et al., 2010. Cooperative multi-AUV tracking of phytoplankton blooms based on ocean model predictions. *OCEANS'10 IEEE Sydney*.

Soares, J.M. et al., 2013. Joint ASV / AUV Range-Based Formation Control : Theory and Experimental Results. In *IEEE International Conference on Robotics and Automation (ICRA)*. Karlsruhe, Germany, pp. 5579–5585.

Soares, J.M., Aguiar, A.P. & Pascoal, A.M., 2012. *Triangular formation control using range measurements : An application to marine robotic vehicles*, IFAC. Available at: http://dx.doi.org/10.3182/20120410-3-PT-4028.00020.

Sotzing, C.C, Lane, D.M., 2010. Improving the Coordination Efficiency of Limited-Commuication Multi-Autonomous Underwater Vehicle Operations Using a Multiagent Architecture. *Journal of Field Robotics*, 27(4), pp.412–429.

Sozer, E.M., Stojanovic, M. & Proakis, J.G., 2000. Underwater acoustic networks. *IEEE Journal of Oceanic Engineering*, 25(1), pp.72–83.

Stevens, B.. & Lewis, F.., 2003. The Kinematics and Dynamics of Aircraft Motion. In *Aircraft Control and Simulation.pdf*. John Wiley & Sons, Ltd, pp. 1–58.

Stojanovic, M., 2003. Underwater acoustic communications. *Wiley Encyclopedia of Electrical and Electronics Engineering*, 47, pp.435–440. Available at: http://onlinelibrary.wiley.com/doi/10.1002/047134608X.W5411/full.

Stojanovic, M. & Beaujean, P.J., 2016. Acoustic Communication. In *Springer Handbook of Ocean Engineering*. Springer, pp. 359–386. Available at: https://link.springer.com/chapter/10.1007/978-3-319-16649-0_15.

Stokey, R.P., 2005. A compact control language for AUV Acoustic Communication. In *OCEANS 2005 - Europe*. Brest,France, pp. 1133–1137.

Stutters, L. et al., 2008. Navigation technologies for autonomous underwater vehicles. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 38(4), pp.581–589.

Suresh, M. & Ghose, D., 2010. Role of information and communication in redefining unmanned aerial vehicle autonomous control levels. *Proceedings of the Institution of Mechanical Engineers Part G-Journal of Aerospace Engineering*, 224(G2), pp.171–197.

Syed, A. et al., 2007. Understanding Spatio-Temporal Uncertainty in Medium Access with ALOHA Protocols. In *Proceedings of the Second ACM International Workshop on UnderWater Networks ({WUWNet}). Quebec:*. ACM, pp. 41–48.

Tan, C.S., Sutton, R. & Chudley, J., 2007. An integrated collision avoidance system for

autonomous underwater vehicles. *International Journal of Control*, 80(7), pp.1027–1049.

Tangorra, J.L., Gericke, T. & Lauder, G. V., 2011. Learning From the Fins of Ray-Finned Fish for the Propulsors of Unmanned Undersea Vehicles. *Marine Technology Society Journal*, 45(4), pp.65–73.

Tena, I., 2016. A View of Autonomous Underwater Vehicle Market in 2016. *Hydro International*.

Tena, I., 2013. Autonomous Underwater Vehicles - A View of the Autonomous Underwater Vehicle Market. *Hydro International*, pp.14–16. Available at: https://www.hydro-international.com/content/article/autonomous-underwater-vehicles.

Tena, I., 2018. Let Your AUV Do The Talking - AUV Swarms.pdf. *Sonardyna*. Available at: https://www.sonardyne.com/let-auv-talking-auv-swarms/ [Accessed May 21, 2018].

Thenius, R. et al., 2016. subCULTron - Cultural Development as a Tool in Underwater Robotics. Available at: http://zool33.uni-graz.at/artlife/sites/default/files/paper_TheniusEtAl.pdf.

Triplett, B., 2008. *Multiple vehicle coordination and cooperative estimation for target tracking with applications to autonomous underwater vehicle systems*. University of Washington. Available at: http://ezproxy.net.ucf.edu/login?url=http://search.proquest.com/docview/304448882?accountid=10003%5Cnhttp://sfx.fcla.edu/ucf?url_ver=Z39.88-2004&rft_val_fmt=info:ofi/fmt:kev:mtx:dissertation&genre=dissertations+&+theses&sid=ProQ:ProQuest+Dissertations+&+The.

Tsiogkas, N. et al., 2015. Facilitating Multi-AUV Collaboration for Marine Archaeology. In *MTS/IEEE OCEANS Conference*. pp. 1–4.

Tytell, E.D., 2004. The hydrodynamics of eel swimming II. Effect of swimming speed. *The Journal of experimental biology*, 207(Pt 19), pp.3265–3279.

U.S. Navy, 2004. *The Navy Unmanned Undersea Vehicle (UUV) Master Plan*, Available at: http://www.navy.mil/navydata/technology/uuvmp.pdf.

Valdivia y Alvarado, P., 2007. *Design of biomimetic compliant devices for locomotion in liquid environments (Ph. D Thesis)*. Massachusetts Institute of Technology. Available at: http://dspace.mit.edu/handle/1721.1/38927.

Vasilescu, I., Kotay, K. & Rus, D., 2005. Data Collection , Storage , and Retrieval with an Underwater Sensor Network. In *In. Proc. 3rd ACM SenSys Conference*. ACM, pp. 154–165. Available at: https://dl.acm.org/citation.cfm?id=1098936.

Wang, X. et al., 2012. Reviews of power systems and environmental energy conversion for unmanned underwater vehicles. *Renewable and Sustainable Energy Reviews*, 16(4), pp.1958–1970. Available at: http://dx.doi.org/10.1016/j.rser.2011.12.016.

Wang, Y., Yan, W. & Yan, W., 2009. A leader-follower formation control strategy for AUVs based on line-of-sight guidance. In *2009 IEEE International Conference on Mechatronics and Automation, ICMA 2009*. Changchun, China: IEEE, pp. 4863–4867.

Watts, C., 2009. *A Comparison Study of Biologically Inspired Propulsion Systems for An Autonomous Underwater Vehicle*. University of Glasgow. Available at: http://theses.gla.ac.uk/id/eprint/1672.

Watts, C. & McGookin, E., 2008. Modeling and Simulation of a Biomimetic Underwater Vehicle. In *Grand Challenges in Modelling and Simulation*. Edinburgh.

Watts, C.M., 2009. *A Comparison Study of Biologically Inspired Propulsion Systems for An Autonomous Underwater Vehicle*. University of Glasgow. Available at: http://theses.gla.ac.uk/id/eprint/1672.

Watts, C.M. & McGookin, E.W., 2013. Surge performance of an underwater vehicle with a biomimetic tendon drive propulsion system. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, 228(4), pp.315–330. Available at: http://pim.sagepub.com/lookup/doi/10.1177/1475090213483226.

Webb, D.C., Simonetti, P.J. & Jones, C.P., 2001. SLOCUM: An underwater glider propelled by environmental energy. *IEEE Journal of Oceanic Engineering*, 26(4), pp.447–452.

Widditsch, H.R., 1973. SPURV - The First Decade.pdf. *Internal Document - University of Washington*. Available at: http://dtic.mil/dtic/tr/fulltext/u2/a050816.pdf [Accessed January 11, 2017].

Wynn, R.B. et al., 2014. Autonomous Underwater Vehicles (AUVs): Their past, present and future contributions to the advancement of marine geoscience. *Marine Geology*, 352, pp.451–468. Available at: http://dx.doi.org/10.1016/j.margeo.2014.03.012.

Wynn, R.B. et al., 2012. *Investigating the feasibility of utilizing AUV and Glider technology for mapping and monitoring of the UK MPA network. Final report for Defra project MB0118*, Southampton.

Yackoski, J. & Shen, C.-C., 2008. UW-FLASHR : Achieving High Channel Utilization in a Time-Based Acoustic MAC Protocol. In *Proceedings of the Third ACM International Workshop on UnderWater Networks ({WUWNet})*. San Fancisco. Available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.145.8956.

Yang, H. & Zhang, F., 2011. Robust Control of Horizontal Formation Dynamics for Autonomous Underwater Vehicles. In *IEEE International Conference on Control and Automation, ICRA*. Shanghai, China, pp. 3364–3369.

Yoerger, D.R. et al., 1998. Surveying a subsea lava flow using the Autonomous Benthic Explorer (ABE). *International Journal of Systems Science*, 29(10), pp.1031–1044.

Yu, J. et al., 2004. Development of a biomimetic robotic fish and its control algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 34(4), pp.1798–1810.

Zeigler, B.P., Kim, T.G., Praehofer, H., 2000. *Theory of Modelling And Simulation.pdf* Second Edi., Academic Press.

Zhang, F. et al., 2007. Control of coordinated patterns for ocean sampling. *International Journal of Control*, 80(7), pp.1186–1199.

Zhang, Y. et al., 2010. Thermocline tracking based on peak-gradient detection by an

autonomous underwater vehicle. *MTS/IEEE Seattle, OCEANS 2010*, pp.1–4.

Zhao, S., Anvar, A. & Lu, T., 2008. Automatic Object Detection and Diagnosis for AUV Operation Using Underwater Imaging Sonar. , pp.1906–1911.