UNIVERSITY OF SOUTHAMPTON

From Biological Group Behaviour to Underwater Vehicle Team Cooperation

by

Yan Hou

A thesis submitted in partial fulfillment for the degree of Doctor of Philosophy

in the

Signal Processing and Control Group

Institute of Sound and Vibration Research

February 2008

Abstract

Cooperating unmanned underwater vehicles (UUVs) is a popular trend for a variety of work, for example long-range, oceanographic surveying and data collection, even repetitive and dangerous missions for military deployment. A team of low-cost underwater vehicles could cover an area quickly and effectively. Team advantages appear obvious when cooperation is applied since a team could achieve an exploratory mission quicker than a single vehicle and in particular a single vehicle has a much greater chance of being lost or of developing a fault which could destroy the mission plan.

Animals often have behaviour which aims to maintain themselves living as groups. Fish schooling is a typical group behaviour and may have lessons to offer the development of team cooperation of UUVs. The idea of this study is inspired from some animals' group behaviour and their coupling modes, especially fish schooling, and focuses on the feasibility and the possibility of applications with a group of underwater vehicles.

Previous work on the development of SUBZERO III, a small, low-cost UUV, has described the dynamic model of the vehicle and this model forms the basis of the vehicle dynamics in a new simulation model design. In order to develop UUV team working, we investigate a modified, behaviour-based group control algorithm and simulate with the dynamic model of SUBZERO III. The behaviour-based control rules in the algorithm are classified with different priority weights. Higher priority rules have higher priority weight values which have a greater effect on the next decision step. In order to adaptively estimate real-time priority weights according to the situation that the vehicles meet, a fuzzy logic control method is used.

Mission scenarios of differing complexities are simulated to assess the stability and

reliability of the group control method. First, the water tide flow, which affects the UUV movement, is considered in the simulation. Second, line formation control and circle formation control are simulated for situations when the mission requires UUVs to remain in a formation.

To conclude, we have demonstrated that the behaviour-based method with fuzzy logic priority weights can be successfully applied for control a team of cooperative unmanned underwater vehicles with a dynamic model of a real UUV, even when vehicles are in a tidal flow environment and when vehicles are in a formation pattern.

Future work of the project is presented including the need for communication sensors and communication network strategies. The feasibility of implementation of the method on a real team of UUVs in practice is addressed.

Acknowledgements

I am grateful to my supervisor, Professor Bob Allen, for a financial support for my research, following my progress, and his constant suggestions during this research.

I would like to thank some former and present members of signal processing and control group.

Finally, I would like to appreciate my wife and my families. I will not finish without their unconditional love and support.

This work has been carried out with the support of the UK MoD Data and Information Fusion Defence Technology Centre under Project DIF DTC 12.8.1.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this introductory chapter, we give the objectives of the project and the contributions we make. The chapter is organized as follows: In section 1.1 we give a short introduction to unmanned, underwater vehicles, which is the motivation behind this research. The contributions of this research are summarized in section 1.2. Finally, the structure of this thesis is presented in section 1.3.

1.1 Motivation and objectives

Today, unmanned underwater vehicles (UUVs) are a reality. Highly sophisticated and complex unmanned, underwater vehicles have been applied to explore physical areas where humans are unable to go due to many constraints, such as in the deep, dangerous and unpredictable ocean. UUVs have become vital resources in military R&D efforts and expensive oceanographic exploration as in ocean surveying, tracking fish populations, monitoring environmental conditions, underwater rescue, and searching for underwater mines. UUVs have also been actively employed in the commercial realm, particularly to assist the oil and gas industries in surveying deep-water regions for potential new energy sources (Ouellette 2002). However, such an UUV is usually expensive due to the very high development cost. From an operational point of view, the deployment and retrieval costs can be enormous, often requiring a support ship and crew and appropriate launch and retrieval systems (Allen 2005). For some missions, a single vehicle may become inefficient in a complex environment or in repetitious tasks.

A popular recent research trend is to investigate the feasibility of a team of cooperating UUVs for work, such as long-range oceanographic surveying and data collection, oil field inspection and maintenance. A team of low-cost, small vehicles could cover an area quickly, e.g. for pollution detection and clearance. The problems then become how we plan and coordinate a mission and control such a team. What is the best method of interacting? How can sensor information be used most effectively in order to achieve a mission? In command, control and mission management, which UUV is 'mission leader' and how should the team be coordinated, particularly since communications would be minimal due to communication difficulties underwater and for covert operations, and navigation has to take place without continuous GPS. These questions provide the motivation for this project.

In order to solve these problems, researchers have tried several different approaches. Recently, researchers have become increasingly aware of animal group behaviour which is evident in nature and may offer clues to the best way forward since they have passed natural selection over many thousands of years. Typical examples of cooperative behaviour are ants foraging, birds flocking and fish schooling. Learning animal group behaviour and applying the principles to real engineering applications could lead to a shortcut to achieving cooperative missions in real situations. Natural group control methods lead to better performance than traditional control methods. Learning from biological group control behaviour is the main motivation for this project.

The main objective of this project is to use behaviour-based rules inspired from animal behaviour on a team of underwater vehicles based upon a realistic dynamic manoeuvring model. The suitability of the method is to be justified under the following conditions: the parameters change, such as the minimum distance between teammates and the minimum distance between vehicle and obstacle; the environment becomes more complex, for example, the tidal flow is considered; the mission is more difficult, such as keeping team in line and circle formations.

1.2 Contribution

In this thesis we illustrate the trend of the research from single to multiple unmanned, underwater vehicles. We propose a behaviour-based method inspired by animal group behaviour. The behaviour-based rules are classified by priority weights. The higher priority rules have a higher contribution on the steering decision. We apply the fuzzy logic method to adapt the relative priority weight values according to the situation the vehicles meet. In order to verify the the feasibility of the approach, the dynamic manoeuvring model of a real underwater vehicle is embedded in the simulation. Furthermore, this approach is investigated on UUV team missions in the complex environment with water flow. The approach is also tested when a team of vehicles are in line and circle formations patterns.

1.3 Publications

Journal paper:

1. Y. Hou and R. Allen, Intelligent behaviour-based team UUVs cooperation and navigation in a water flow environment, *Ocean Engineering*, 35(3-4), 2008, 400-416

Conference papers:

1. Y. Hou and R. Allen, Behaviour-based Rules with Fuzzy Logic Controlled Priority Weights in Multi-UUVs Team Cooperation, *IEEE OCEANS 2007 - Europe*, 18-21 June 2007, 1-6.

2. Y. Hou and R. Allen, Circle Formation Control for a Team of Cooperative UUVS by Behaviour, *Biological Approaches for Engineering March 2008*, Southampton, 17-19 March 2008.

3. Y. Hou and R. Allen, Behaviour-based Circle Formation Control Simulation for Cooperative UUVs, *NGCUV 2008*, Killaloe, Ireland, 8-10 April 2008, accepted

4. Y. Hou and R. Allen, Multi-UUVs Team Formation Control by a Behaviour-based Method with Fuzzy Logic Adapters, *CIMTEC 2008*, Italy June 8-13, 2008

1.4 Outline of the Thesis

Chapter 2 describes some possible team UUV cooperative application scenarios and illustrates their potential advantages and disadvantages, compared with single UUV applications. We review the recent team cooperative applications and the methods that have been used.

Chapter 3 reviews and summarizes some biological collective behaviour cases. Collective behaviour of some biological organisms and the coupling modes they apply are introduced, together with some simple behavioural rules. The similarities between a group of fish and a team of UUVs are emphasized. We compare the communication modes between animals with UUV's communication and sensor technologies.

Chapter 4 illustrates the basic Boids method which inspired the approach developed in this project and describes a weighted, behaviour-based, group control method. A simple simulation scenario integrated with the dynamic maneuvering model of the SUBZERO III underwater vehicle (Feng et al. 2003) is implemented and the results are presented and discussed.

Chapter 5 designs fuzzy logic controllers instead of constant priority weights, to estimate the priority weights in real-time. The simulation results are tested with different parameters settings.

Chapter 6 complicates the mission scenario environment by adding the effect of water flow. The simulation results are shown and discussed.

Chapter 7 presents the implementation of UUV formation control by reactive behaviour. The formation behaviours are integrated with other collective rules and navigation rules to enable a team of UUVs to reach navigational goals, avoid hazards and simultaneously remain in formation. The line and circle formations are investigated.

Finally, the conclusion of this project and possible future work are presented in Chapter 8.

Chapter 2

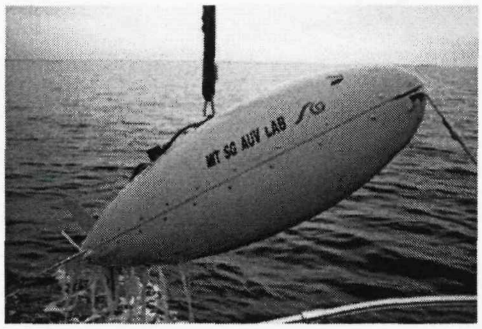
Background of UUV Team Cooperation

This chapter reviews previous and current research of both single and multi UUVs. In order to elaborate the capabilities and feasibilities of multi UUVs deployed in applications, the advantages and disadvantages of single UUV and multi UUVs applications are examined. Finally, the chapter reviews some cooperative algorithms for multi-robot applications.

2.1 Underwater vehicles review

2.1.1 Single UUV

Unmanned underwater vehicles are variable size, autonomous, untethered submersibles. They are intended to provide researchers with a long-range, low-cost, rapid response capability to collect pertinent environmental data. Using an UUV can be more economical than using a large ship. In a rescue mission, an UUV can be more quickly deployed to accomplish the mission. There are numerous applications for UUVs, such as oceanographic surveys, operations in hazardous environments, underwater structure inspection, and military deployment. We only summarize several typical UUVs applications here since we focus on multi-UUV applications.



(a) Odyssey II AUV

(b) AUTOSUB AUV

Figure 2.1: Single AUV examples: Odyssey II and AUTOSUB

The ARPA (Advanced Research Projects Agency) UUV program was developed and used in 1988 when two large UUVs were built for tactical naval missions, particularly ocean minefield search by Charles Stark Draper Laboratories. These vehicles are the largest, the most capable and the most expensive UUVs built to date. The UUVs were the first to accomplish many important UUV tasks, but the project was suspended due to the high cost of vehicle support (Brancart and Claude 1994). The MIT underwater vehicles Laboratory Sea Grant College Program has built a series of *Odyssey* AUVs[1] for a number of years (Bellingham et al. 1994, Fricke 1994). Odyssey II (see Figure 2.1(a)) has the maximum stable forward speed of 0.5m/sec, and operational missions follow a cruise or survey profile. Vehicle unit costed $75,000 high although it has been used for demonstrating operational missions in rivers, in ocean and under Arctic ice (Bellingham et al. 1994, Fricke 1994, Brancart and Claude 1994). In order to improve the quality of data collected on survey missions, Monterey Bay Aquarium Research Institute (MBARI) and the Stanford Aerospace Robotics Laboratory built a hybrid ROV-AUV capable of traveling and then plugging a cable into a seafloor network to become an ROV for oil and gas exploration (Marks et al. 1992; 1994; July 19-20 1994). The National Oceanography Centre in Southampton launched an AUV project called AUTOSUB. AUTOSUB (see Figure 2.1(b)) is a long range, deep diving AUV with a wide range of possible applications. The vehicle is 7m long x 1m diameter and weighs 1500kg in air (Babb 1993). The maximum depth rate is 1600m and the maximum range is 800km. AUTOSUB has been successfully applied for ocean data collection and under-ice survey (Collar and McPhail 1995).

However, the technical risks increase with the development of UUVs. UUVs use many complex systems, each of which can present risks of failure with the ultimate consequence being the loss of the vehicle (Manley 2007). A major system critical to all UUVs is the stored energy required to operate the vehicle. This energy problem increases the possibilities of the loss of vehicles. A single UUV is usually expensive due to the very high development cost. Loss of the vehicle can be devastating due to the high economical value. In 2005, Southampton Oceanography Centre (SOC)

[1]Autonomous Underwater Vehicles

reported that their first Autosub has been lost under in the Antarctic some 17km from the edge of the ice shelf beneath ice over 200m thick. From an operational point of view, the deployment and retrieval costs can be enormous, often requiring a support ship and crew and appropriate launch and retrieval systems (Allen 2005). For some missions, a single vehicle may become inefficient in a complex environment and a repetitious task.

In conclusion, most of work so far has focused on control of a single agent, but increased efforts have begun to address systems composed of a team of low-cost small vehicles. Compared with a single expensive UUV, a team of low-cost, small vehicles could offset the insufficiency and risk of a single UUV. Next, a team of UUVs is discussed.

2.1.2 Team of UUVs

The employment of multiple UUVs has become a popular interest due to the significant advantageous reasons for both military and commercial applications. There are several advantage for this increased interest. Some tasks may be inherently too complex for a single UUV to accomplish. Multiple UUVs provide the ability to survey large ocean areas more rapidly and economically than can be accomplished with a ship or a single UUV, and increased operational envelopes due to weather insensitivity. The overall performance, robustness and flexibility of the system is strengthened. Using multiple UUVs rather than a single UUV can have several advantages but leads to a variety of design tradeoffs. Each collective individual UUV can be simple in terms of individual physical design rather than a large and complicated UUV. The whole system becomes more economical and more scalable. The chance of overall failure can be decreased by multiple UUVs. In order to illustrate the above advantages, some possible team UUVs application examples are discussed.

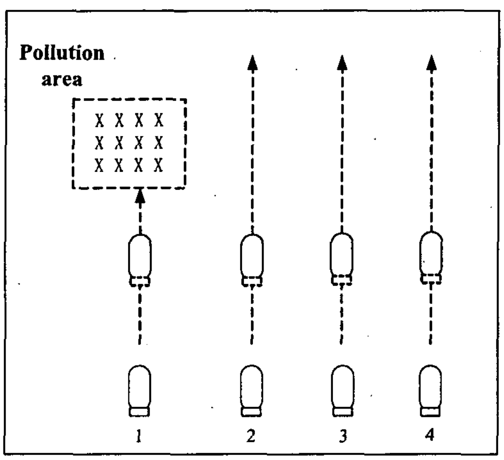
2.1.3 Team of UUVs: possible applications

Several team UUVs mission scenarios are introduced in Table 2.1 and the advantages are given for each scenario.

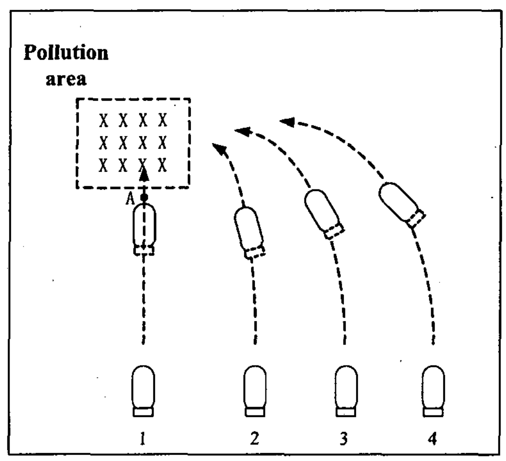
Group behaviour scenario	Advantages
Environmental monitoring	A group of vehicles have more monitoring range than single one. They can rapidly survey large areas.
Oceanographic exploration	A team of vehicles can share information collected at geographically different places and simultaneously carry out actions at these places.
Underwater rescue	May accomplish tasks that cannot be accomplished by a single vehicle (ex. Transporting a heavy object whose weight exceeds the individual's capacity)
Mine detection and clearance	A team of vehicles can cover a search area quicker than a single vehicle and can share mine position information when a mine is located. A team has inherent redundancy during mine clearance.
Group of torpedoes	Torpedoes can track one or more moving targets. A group has more chance to attack the targets than a single torpedo. If one of them is intercepted, others still can keep tracking. A group has less chance of being intercepted than a single torpedo.

Table 2.1: Mission scenarios for a team of UUVs

Pollution surveying and tracking has been investigated over many years. Traditional map search methods have been used for pollution tracking. However, the new method of 'swarm collective search' can do much more intelligent team work. In Figure 2.2(a), each vehicle only searches by following the direct trace before it finds the pollution area. Eventually only vehicle 1 can detect pollution area and search the range within the area. In Figure 2.2(b), at **A** point, vehicle 1 transmits a signal to other vehicles to notify them the location of the pollution area and vehicles 2,3 and 4 then all turn to the direction of pollution area. When all vehicles arrive the pollution area, they can search the pollution area range together. Obviously, a group of vehicles spend less time on detection of the pollution area range than only one vehicle. The method in Figure 2.2(b) does a more effective job than 'traditional map search'.



(a) Map search mode. Each vehicle only search by following the direct trace before it finds the pollution area. Eventually only vehicle 1 can detect pollution area and search the range within the area.



(b) Collective search mode. At A point, vehicle 1 transmits signal to other vehicles to tell them he has found the pollution area. Vehicle 2,3 and 4 all turn to the direction of pollution area in order to determine its extent.

Figure 2.2: Pollution tracking by swarm vehicles

In the military field, torpedoes are widely employed. A torpedo is an UUV with a high cruise speed. A group of torpedoes has more chance to attack a target and have less chance to be blocked than a single torpedo. In Figure 2.3, torpedoes A and B are destroyed by perhaps, a mechanical problem or blocked by the enemy's missile.

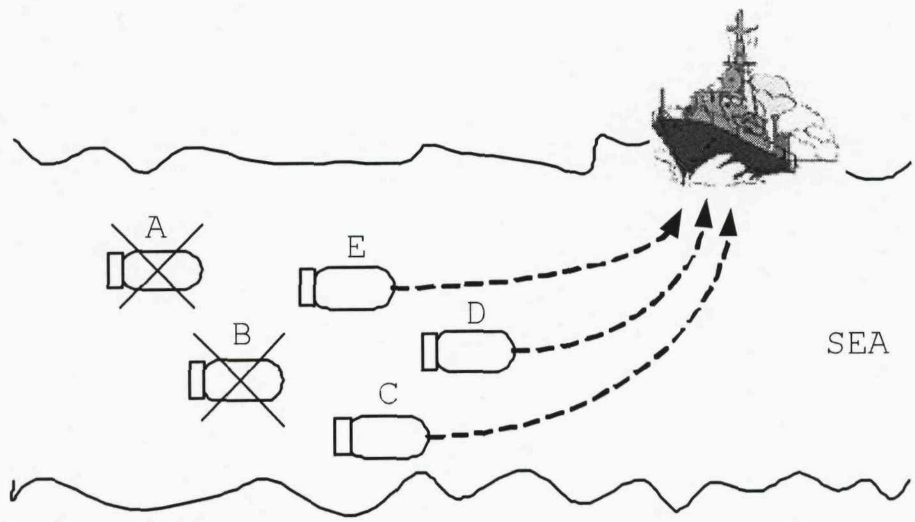


Figure 2.3: Cooperation of a team of torpedoes

Nevertheless, torpedoes C, D and E are still able to attack the warship and complete the mission. This superiority appears in biological behaviour. A fish predator can only eat a few fish; most fish in a school will dodge the predator and keep going. However, the requirements of sensor response time and accuracy are extremely strict because the torpedoes' speed may be much higher than a normal UUV. In practice, different requirements of sensors are subject to different specifications which include group size, distance between each UUV and speed of UUVs.

Overall, the above examples of cooperative underwater vehicle teams have indicated the significance and advantages for commercial and military purposes. The next section reviews the applications of current team cooperation and the methods used to achieve this behaviour. The review of applications is classified by three types of applications: computer graphic multiple agents simulation, cooperative multiple robots and cooperative team of underwater vehicles. The review of methods implemented is classified by the two aspects: non-behaviour-based and behaviour-based.

2.2 Review of team cooperation applications and methods

The cooperative multiple mobile robot agents and their applications have been widely developed by computer graphic programmers and engineering researchers. In the virtual environment and game community, the most common approach to simulating group movement is to use flocking. The concept of flocking **Boids** was introduced by Reynolds (Reynolds 1987). He simulated a flock of birds which each bird was implemented as an independent "actor" that navigates according to its local perception of the dynamic environment. Artists created complex interactive systems and simulated crowds. Tim Burton's "Batman Returns" was the first movie to make use of this kind of technology for rendering, realistically depicting the movements of a group of penguins using the Boids system. The Lord of the Rings film trilogy made use of similar technology, known as Massive, during battle scenes. Kamphuis and Overmars (2004a;b) carried out an experiment that generated paths exhibiting group coherence as often required in games, like passing on the same side of obstacles and waiting for fellow teammates to catch up. The experiments also simulated the behaviour of whole armies with varying sizes in a computer game.

Cooperative mobile robotics research has already been applied in industrial, commercial and scientific applications. We review mobile robots or simulation of mobile robots, where a mobile robot is taken to be an autonomous, physically independent, mobile robot. Most mobile robots are represented by ground and air vehicles. Enabling multiple robots to cooperatively carry, push, or manipulate common objects has been a long-standing, yet difficult, goal of multi-robot systems. Many works have addressed the box-pushing problem, for widely varying reasons. Donald et al. (1994) studied two box-pushing cases in terms of their internal communication and hardware requirements. Cooperative manipulation of large objects is particularly interesting in that cooperation can be achieved without the robots even knowing of each others' existence (Sen et al. 1994). Other works in the class of box-pushing/object manipulation include (Wang et al. 1994, Brown and Jennings 1995, Mataric et al. 1995) and (Sasaki et al. 1995). If we study the team behaviour of ants, we notice that the ants always congregate together to raise a small stone which is many times heavier than an ant. We may learn from ants to solve the box-pushing problem.

A team of autonomous mobile robots were established in Reading University (Hu et al. 1998). The robots are equipped with an infrared communication system as well as ultrasonic sensors for detecting obstacles. Each robot is physically small, with a width of 140mm, a length of 130mm and a height of 140mm. Four robots were conducted to work in a team with shared experience. Inspired from this experiment, sharing experience through a communication system could be used for a team of UUVs.

Competition in multi-robot systems, such as that found in higher animals, including humans, is being studied in domains such as multi-robot soccer. Asada et al. (1999) discussed many of the advances in Artificial Intelligence on RoboCup; and Noda et al. (1998) and Tambe et al. (1999) presented different behaviour for different roles, such as offensive for a forward, defensive for a defender, goal saving for a goalie. In a team of UUVs, we could assign different tasks for different roles. For example, in a pollution surveillance mission, some UUVs could take responsibilities for pollution searching, some UUVs could take in charge of transmitting the pollution information signal to a surface ship.

Several research have been carried out in the area of localization, mapping, and exploration for multi-robots teams. One example of this work was given in Fox et al.

(2000). Stergios and George (2002), Madhavan et al. (2004) enabled a group of mobile robots to be simultaneously localized by sensing their teammates and combining positioning information from all the team members. Another example was provided by Schmitt et al. (2002) who developed and analyzed a probabilistic, vision-based state estimation method that enabled robot team members to estimate their joint positions in a known environment.

Ferri et al. (2006) developed a group of small (17cm x 17cm) robots and applied a cooperative robotic system to localize a gas source in an indoor environment with no strong airflow. A scenario of an air vehicle swarm searching for and mapping a chemical cloud within a patrolled region was simulated by A.Kovacina et al. (2002). If a cloud is detected, the UAVs (Unmanned Air Vehicles) can take many actions, such as attempting to map the size and density of the cloud. Also, if a UAV finds the contaminated cloud, it must not return to the home base, but to a predefined decontamination centre.

In the meantime, the cooperative robotic system has been considered for military applications. Today UAVs provide a safe and cost-efficient way of performing many tasks, such as reconnaissance and battle damage assessment. In order to fully realize the capabilities of UAVs, multiple UAVs need to be deployed simultaneously and they must have the ability to realize the presence of the other UAVs. Orbital Research is currently developing several coordination algorithms that to control groups of UAVs to perform such tasks as formation flying, dynamic task assignment, reconnaissance, and chemical cloud tracking (Orbital 2003). Cassinis et al. (1999) used teams of co-operating robots to detect of anti-personal landmines. The simulation results showed that coordinated searching strategy had a better performance than a random search strategy. Formation control is further explored in chapter 7.

2.2.1 Current UUV team cooperation applications

Besides the applications on the ground and in the air, a team of cooperating unmanned underwater vehicles (UUVs) has the potential to dramatically enhance military and scientific missions in marine environments. They could locate, map, and track three-dimensional time-varying phenomena, they can quickly survey large areas, and they can adaptively measure areas of interest with high fidelity. Section 2.1.2 has described the advantages of cooperative multiple UUVs applications. This section lists several current applications of UUVs cooperation.

A low-cost small underwater vehicle, URIS (Batlle et al. 2001), has been developed in the Institute of Informatics and Applications of the University of Girona. Each vehicle provides an ability to survey shallow coastal waters and waters up to 100m in depth. Multiple such vehicles have been designed and tested in virtual environment for marine and underwater environmental surveillance and protection. Serafina is the name given to small light-weight submersibles currently being developed in Australian National University (Kalantar and Zimmer 2004). Their primary purpose is to investigate the possibilities of creating schools of autonomous underwater vehicles capable of exploring the underwater landscape.

The U.S. Navy has become interested in employing multiple intelligent UUVs for the Underwater Mine Counter-Measures (UMCM) problem. Bishop (2004) developed numerous UMCM UUVs to improve efficiency and shorten time taken to conduct searches for mines while allowing greater robotic autonomy and cooperation without endangering people.

Multiple UUVs has been used for enhancing connectivity in underwater ad-hoc sensor networks (Seah et al. 2005). Underwater sensor networks typically comprise sensor nodes that are deployed in sufficiently large numbers for data collection, monitoring and surveillance. The acquired data is relayed by the sensors over multihop wireless acoustic communications links to sinks and collection points. Seah et al. (2005) utilized multiple underwater unmanned vehicles to enhance connectivity. The UUVs patrol the areas where connectivity is likely to be poor to overcome temporal interference and if necessary deploy more sensors to repair the breaks in connectivity. In the event that the network becomes partitioned, the UUVs can also serve as local sinks to the sensors in the isolated partitions, and ferry the data from the isolated sensors to the nearest connected part of the network. The results have shown that there is great potential in further improving the performance of underwater sensor networks, e.g. cooperation between mobile and static nodes for localization and networking, as well as, efficient use of UUVs as message ferries to bridge between nodes.

In conclusion, engineers and researchers have increased interest in cooperative robot and vehicle applications from commercial to the military fields. However, compared to the current state of development, and the many applications of single UUV's, most multiple UUV applications are still at the research and development stage. From this research on multiple robot and vehicle applications, we put forward the idea that developing multiple UUV team has potential for underwater applications.

2.2.2 Current control methods for team coordination

The design of the control method is crucial for the successful development of cooperative vehicle teams. There are many different control strategies that have been considered for the coordinated control of multiple vehicle systems. The taxonomy for the classification of these control and coordination methods can be divided into two aspects: non-behaviour-based and behaviour-based methods.

There are only a few cooperative algorithms which are not bio-inspired, such as the probabilistic roadmap (RPM) (Bayazit et al. 2002) method and simultaneous localization and mapping (SLAM) (Williams et al. 2002). These algorithms require that accurate map data is available for their execution. However, the availability of accurate underwater maps is the biggest problem since the underwater environment is variable and unpredictable, for example tidal flow will complicate the situation. Although an entire map could be built by local submaps, it needs much more bandwidth to transfer the local map information. Therefore, the algorithms which are based on underwater maps are not currently suitable for underwater applications.

Nearly all of the work in cooperative mobile robotics are related to behaviour-based control. Behaviour-based algorithms have had a strong influence in much of the cooperative mobile robotics research to date. Because the behaviour-based paradigm for mobile robotics is rooted in biological inspiration, many cooperative robotics researchers have also found it instructive to examine the social characteristics of insects and animals, and to apply these findings to the design of multi-robot systems (Cao et al. 1997).

The most common methods use simple local behaviour rules inspired from various biological societies, particularly ants, bees, and birds. The development of similar behaviour is transferred into cooperative robot systems. Many areas of biological inspiration and their applicability to multi-robot teams are fairly well understood since the behaviour-based rules are easier than mathematical models.

The first small contribution for the basic collective behaviour is made by Craig Reynolds in 1987 (Reynolds). Reynolds summarized three basic behaviour rules (known as **Boids**) to simulate bird flocking. As the basic idea formed the inspiration for this project, the **Boids** rules will be described in Chapter 4. Stephens et al. (2003) added more behavioural rules to **Boids** in order to create artificial fish to act autonomously in response to sensory input from the environment and from other fish. Joshua and Gary (2004) used the **Boids** model to simulate unmanned aerial vehicles swarm scenarios. Both works investigated that the basic **Boids** model can be efficiently used for animation simulations.

In 1995, Kennedy and Eberhart (1995) studied the social behaviour of birds flocking and fish schooling from which Particle Swarm Optimization (PSO) was developed as an evolutionary computational technique. The PSO system is initialized with a population of random solutions and searches for optima by updating generations. To apply PSO on underwater vehicles, Sheetal and Venayagamoorthy (2004), Doctor et al. (2004) simulated searching a given problem space for a target. However, the PSO algorithm is most likely used for computing optimization and is not a solution of cooperative control for a team of UUVs.

Ant behaviour has been summarized to generate the local pursuit algorithm and to control a group of small robots to push a box (Hristu-Varsakelis and Shao 2004, Kube and Bonabeau 2000). 'Local pursuit' allows members of the group to overcome their limitations with respect to sensing range and available information through the use of neighbour-to-neighbour interactions. Local pursuit enables the group to find an optimal solution by iteratively improving upon an initial feasible control. However, like the PSO algorithm, the local pursuit algorithm is not a applicable solution for a team of cooperative UUVs because it does not consider obstacle and collision avoidance, and it is usually used as a computational optimization method. Zhang et al. (2007) employed a simple self-reinforcement learning model inspired by the behaviour of social insects to differentiate the initially identical robots into 'specialists' of different tasks. Although the tasks were assigned to different levels, the task assignment mechanism could not be adapted to the change of environment.

Carreras et al. (2000) summarized some typical behaviour, such as go to, obstacle avoidance, wander, station-keeping, target-following, target-tracking, avoiding trapping and bottom-following, and two of the most suitable architectures to guide AUVs: Schema-based approach and Subsumption architecture. In each architecture, each behaviour generates a 3-dimensional vector that denotes the direction to be followed by the vehicle. In the Schema-based approach, each schema operates as a behavioural intension. The coordination method consists of vector summation of all motor schema outputs and normalization. The Subsumption architecture is a method of reducing a robot's control architecture into a set of task-achievement behaviour or competence represented as separated layers. Individual layers work on individual goals concurrently. Layers are organized allowing higher layers to inhibit the outputs of lower layers. The paper summarized that the Subsumption architecture has better robustness and worse performance because it has only one active behaviour. Schema-based approach has better performance and worse robustness because all behaviour rules are active. Bishop (2004) presented that the Subsumption architecture did more robust control for robots performing underwater mine countermeasures because only one active behaviour can avoid mines more quickly. However, when the vehicles must implement multiple tasks simultaneously, neither method can achieve robustness and performance together.

To summarize, we could combine Schema-based approach and Subsumption architecture for multiple vehicle control to achieve better performance and robustness. Inspired from behaviour-based methods, we use a novel behaviour-based approach to control UUV team cooperation missions in different complex underwater environments. In order to compute the priority weights for behaviour rules according to the situation that vehicles meet, the fuzzy logic method is applied in chapter 5. The combining of the behaviour rules determines the final steering direction of next step. The objective of this method would be to manoeuvre and direct a team of UUVs more effectively with more qualitative functionality under different conditions.

2.3 Summary

In this chapter, we started with the review of the development of a single UUV and multiple UUVs. The comparisons between a single UUV and multiple UUVs have been outlined to indicate the trend of using multiple UUVs. Then we explained some possible mission scenarios of multiple cooperative UUVs to identify the advantage of multiple UUVs. Some team cooperation application and methods have been reviewed and the behaviour-based methods have been focused on to lead to our behaviour-based method. Finally, we gave a brief introduction of behaviour-based method we use in the following chapters.

Chapter 3

Review of the Biological Basis for Co-operative Behaviours

This chapter gives an introduction to self-organization from a biological perspective. We also will describe some biological cases such as ant trail formation, bees nectar source selection, bird flocking and, especially fish schooling, because fish schooling behaviour is probably the most similar to underwater vehicles team cooperation. In order to indicate the feasibility of using biological behaviour, we express the similarities between fish schools and a team of UUVs. Also we give a comparison of the communication modes between animals and between UUVs.

3.1 Biological background of group behaviour

A wide variety of insect and animal behaviour is seen in nature. A flock of birds fly across the sky. A group of ants forage for food. A school of fish swims, turns and flees together,etc (Liu and Passino 2000). This kind of 'collective group behaviour' is so potentially valuable that biologists and computer scientists in the field of 'artificial life' have studied how to model biological group to understand how such 'social animals' interact, achieve goals, and evolve. Moreover, engineers are increasingly interested in this kind of group behaviour since it may be applied in, for example, optimization, robotics, traffic patterns in transportation systems, and military applications (Liu and Passino 2000).

3.1.1 Self-Organization Definition

What is **Self-Organization**?

The clear definition of self-organization is expressed as: Self-organization is a process in which pattern at the global level of a system emerges solely from numerous interactions among the lower-level components of the system. Moreover, the rules specifying interactions among the system's components are executed using only local information, without reference to the global pattern. In short, the pattern is an emergent property of the system, rather than a property imposed on the system by an external ordering influence (Camazine et al. 2003).

We are interested in the mechanisms and behaviour of grouping in animal groups. Our goal is to explore how the individual animal decides to move, stop, or orientate itself to keep themselves within a group. Some typical animals with group behaviour, such as ants, bees, birds and fishes, are discussed.

The general information of above the animals with beetles aggregation and spiders prey capture activity are summarized in Table 3.1 which enables a comparison of collective behaviour to be made across the different animal species established.

Animals	Collective process	Coupling Mode
Ants	Foraging activity and more	Pheromone and Mechanical contact
Bees	Nectar source selection and group decision making	'waggle dance' and pheromone
Fish	Coordinated movements of fish in a school	Vision and Lateral line system
Beetles	Clustering Process feeding aggregation	Chemical
Spiders	synchronous prey capture activity	Mechanical web vibrations

Table 3.1: Summary of collective behaviour across the different animal species

3.1.2 Trail formation in ants

One study of swarm group behaviours investigated the foraging behaviour of ants (Bonabeau and Theraulaz 2000). Although the capabilities of a single ant are very limited, ants can collectively establish the shortest route between a source of food and their nest, and efficiently move the food to their home (Camazine et al. 2003). Ants communicate with each other through the use of pheromones; chemical substances that attract other ants. As the ants move, they lay down a trail of these pheromones that other ants can follow (Tarasewich and McMullen 2002). The probability that a certain path will be chosen significantly depends on the amount of pheromone that has been deposited on that path. The higher the amount of pheromone, the higher is the probability that a path is chosen (Lučić and Teodorović 2002).

Figure 3.1 shows a simple model which is simulated by computer (Bonabeau and Theraulaz 2000). At first the ants explore their environment randomly. Then they

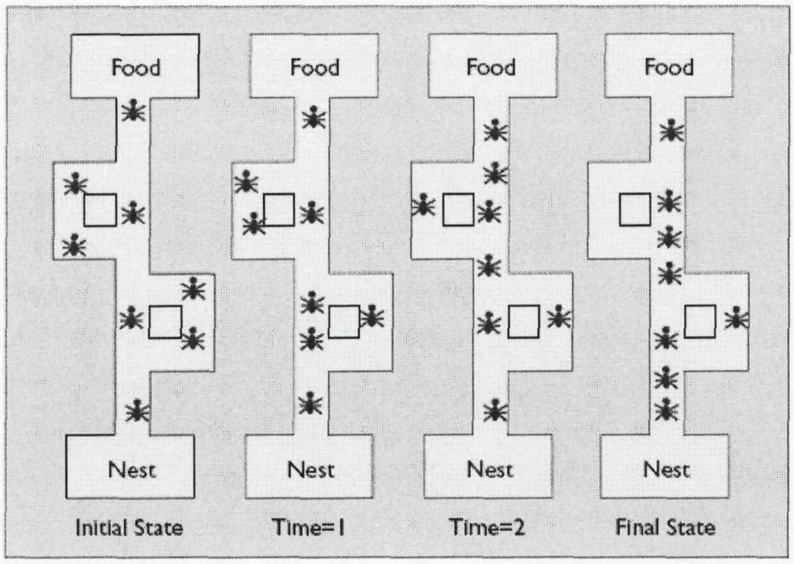


Figure 3.1: Distribution over time of ants over a set of pathways between a nest and a food source

establish trails that connect all of the food sources to the nest. Next they keep laying down pheromone on the trails of the sources which are the closest to the nest and most ants then follow the shortest trail which has the stronger pheromone signal.

However, there is a chance that an ant will not follow a previous or well-marked trail (Bonabeau and Theraulaz 2000). This is beneficial in not only allowing the discovery of shorter or alternate pathways, but also new sources of food. The pheromones also evaporate over time.

There are several available applications based on ants biological behaviour. Researchers have tried to use an ant inspired system (AS) to optimize problems such as the asymmetric traveling salesman problem (TSP) (Lučić and Teodorović 2002, Dorigo et al. 1996). Bullnheimer et al. (1999) used an ant approach to investigate a vehicle routing problem. The goal of this problem was to minimize the cost for vehicles going from a central depot to a series of customers. The results showed that an ant-based method could successfully construct a good solution.

3.1.3 Nectar source selection by honey bees and group decision making

In spring and summer, lots of honey bees are dispatched into the surrounding countryside to gather the colony's food (Camazine et al. 2003). Bees 'waggle dance' to communicate locations of food found when foraging, and of potential nest sites when a colony moves during swarming (Visscher 2003). Nest-site choice by a honey bee swarm is an impressive example of group decision making. When bees leave their colony, scouts search the countryside for sites. They then return to the swarm, and communicate the distance to and direction of the sites that they have found with waggle dances.

The cooperative food gathering behaviour of bees is determined by information provided by the forager bees. Yonezawa and Kikuchi (1996) simulated an algorithm which contained the information selection rules. Rules are that bees select honey by quality and distance from the honeycomb to the feeding area. They showed a scheme which can explain the process of bees collective intelligence (Figure 3.2).

There are several models to describe how honeybee swarms select a new nest-site. The model named 'SMALLTALK' was simulated by de Vries and Biesmeijer (1998). In this simulation, they obtained a set of rules that was necessary and sufficient for the generation of the collective foraging behaviour observed in real bees. The simulation outcome agreed with the observed data.

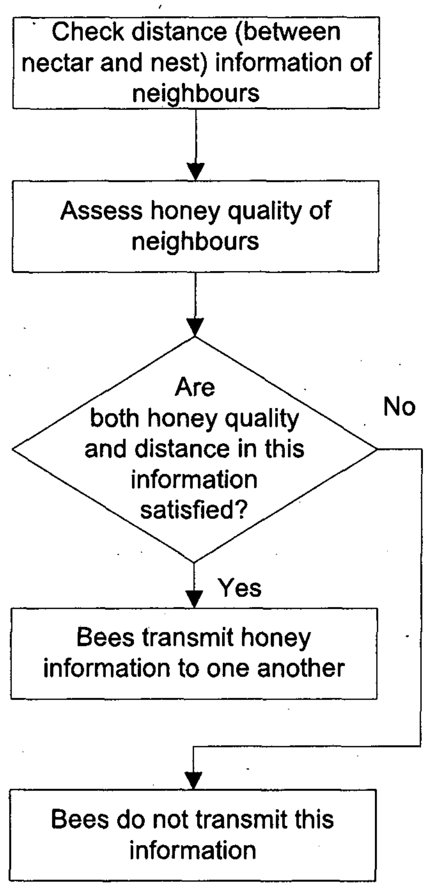


Figure 3.2: Selection rules of honey source information

3.1.4 Fish schooling

A term of underwater vehicles could, perhaps, be regarded as analogous to a school of fish. Schooling of fish is one of the most familiar forms of animal social group behaviour. A school of fish swims and turns together. Schooling fish maintain a remarkably constant geometric orientation to their fellows, heading in the same direction when they are schooling for most of the time. Their bodies are parallel and with virtually equal spacing from fish to fish (Shaw 1962).

Introduction of fish schooling

The whole school is the leader and each individual is a follower (Partridge 1982). Speed increases with size and the fish of a species therefore tend to sort themselves out by size and by generation in the sea. Fish behaviour is integrated. There is no central "control system". School organization must be dominated by internal factors in order to maintain stability.

Many advantages can be cited in favour of the behaviour to show why it is so effective.

1. When a shark attempts to attack a group of fish, all the fish within the group will flee for their lives. The shark has to try to target one of many fish to catch. Therefore increasing group size will decrease the chance of any individual fish being caught and killed by a predator.

2. It makes finding food easier; the sight range of all the fish in a school is larger than that of a single fish. Once a fish finds food, others will be quickly informed of where the food is located by transiting the food location information.

3. The potential advantage in hydrodynamics effect in schooling provides a more efficient way for an individual fish to move through the water (Partridge 1982).

Coupling Mode of schooling behaviour

To maintain its position in a school each fish must constantly monitor the positions and velocities of its neighbours. Most schooling fish rely on vision and the lateral line[1] system (Partridge 1982). If fish are both blinded and have their lateral lines

[1] In fish, the lateral line is a sense organ used to detect movement and vibration in the surrounding water (Partridge 1982).

cut they fail to maintain position (Partridge and Pitcher 1980). This suggests that information from both the eye and the lateral line is utilized when fish school. Vision is most important for maintaining the attraction and when a fish swims too far from its neighbours it moves closer to nearby fish; the lateral line is most responsible for repulsion. By the lateral line, a fish can sense whether it is too close to a neighbour and must move away to avoid collision. We can conclude that vision is the more important sense for maintaining distance from, and angle to, the nearest neighbour and the lateral line appears to be most important for determining the neighbour's speed and direction.

A model of schooling based on self-organization

After we know that both vision and the lateral line are coupling modes of schooling, the fundamental problem facing a fish in a school is how to optimize its ability to respond rapidly to its neighbour's actions, to reduce its risk of colliding with a neighbour or being captured by a predator. To understand this, a number of researchers have developed models to help to explain this behaviour.

The following rules are quite simple, but they can show how fishes incorporate behavioural and sensory capabilities as they move within a school. The main features and assumptions include (Camazine et al. 2003):

1. Each fish in the school follows the same behavioural rules so the school has no leaders.

2. Each fish responds to its neighbours to avoid collision.

3. To decide where to move, each fish uses some form of weighted average of the position and orientation of its nearest neighbours.

A group of researchers from Tokyo University (Sannomiya et al. 1990) expressed an aggregated school model for the behaviour of fish school with many individuals. Another two groups from Japan (Hattori et al. 1999, OBOSHI et al. 2003) presented models of the interaction between a predator and prey and studied the responsive behaviour of fish groups induced by attacks of a predator under various situations. Recent mathematic models of school behaviour have been proposed. Takagi et al. (2004) and Adioui et al. (2003) applied a model-based on Newton's equation of motion and evaluated the model with parameter M (quantity of information exchange) which is the number of neighbours that affect an individual's behaviour. The results suggested the model can describe the behaviour when M is greater than 2 and less than 10, and small a value for M induced a behavioural pattern that reflected low cooperation.

3.2 Similarities between a school of fish and a team of UUVs

There are several similarities between a fish school and a team of UUVs. We can regard a fish as a small simple UUV with intelligent control and self-organization abilities.

First, the complex and unstable underwater environment is the biggest problem that fish and UUVs face. Different fish species have the ability to adaptively live in different environments, such as shallow water, deep ocean, etc. Likewise, different underwater missions require UUVs to cruise in different underwater environments. UUVs must learn the abilities of fish to ensure stability in their different environments.

Furthermore, the locomotion of fish and UUV are similar. Unlike the terrestrial environment, fish and UUV swim in a multi-dimensional environment. In the vertical plane, fish use their 'swim bladder' organ to control buoyancy. The swim bladder fills or releases gas to increase or decrease the buoyancy. The sets of muscles on side of the backbone alternately contract and form S-shape curves. When the back fin changes to curve shape, backward force is created. The fish move forward by this backward force. UUVs usually use a pressure sensor to detect water pressure information and consequently the depth of the vehicle.

Finally, UUVs have similar sensors for detection and communication sensors like fish. Fish rely on vision and the lateral line to detect the neighbour and obstacles. The current technologies, sonar and vision sensors, are used on UUVs by researchers and acoustic communication systems are used for exchanging the information between vehicles. The details of communication mode are further described in the following section.

3.3 Communication mode of animal and UUVs

Communication is the most important section in the whole process of group collective behaviour. We can classify two different typical communication modes (coupling mode) of animals. One is explicit communication that information exchange between animals. Ants communicate with each other through the use of pheromones, chemical substances that attract other ants. As the ants move, they lay down a trail of these pheromones that other ants can follow. Honey bees communicate the location of food by a symbolic 'waggle dance' language.

Figure 3.3 shows the waggle dance which tells other bees the direction, distance and quality of honey. The other typical communication mode is implicit communication which is only based on local information without information exchange. Birds flocking

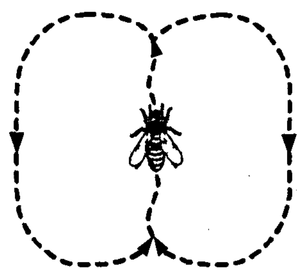


Figure 3.3: Honeybee's waggle dance

rely on vision and fish schooling rely on vision and the lateral line. Their behaviour only depends upon the local information which is detected by their special 'sensors'.

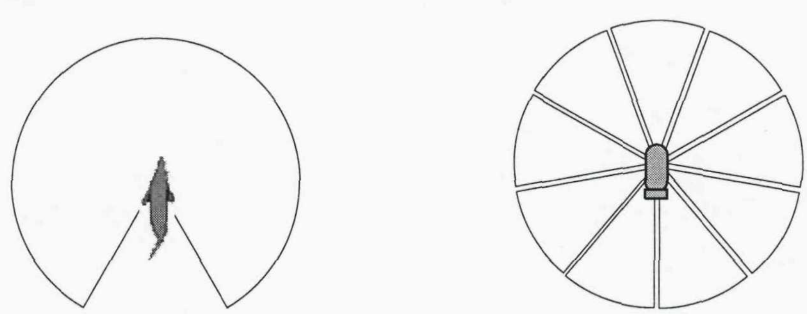
	Communiation mode	**Size**	**Distance between two neighbours**	**Forward speed**	**Response time to the signal**
Birds	vision	0.5m	0.5 - 1m	4.5 - 17.5m/s	15ms
Fish	vision	0.5 - 1.2m	0.5 - 1.2m	1.67 - 2.22m/s	20ms
	lateral line				
UUVs	sonar	1m	1m	2 - 4m/s	3.5ms - 1s
	underwater acoustic communication				5ms

Table 3.2: Comparison of communication modes between animals and UUV's

Since the underwater environment is more complicated than terrestrial environment, the communication mode of fish schooling is worthy of research for potential applications and for underwater vehicles. From section 3.1.4, we know that fish vision and the lateral line are the organs for maintaining the attraction and repulsion. By both organs fish can maintaining a certain distance with neighbours and avoid collision with neighbours and obstacles. In order to avoid collisions between each other in the collective process, the sensor must have enough speed to detect the information change and enable evasive action to take place.

The response time of fish vision and the lateral line includes data reception by the sensors, data processing by the brain and the time of transmitting control signals to the corresponding muscles. From Table 3.2, the response time of fish is about 20 milliseconds which is quick enough to detect information changes and adjust their motion (Iwai and Hisada 1998). Underwater sonar has a minimal response time of 3ms and underwater acoustic communication has 5 milliseconds response time (Bechaz and Thomas 2002). Compared to the animals' length and the sense organs response time, engineering has developed sensors of sufficiently high speed to be able to satisfy the requirements for group missions.

Fish vision perception covers a 300 degree spherical angle that extends out to a radius (typically 1-1.5 metres) defined by the waters translucence (Tu and Terzopoulos 1984) (Figure 3.4(a)).



(a) Fish vision sensor covers a 300° angle.

(b) UUV's multi-sensors covers 360°.

Figure 3.4: Range comparison between fish view and a UUV's multi-sensors

Acoustic Sonar is a strong directional sensor like fish vision and multi-sonar can cover a 360 degree angle (Figure 3.4(b)). From this point of review, multi-sonar can do better than fish vision. Explicit communication, which is based only upon local information, is good for detecting nearby obstacles and distance to neighbours.

There are some researchers who have already successfully built a single UUV that is able to navigate through an unknown environment, build maps, and act within the environment without outside intervention (Coiras et al. 2005).

Sometimes, missions require that each vehicle must be able to coordinate their operation by exchanging configuration, location and movement information. Underwater GPS is a kind of 'implicit communication' which is specially designed for exchanging information. Bechaz and Thomas (2002) provide a solution which is named "GIB Portable Tracking System". The GIB (GPS Intelligent Buoy) portable tracking system is based on a network of surface buoys that measures the travel time of acoustic signals emitted by a pinger mounted on the UUV. As shown in Figure 3.5, Buoys, like mobile stations in a mobile phone network, transmit signals to the GPS satellite and the signals go through the reverse path to the UUV.

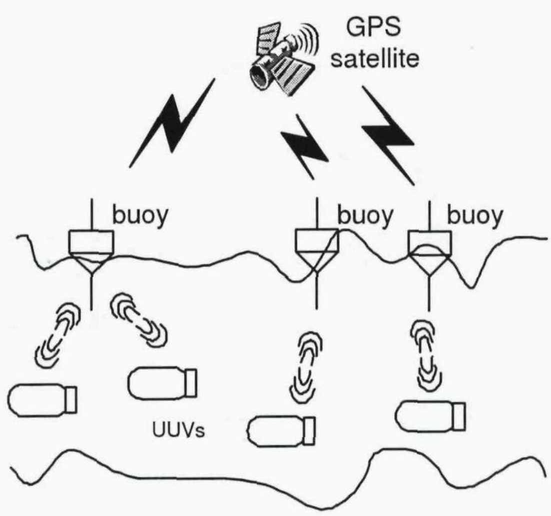


Figure 3.5: GIB System Principles

However, underwater GPS communication has a major disadvantage that GPS must be based on exact maps which means that underwater vehicles only with GPS cannot detect obstacles if the map is inaccurate. Therefore, underwater acoustic communications is much better for communicating between vehicles which both have the transmitter and receiver. Two researchers from Portugal presented two teams of UUVs which must coordinate their motions (Sousa and Pereira 2003). The first team of vehicles act as buoys to provide a Local Positioning System service to the second team. Unlike terrestrial communications, underwater communication is much more difficult to implement due to complicated and unpredictable underwater environment and is a significant barrier to UUVs working as a team. In chapter 8, we will compare various sensors and communication network strategies in the cooperation missions.

3.4 Conclusion

In this chapter, from some animal behaviours, such as ants trail formation, bees nectar source selection and fish schooling, we have the basic idea of collective behaviour and realize that we could learn from this to design a behaviour-based algorithm for co-operative control. In addition, from the similarities in locomotion and communication modes between grouping fish and cooperating UUV's, and since they operate in the same underwater environment, we can further conclude that animal collective behaviour, in particular fish schooling, can potentially be applied for controlling a team of cooperative underwater unmanned vehicles.

Chapter 4

Priority Weighted Behaviour-Based Method for Team Cooperation and Implementation

An accurate model represents a real-world object at a high level of fidelity. The corresponding real-world object being modeled here is the natural cooperative team behaviour seen in bees, fish and birds. From these natural systems one can derive a common basic behaviour for a cooperative team. This chapter introduces the basic **Boids**[1] ideas and extends this to a weighted behaviour-based rules method. We simulate a simple multiple vehicle cooperative mission to assess this method in practice.

4.1 Basic Boids ideas

In 1986, Craig Reynolds (Reynolds 1987) developed a computer model of coordinated animal motion such as bird flocks and fish schools. It was based on three-dimensional computational geometry of the sort normally used in computer animation or computer aided design. They called the generic simulated flocking creatures **Boids**. Each object

[1]**Boids** is the method and **boid** is a single object in a group.

in the group is called a **boid**. The basic flocking model consists of three simple steering behaviours (Figure 4.1) which describe how an individual boid manoeuvres based on the positions and velocities of its nearby flockmates.

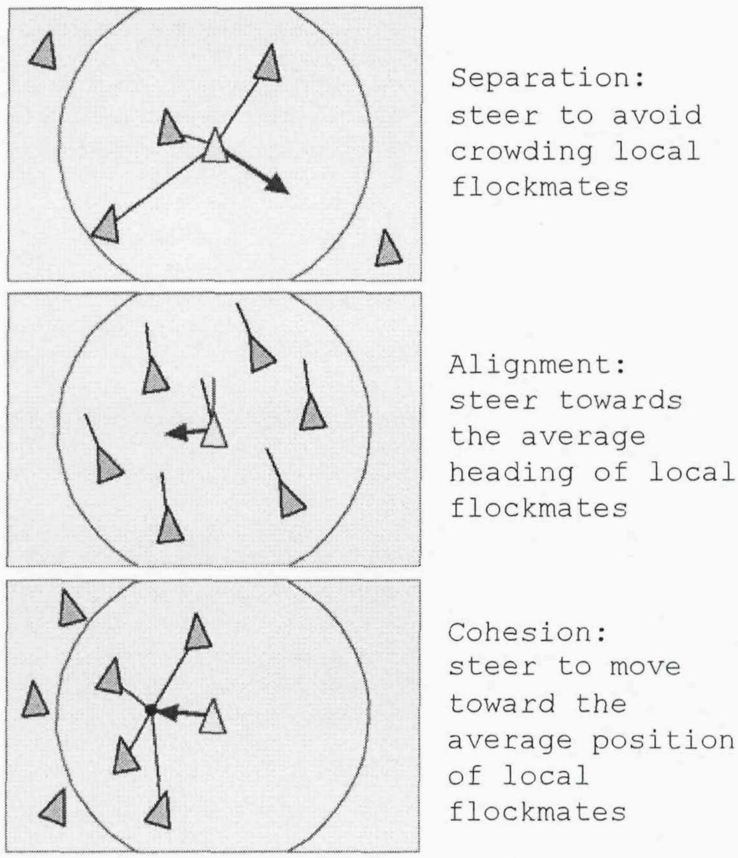


Figure 4.1: Three simple steering behaviours used in the Boids model

The collective behaviours were expressed as:

- The top subfigure in Figure 4.1 Collision avoidance: The central triangle boid attempts to avoid collisions with neighbours or obstacles. This behaviour provides the ability for a boid to keep a distance from the neighbours and keep away from the obstacles.

- The middle subfigure in Figure 4.1 Velocity Matching: The central triangle boid attempts to match velocity (speed and direction) with neighbours. This behaviour provides the ability for a boid to go with the average direction of other teammates within the group.

- The bottom subfigure in Figure 4.1 Flock centering: The central triangle boid attempts to stay close to the average position of local neighbours. This behaviour provides the ability for a boid to remain within the group.

Each boid reacts only to flockmates within a certain neighbourhood around itself. As shown in Figure 4.2, the neighbourhood is characterized by a distance (measured from the centre of the boid) and an angle, measured from the boid's direction of flight. The angle is usually less than 360° because the detection sensors of animals do not operate over a full 360° range. It defines the region in which flockmates influence a boids steering.

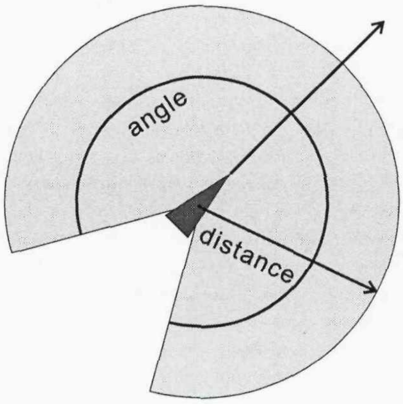


Figure 4.2: A boid's neighbourhood area

Each boid has a central processor to control the set of behaviours and to store internal states, such as the position and the velocity. It is easiest to encapsulate these into an object, in the sense of object-oriented programming. Reynolds (Reynolds 1987) used these three distributed behaviours shown in Figure 4.1 that were used in the simulation.

If a boid matches velocity with its neighbours, then it will probably avoid collisions. So static collision avoidance tends to establish minimum distances between boids and velocity matching tends to maintain it.

Each of the three behavioural rules is expressed as an velocity request. The requests are weighted according to priority and then averaged to give a final velocity vector.

Inspired from the **Boids**, a vehicle can be regarded as a **Boids** central processor which controls the speed and heading direction. From **Boids** rules, we summarize several behaviour-based rules to satisfy the requirement of simulation for a team cooperative vehicles. Combining and coordinating multiple behaviours gives a complex relationship between a robotic system and the real world. The global structure is represented with behaviours in parallel and outputs are channeled into a coordination function that produces an appropriate response (Figure 4.3). The coordination function assembles all the active behaviour responses and generates a final vector response. We assign the priority weight for each behaviour-rule. The final decision is judged by the priority weights.

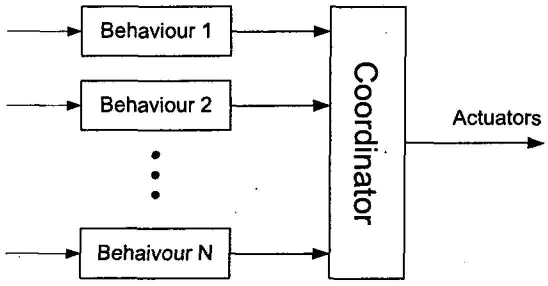


Figure 4.3: Behaviour-based structure

4.2 The behaviour-based rules controlled by constant priority weights

In this project, the algorithm which is used in the simulation is inspired from **Boids** rules but has several new behaviour rules 3, 5 and 6 added.

- Rule 1: going towards the centre of the group
- Rule 2: keeping matching velocity
- Rule 3: navigating to the target
- Rule 4: avoiding collision with teammates
- Rule 5: avoiding obstacles
- Rule 6: remaining within a certain area

However, these rules are modified to satisfy the demand of different operational scenarios. It is not difficult to find potential problems when velocity is calculated by the **Boids** behaviour rules. Different rules must have different priority requirements. For example, an obstacle avoidance rule must have a higher priority than a target navigation rule. Otherwise the vehicle could hit an obstacle due to a low priority rule. How to decide priority weight for each behaviour rule is a problem we need to deal with. We classify the behaviour rules to 3 different priority levels. A higher priority rule has a higher priority weight value assigned to give much more effect to the next step velocity decision. Each behavioural rule i is applied for generating a steering angle factor β_i to build the next step desired steering command. In the following sections, the behavioural rules are described and the steering angle factors are formulated respectively. The next step desired steering angle is calculated by the combination of all steering angle factors multiplied with relative priority weights. We also must ensure that the angle β_i is in the proper quadrant. The priority weight is the most important factor which affects the intelligent decision. An intelligent vehicle, like a fish, decides the priority of the current rule according to the situation that it meets and relevantly assigns a weight value to the rule. We focus on the behaviour of a single vehicle in the group. Other vehicles within the group individually do the same thing. For simplification, we only consider 2D motion of the vehicle in the horizontal plane in this project.

4.2.1 Rule 1: going towards the centre of the group

This rule makes all vehicles move towards the centre of the entire group. As a result, the vehicles will not leave the team. However, one vehicle cannot possibly detect the position of all vehicles within the group due to the sensor range limits. For simplification, the range of sensor detection is defined as a disc of a certain radius as shown in Figure 4.4.

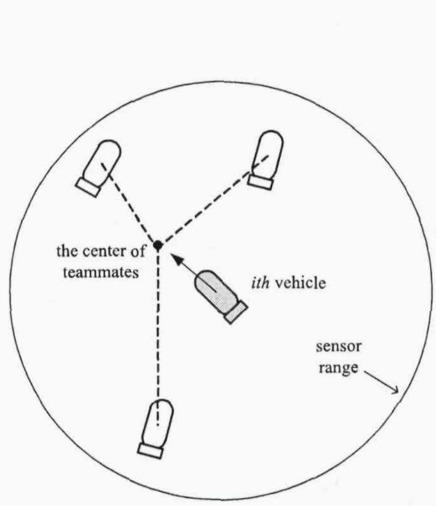


Figure 4.4: Rule 1: going towards the centre of the group

The steering angle factor is the direction angle towards the average position of teammates as the following equations describe.

$$px_{ave} = \frac{px_1 + \cdots + px_{i-1} + px_{i+1} + \cdots + px_N}{N-1} \tag{4.2.1}$$

$$py_{ave} = \frac{py_1 + \cdots + py_{i-1} + py_{i+1} + \cdots + py_N}{N-1} \tag{4.2.2}$$

$$\beta_1 = \arctan\left(\frac{py_{ave}}{px_{ave}}\right) \tag{4.2.3}$$

where β_1 is the steering angle factor calculated from rule 1, (px, py) is the position of each of the teammates in the reference frame of coordinates, N is the number of the teamates, i is the ith teammate. Denominator is $N-1$ because the ith component (such as the UUV of interest) is not included.

4.2.2 Rule 2: keeping matching heading direction

The speed of the vehicles is constant. This rule keeps vehicles' heading directions as similar as possible when they are manoeuvring. A vehicle receives information of its teammates and then calculates the steering angle factor β_2 by equation 4.2.6. As shown in Figure 4.5, the central vehicle intends to turn β_2 degree to keep matching the heading directions of teammates within the sensor range.

$$vx_{ave} = \frac{vx_1 + \cdots + vx_{i-1} + vx_{i+1} + \cdots + vx_N}{N-1} \tag{4.2.4}$$

$$vy_{ave} = \frac{vy_1 + \cdots + vy_{i-1} + vy_{i+1} + \cdots + vy_N}{N-1} \tag{4.2.5}$$

$$\beta_2 = \arctan\left(\frac{vy_{ave}}{vx_{ave}}\right) \tag{4.2.6}$$

where β_2 is the steering angle factor calculated from rule 2, (vx, vy) is the velocity of each vehicle in the reference frame of coordinates.

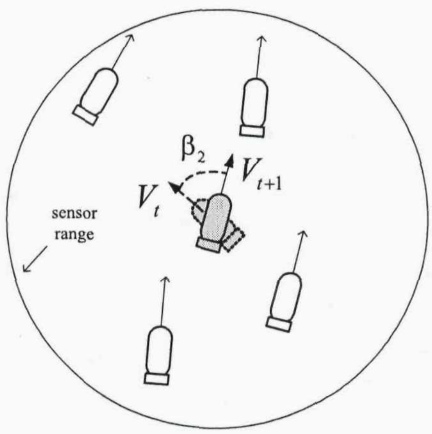


Figure 4.5: Rule 2: keeping a matching velocity

4.2.3 Rule 3: navigating to the target

Some guidance systems for underwater water vehicles has been well developed: way-point guidance by line of sight (LOS), vision-based guidance, Lyapunov-based guidance and electro-magnetic guidance. Among these guidance laws, LOS is one of the most widely used guidance strategies for UUVs due to its ease of implementation (Naeem et al. 2003). It has been applied for single UUV navigation missions (Belkhouche et al. 2006) (Wu et al. 2006). The line of sight guidance law is based on the geometry of the interception scenario. The system can be described in a relative system of coordinates as shown in Figure 4.6 which shows a idealized case since in practice the sensors maybe position in the nose of a vehicle. The UUV and the target are shown respectively. The positions for Target and UUV in the reference frame of coordinates are given by the vector (x_t, y_t) and the vector (x_U, y_U). The notations in Figure 4.6 represent that LOS_{TU} is the line of sight UUV-Target and θ_{TU} is the line of sight angle.

In a waterflow-free environment, the line of sight angle θ_{TU} to the target is written as equation 4.2.7.

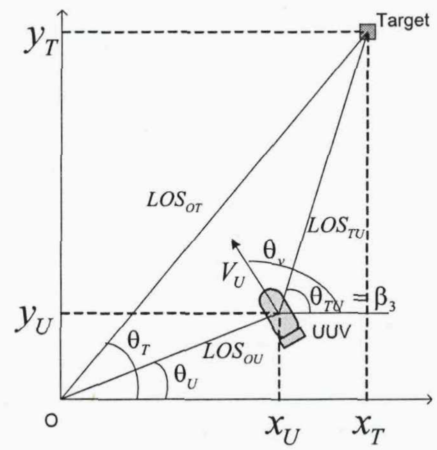


Figure 4.6: Rule 3: navigating to target

$$\theta_{TU} = \arctan\left(\frac{y_t - y_U}{x_t - x_U}\right) \tag{4.2.7}$$

The steering angle factor β_3 is equal to θ_{TU}:

$$\beta_3 = \theta_{TU} \tag{4.2.8}$$

4.2.4 Rule 4: avoiding collision with teammates

Rule 4 ensures that vehicles do not hit each other when they are too close. In Figure 4.7, L_{mt} is the minimum tolerable distance between two vehicles. When a vehicle detects there is one teammate within the intolerable distance, it will try to steer away from the position of the teammate. Sometimes, there are two teammates within the dangerous intolerable distance, in which case the vehicle will turn more away from the teammate which is the nearest to itself. From Figure 4.7, the vehicle on the right is the nearest to the central vehicle and so the central vehicle will steer away from that vehicle.

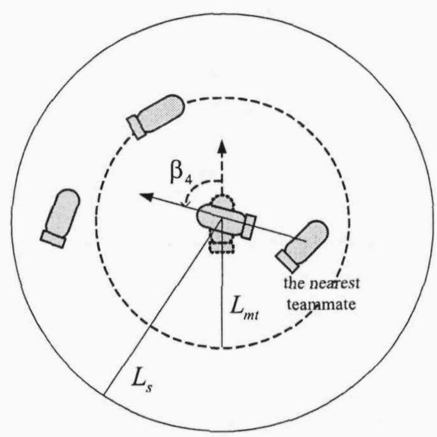


Figure 4.7: Rule 4: avoiding collision with teammates. The dotted vehicle denotes the current orientation of the central vehicle. The overlapped solid vehicle denotes the desired orientation of the central vehicle.

The steering angle factor β_4 can be written as:

$$\beta_4 = -\arctan\left(\frac{py_{nearest} - py_{current}}{px_{nearest} - px_{current}}\right) \tag{4.2.9}$$

where β_4 is the steering angle factor calculated from the rule 4 and $(px_{nearest}, py_{nearest})$ is the coordinate of the position of teammate which is the nearest to the current vehicle.

4.2.5 Rule 5: avoiding obstacles

Rule 5 produces similar behaviour to rule 4. We assume that the shape of obstacles are cycloidal. As shown in Figure 4.8, the vehicle will steer away from the obstacles when the distance L_o between vehicle and obstacle is smaller than the minimum tolerable distance L_{mo}. The desired direction is directly opposite to the centre of obstacle. The vehicle finds the quickest way to steer to the desired direction. In Figure 4.8, the action of steering right will spend less time to the desired direction than the action of steering left since angle β_5 is smaller than angle β_5'. When the obstacle is directly ahead, the vehicle chooses either left or right to steer away from the obstacle. The steering angle factor β_5 is defined by equation 4.2.10.

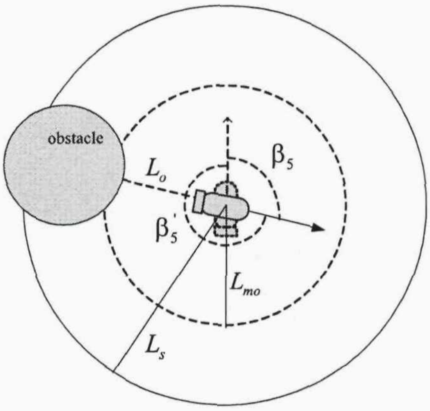


Figure 4.8: Rule 5: avoiding obstacles

$$\beta_5 = -\arctan\left(\frac{py_{obstacle} - py_{current}}{px_{obstacle} - px_{current}}\right) \tag{4.2.10}$$

where β_5 is the steering angle factor calculated from rule 5 and $(px_{obstacle}, py_{obstacle})$ is the coordinate of the position of an obstacle. In practise it is difficult to know the centre position of an cycloidal obstacle, sometimes, an irregular shape obstacle. The L_o is the minimum distance that the sensor can detect to the obstacle whatever the shape of the obstacle is.

4.2.6 Rule 6: remaining in certain area

In some mission scenarios, the vehicles are required to remain in a specified area. For example, vehicles only search a certain area for the source of pollution. The smaller the area searched, the less the energy consumed. We can look upon an area boundary as an obstacle and the desired direction is the opposite of the boundary. As shown in Figure 4.9, the L_{mb} is the minimum tolerable distance between vehicles and a boundary. The equation 4.2.11 generates the steering angle factor β_6.

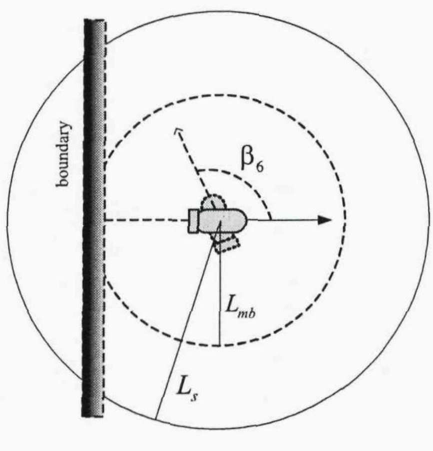


Figure 4.9: Rule 6: remaining in an area

$$\beta_6 = -\arctan\left(\frac{py_{boundary} - py_{current}}{px_{boundary} - px_{current}}\right) \tag{4.2.11}$$

where β_6 is the factor calculated from rule 6 and $(px_{boundary}, py_{boundary})$ is the position of an area boundary which we have defined.

4.3 Priority weight classification for behaviour-based rules

Weight is the most important factor that affects intelligent decision making. Intelligent vehicles, like fish, must decide the priority of a current rule and assign a relevant weight value to the rule when it needs to make the next step steering decision.

Basically, we classify the six behaviour rules into three classes as shown in Figure 4.10. The highest priority class 1 includes rules 4, 5 and 6 which may cause the damage to the vehicle. The lowest priority class 3 includes rules 1 and 2 which have a less serious role. Between the lowest and the highest class, rule 3 is lower than the highest priority class because it would not lead to any vehicle damage, but it is higher than the lowest priority class because it can reduce the travel time to the target.

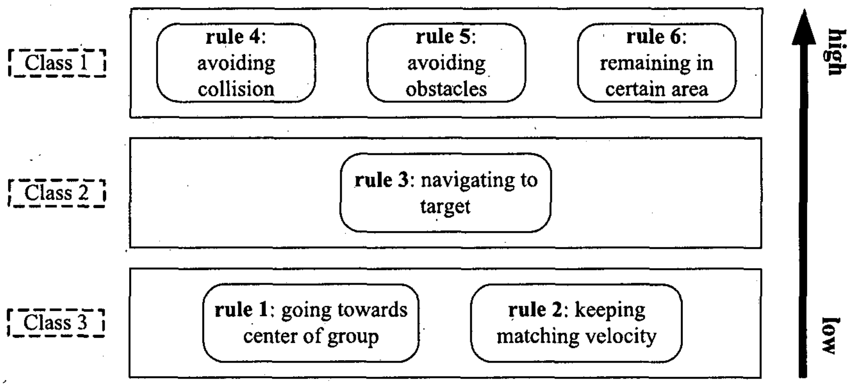


Figure 4.10: Three priority classes for the six behaviour rules

According to the hierarchy in Figure 4.10, we define weights $w_i \in 1, 2, \cdots, 6 (i = 1, 2, \cdots, 6)$ for the six rules. The highest priority rule will have the highest weight $(w = 6)$. The steering angle factors from equation 4.2.3 to equation 4.2.11 are multiplied by their defined weight values and are combined to get the final desired steering angle.

$$\beta_{desired} = \arctan\left(\frac{\sum_{i=1}^{6} w_i \sin\beta_i}{\sum_{i=1}^{6} w_i \cos\beta_i}\right) \tag{4.3.1}$$

where $\beta_{desired}$ is the desired next step steering angle and w_i is weight value of ith rule.

4.4 Dynamic UUV model and simulation environments

This section introduces the SUBZERO III vehicle and the dynamic physical model of vehicle used in the simulation.

4.4.1 SUBZERO III

SUBZERO vehicle is a low cost underwater flight vehicle (Figure 4.11) which was originally built by Dawson (Dawson et al. 1992) and Lea (Lea et al. 1999) and upgraded by Feng (Feng et al. 2003) at Southampton University . The UUV is a torpedo-shaped flight vehicle of 1m long and maximal 10cm in diameter and neutrally buoyant.

The vehicle includes the following subsystems(Feng et al. 2003):

- Two micro controllers (MCUs):Motorola 16 bit processor with 2x16 bit 8 channel A to D converters
- Power supply: Motor battery pack of 9.6V(8x1.2V) NiMH battery pack (3000mAH). Electronics battery pack of 6V(5x1.2V) NiMH battery pack (2700mAH).

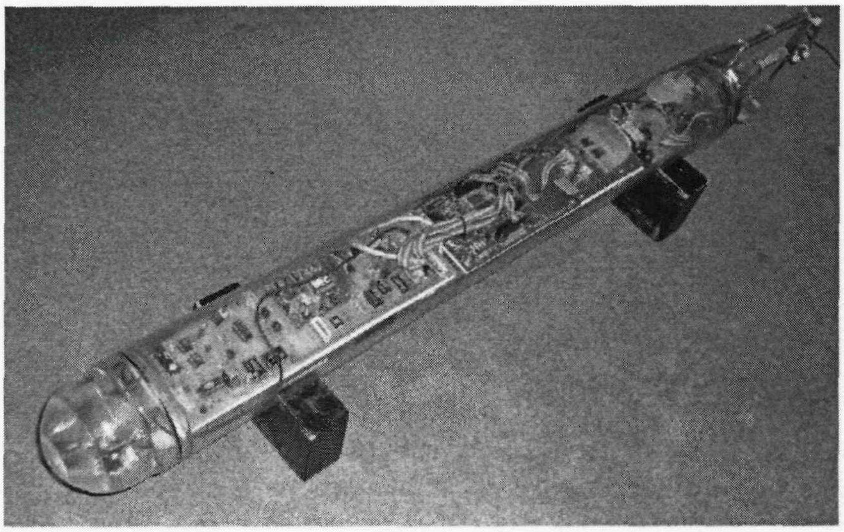


Figure 4.11: SUBZERO III

- Mechanical design: 5mm acrylic hull and servomotors which are for control of stern-plane and rudders for diving and steering.
- Communications and control: the communications between the vehicle and the control computer is through the COM port of a laptop computer. A joystick is used for controlling the vehicle.

The basic data of the vehicle is shown in Table 4.4.1.

Table 4.1: The basic data of the vehicle

Mass	**Length**	**Diameter**	**Weight**	**Cruise speed**
m=7kg	l=0.98m	d=15cm	W=69.0N	u=1.3m/s

4.4.2 Physical model of dynamic manoeuvering

As an agent in a group, each vehicle must automatically operate without any external human control. In order to make vehicles 'think' for themselves, a central control function will be programmed into the main CPU.

In the simulation design, we use the following parameters for each vehicle.

- For simplification, the forward speed of vehicle $V_{ave} = 1.5m/s$. It is faster than real cruise speed.
- The angular change velocity $\omega = \pi/5rad/s$.
- The motion updating frequency is 10Hz.

In the meantime, we assume that the vehicles can collect the adequate local information by sensors. In practice, sonar, underwater acoustic equipments and underwater GPS, will be assembled on the vehicles. This will be further discussed in the future work.

The dynamic model of manoeuvering control includes speed and direction control at each time step. The next step velocity ($V_{nextstep}$) is calculated from the current velocity ($V_{current}$) and the desired velocity ($V_{desired}$) at each updating time point. The velocity includes speed(S) and direction(θ).

The next step velocity $V_{nextstep}$ is computed via the dynamic model of the vehicle. The desired updating velocity $V_{desired}$ is computed via the bio-inspired rule-base. The desired velocity $V_{desired}$ for keeping the group together or avoiding obstacles is judged at each updating time point. The time gap between two desired velocity updating points is longer than the time gap of next step velocity $V_{nextstep}$ updating because the sensors must spend time collecting the information and the processor spends sometime

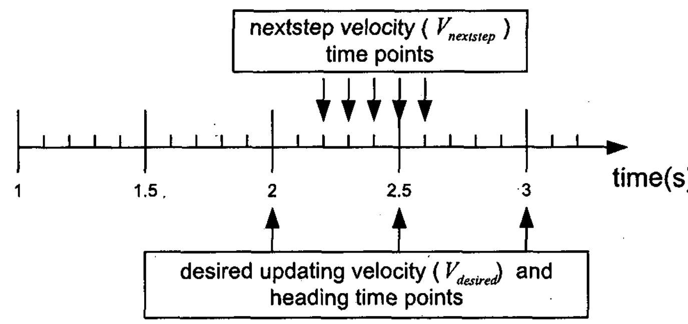


Figure 4.12: Velocity calculation at time scale. The desired velocity $V_{desired}$ updating frequency is 2Hz. The next step velocity $V_{nextstep}$ updating frequency is 10Hz.

implementing the algorithms. High speed sensors and processors will help vehicles to make the next step decisions much quicker in future implementations. Here $V_{desired}$ is updated at each 0.5 second and $V_{nextstep}$ is updated at each 0.1 second as shown in Figure 4.12.

Motion analysis of an underwater vehicle involves six degrees of freedom (DOF), as six independent coordinates are required to determine the position and orientation of a rigid body in three dimensions. The first three coordinates and their time derivatives represent the translational position and velocity while the last three describe the rotational angle and angular velocity.

When considering a dynamic model of an underwater vehicle, it is convenient to use two coordinate systems: a global (earth-fixed) coordinate system or frame XYZ and a body-fixed system $X_0Y_0Z_0$ as shown in Figure 4.13. In terms of vehicle position and motion, the earth-fixed system is the frame of interest whereas the equations describing the vehicle's behaviour are more easily developed in the body-fixed system (Lea et al. 1999). The six DOF vehicle model is decoupled into two reduced dynamical systems: a depth-pitch model that considers motion in the vertical plane and a plane-yaw model that studies the motion in the horizontal plane.

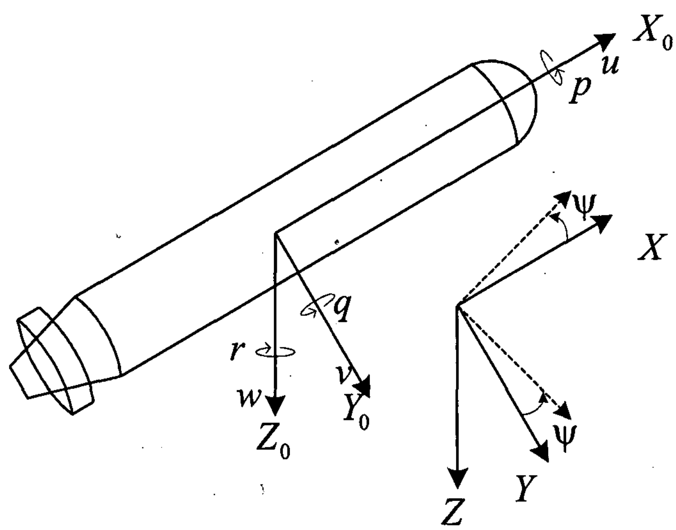


Figure 4.13: Coordinate systems of an underwater vehicle

In this project, we only consider the 2D motion of the vehicles in the horizontal plane. The kinematic equations of motion for an UUV on the horizontal X-Y plane can be written as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ r \end{bmatrix} \tag{4.4.1}$$

where x and y represent the coordinates of the position of the vehicle and $\boldsymbol{u}$ and $\boldsymbol{v}$ are the surge (forward) and sway (side) velocities, respectively, defined in the body-fixed frame. The orientation of the vehicle is described by the angle ψ measured from the X axis and $\boldsymbol{r}$ is its yaw (angular) velocity (Repoulias and Papadopoulos 2007).

In order to graph the velocity on the coordinate plane, the velocity is represented by Cartesian coordinate (x, y). In order to ensure the velocity is mapped on the correct quadrant, Figure A.1 and procedure 3 in Appendix A describe how to map the velocity with Cartesian coordinate on polar coordinates system.

4.4.3 Dynamic model: steering control procedure

In our simulations, the speed of vehicles is constant. The positions of vehicles are only updated by changing heading directions. To combine the dynamic manoeuvering model with the team cooperation algorithm in the simulation, the model is formulated by procedure 1. The vehicle steers step by step to the current desired velocity direction until the next desired velocity direction is calculated. θ_{temp} is the temporary variable to save the potential next step steering direction. $\theta_{current}$ is the current steering direction. $\theta_{nextstep}$ is the next step steering direction. $\theta_{desired}$ is the desired updating steering direction. ω is the angular change velocity which is equal to yaw velocity r in Figure 4.13. When the time is increasing, the vehicle steering direction is changing until it reaches the desired updating steering direction.

Procedure 1 Dynamic model: steering control

repeat
  $\theta_{temp} = \theta_{current} + \omega \times t$
  if θ_{temp} has not reached $\theta_{desired}$ **then**
    $\theta_{nextstep} = \theta_{temp}$
  end if
until $\theta_{temp} = \theta_{desired}$

The vehicle without steering dynamics changes its heading direction instantly. In this case, the vehicle heading direction directly steers to the new desired updating steering direction as soon as it is been calculated.

In rule 5 avoiding obstacles, we have described that the vehicle will choose either left or right to steer away the obstacle. Procedure 2 shows how the vehicle choose left or right in order to choose the quickest direction to reach the desired updating angle. Figure 4.14 shows an example that the vehicle needs to turn right when $\theta_{desired}$ is on the right side of $\theta_{current}$.

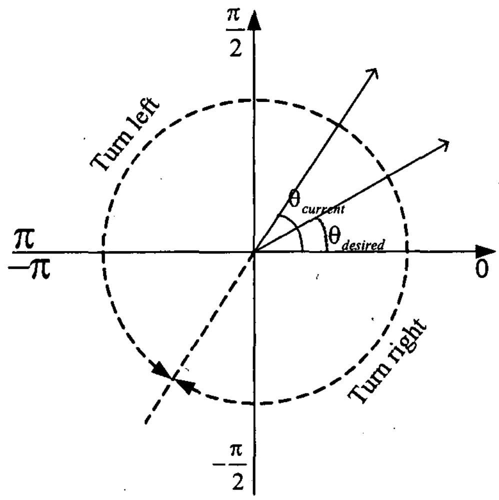


Figure 4.14: Steering control decision

The mathematical procedure for steering control varies when the current velocity direction of the vehicle lies in a different quadrant. The Figure B.1(a) and procedure 4 in Appendix B describe the details of the steering control procedure when the current velocity direction of the vehicle is in 1st or 2nd quadrant. When current velocity direction of the vehicle is in 3rd or 4th quadrant, figure B.1(b) and procedure 5 in Appendix B are used for steering direction control.

Procedure 2 Steering direction control decision

if $\theta_{desired}$ is on the right side of $\theta_{current}$ **then**
  repeat
    Vehicle turns right
  until $\theta_{current} = \theta_{desired}$
end if
if $\theta_{desired}$ is on the left side of $\theta_{current}$ **then**
  repeat
    Vehicle turns left
  until $\theta_{current} = \theta_{desired}$
end if

4.5 Simulations and Results

In order to assess the feasibility of the approach in applications of UUVs teams, we not only define a simple team cooperation mission scenario to simulate but also include a simple dynamic manoeuvring model of the vehicle (in this case SUBZERO III). To simulate underwater vehicles, the underwater communication environment must be considered. The real vehicle sensors, underwater communications channels and navigation/positioning estimates are, in practice, corresponded by noise. However, accurate information is required to ensure the algorithm works properly in practice. In order to focus on assessing the central algorithm in simulation, we make the following assumptions to clarify that all required information are valid. The position of the start point and initial velocity are set by operator before the mission starts. The sensor range of detecting neighbouring vehicles is predefined. The position and velocity vectors of the neighbouring vehicles can be updated in real-time. The position of obstacles and targets can be measured.

The work flow chart in Figure 4.15 describes a single control loop of a vehicle in the team. At the beginning, the vehicles are initialized with positions and velocities. The vehicles then continually update their motions to steer to the desired steering angle. At each updating time point, the vehicles calculate the desired steering angle using the behaviour-based rules. Each behaviour-based rule r_i generates a steering angle factor β_i which is multiplied the relative priority weights and then combined with the relative steering angle factors from the other rules to generate the final desired steering angle $\beta_{desired}$. According to the velocity calculation time scale in Figure 4.12, the vehicle computes the desired updating velocity at each desired updating velocity time point. When the time points are not desired updating velocity time points, the vehicles tend to steer to the desired angle $\beta_{desired}$ and get the new positions via the dynamic model of the vehicle. The mission ends as soon as the vehicles reach their goal.

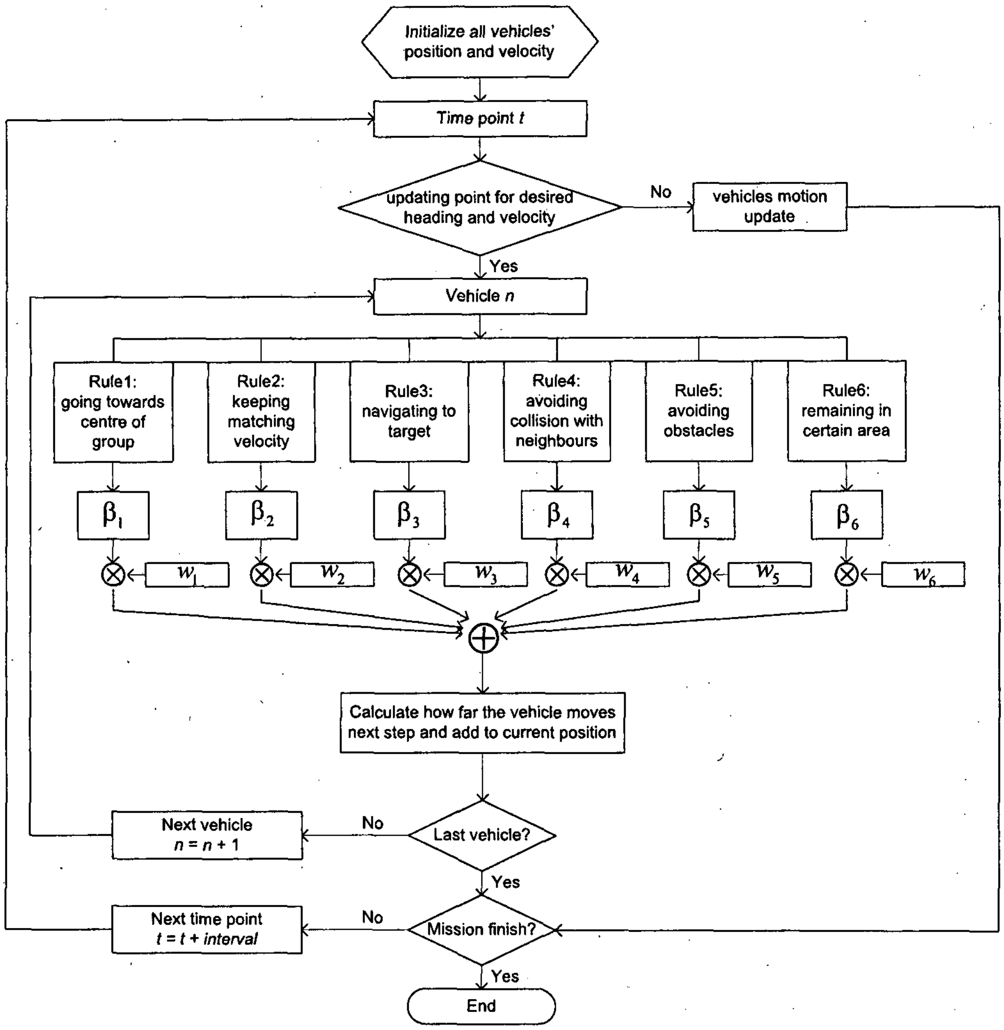


Figure 4.15: The flow diagram of real-time team cooperation control in the simulation

4.5.1 Simulation initialization

The simulation mission is simply designed to explain how the algorithm works and to assess performance. The area is limited to a $100 \times 100 m^2$ square pool. A group of vehicles depart from the starting line and then fly cooperatively and avoid obstacles until they reach the target. There are several parameters which need to be initialized and Table 4.2 lists all parameters and their values.

Table 4.2: Parameters defined in the simulation

Parameters	Descriptions	Value	Unit
V	the forward speed of a vehicle	1.5	m/s
ω	the rotational speed of a vehicle	$\pi/5$	rad/s
$F_{updating}$	the desired velocity update frequency	5	Hz
L_{mt}	the minimum tolerable distance between two vehicles	4	m
L_{mo}	the minimum tolerable distance between vehicle and obstacles	5	m
L_{mb}	the minimum tolerable distance between vehicle and group boundary	4	m
$P_{initial}$	The coordinates of the initial position of vehicles	(20,0) (25,0) (30,0) (35,0) (40,0)	m
$P_{obstacles}$	The coordinates of the centre and radius of obstacles	(45,30,5) (68,58,8) (66,31,6)	m

According to the different priorities of rules, we define $w = [1, 1, 3, 6, 6, 6]$ as weight values for rule 1 to rule 6. Rules 4, 5 and 6 have the maximal weight value 6. Rules 1 and 2 have the minimal weight value 1 and rule 3 has a mid-weight value of 3 (see Figure 4.10).

4.5.2 Mission implementation results

Figure 4.16 shows 12 snapshots during the scenario simulation. Each snapshot plots the vehicles' status at one of 12 time points to explain how the algorithm is working.

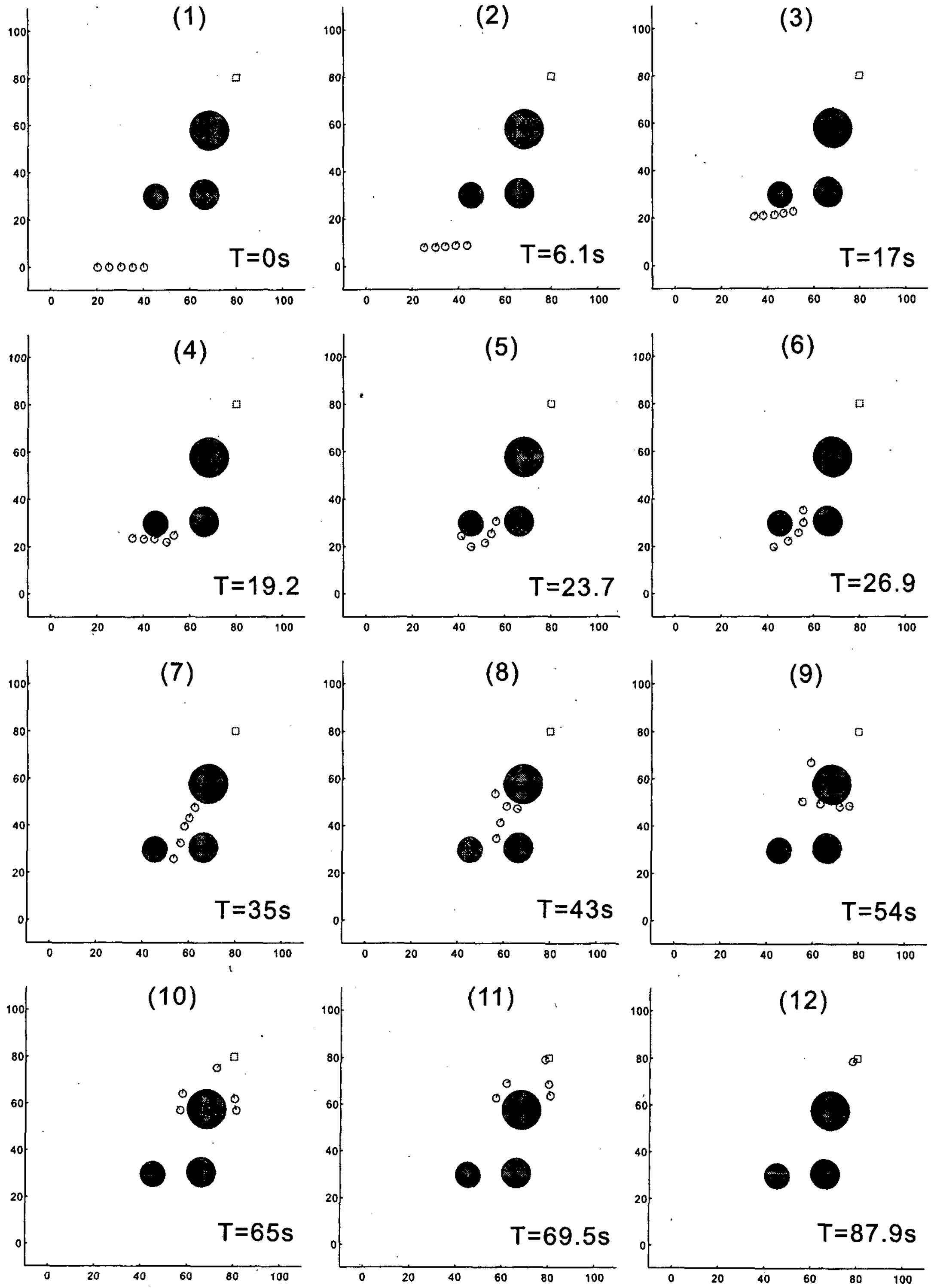


Figure 4.16: 12 snapshots of scenario simulation

Snapshot 1 Five vehicles depart from their initial positions.

Snapshot 2 They group together and move towards the target.

Snapshot 3 The vehicles encounter an obstacle within their the minimum intolerable distance to an obstacle.

Snapshot 4 They decide to turn right in order to avoid the obstacle.

Snapshot 5 The first vehicle on the right side detects another obstacle on the way.

Snapshot 6 Vehicles found a way between two obstacles to make sure they avoid both obstacles and remain as a team.

Snapshot 7 The last obstacle is located by the first vehicle.

Snapshot 8 The first vehicle turned left to avoid the obstacle but the second vehicle turned right. The way how to choose left or right to steer is described in section 4.4.3. However, this causes the vehicles split up. The solution of avoiding splitting by using circle formation control is addressed in Chapter 7.

Snapshot 9 The other vehicles rotate to avoid the obstacle.

Snapshot 10 All vehicles pass the obstacle, regroup and move to the target.

Snapshot 11 The first vehicle reaches the target.

Snapshot 12 The last vehicle reaches the final target and the mission ends.

From Figure 4.16, we can see that the weights lead to safe decisions to control the group to complete the mission. In the next step, we will set different weights to investigate how weights affect the control of the vehicle.

4.5.3 Comparison between different weights setting

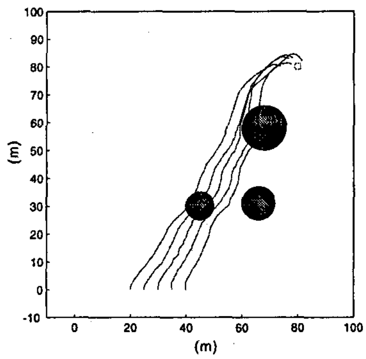
First of all, we give the same weight value to all rules and then give a low weight to the high priority rule and validate how priority weights affect decisions for the motion of the vehicles. The previous mission goal was to ensure all vehicle arrive the target. Here mission goal has changed to end as soon as one vehicle reaches the target because we only compare the procedure of collision and obstacle avoidance under different priority weight settings. From Figure 4.17(a) and Figure 4.17(b), we can clearly see that vehicles cross obstacles which means vehicles will collide with the obstacles although they remain in a group. Hence incorrect weights can place the vehicles in danger.

After this, the weight value for the navigating rule was adjusted to 1 and the trajectory is shown in Figure 4.17(c). Compared with the trajectory in Figure 4.17(d), we can see that the trajectory in Figure 4.17(c) is less direct to the target because the reduced weight value leads to a reduced effect in the velocity decision equation 4.3.1 and this is confirmed by the increased travel time (Table 4.3) to finish the mission.

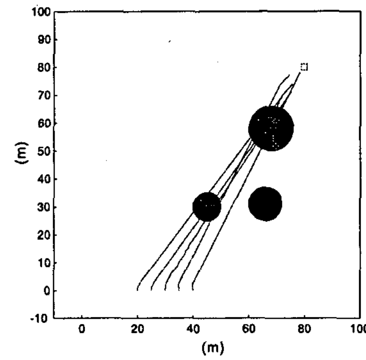
Table 4.3: Travel time comparison between different weight settings

weights	[1 1 1 6 6 6]	[1 1 3 6 6 6]
figure	4.17(c)	4.17(d)
travel time	84s	73s

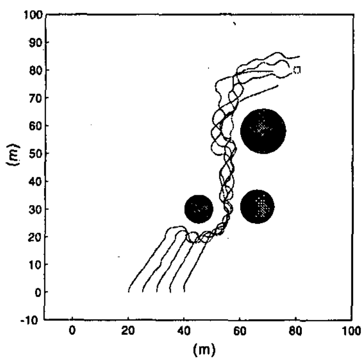
Since the simulation with weights [1 1 3 6 6 6] has the minimum travel time, we are using it in the following simulations in this section.



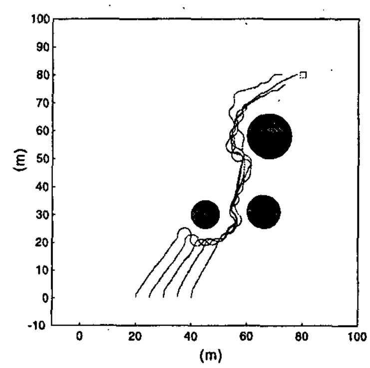
(a) Same weight values to all rules. Trajectories with weights (1 1 1 1 1 1).



(b) High weight values to low priority rules. Trajectories with weights (6 6 3 1 1 1)



(c) Reduce weight for navigating rule to 1. Trajectories with weights (1 1 1 6 6 6).



(d) Trajectories with weights (1 1 3 6 6 6)

Figure 4.17: Vehicles trajectories with different priority weight values

4.5.4 Relationship between group number and the minimum distance to obstacles

When the number of vehicles in a group is being increased, the chance of crashing each other rises. We notice that increasing the minimum tolerable distance to an obstacle may solve the problem. Therefore, groups of up to 12 vehicles are simulated to investigate the relationship between group number (N_v) and the minimum tolerable distance to obstacles (L_{mo}).

Figure 4.18 plots the results of 6 simulations with vehicle numbers from 3 to 12 in which the minimum tolerable distance was reduced until collision occurred. From this figure, it is clear that the minimum tolerable distance to obstacles must increase when the number of vehicles increase. Larger distance to obstacles gives more time so that vehicles in a group have more space in which to find a way to avoid a collision. However, the relationship depends not only on the number of vehicles and the minimum tolerable distance to obstacles, but also the initial position of the vehicles, vehicle speed, the position of the obstacles and the location of the destination. Therefore, the relationship in Figure 4.18 can only be used in this simulation. It cannot be used as an estimation of the minimum tolerable distance to obstacles when designing for different situations. It merely gives us a clue that we must increase the distance to obstacles when increasing the number of vehicles and exercise care in the design stage, using simulation where possible to assess the consequences of changes in these design parameters.

From these results, we can conclude that these behaivour-based rules can be used for updating vehicles' motion in team cooperation. The priority weights specify which rules are the most important emergent behaviour to emphasize when vehicles need to make a decision of the next step in their mission trajectory. However, the weights in

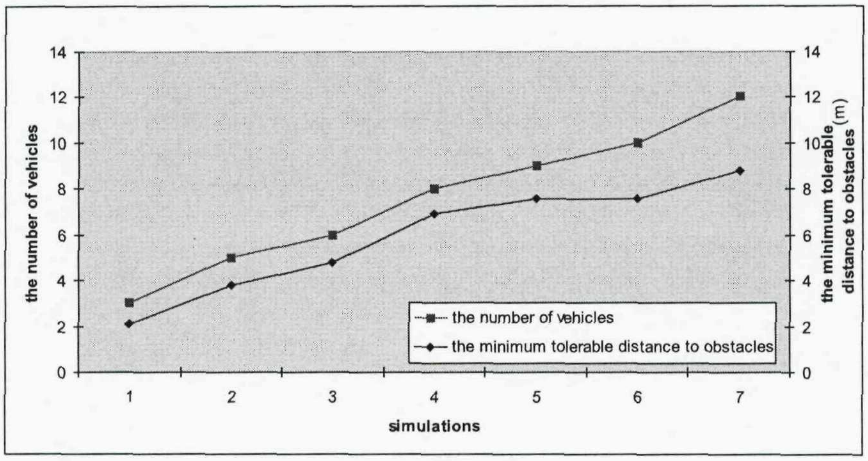


Figure 4.18: Relationship between number of vehicles in a group and the minimum distance to obstacles

the simulation were given manually and their constant weight values were not adapted to meet the changes in the complex environment that the vehicles meet. This may be a reason why the trajectories of the vehicles are not always the quickest. In the next section, we use a fuzzy logic method to estimate the weight values for rules online. Weights will be estimated by a fuzzy logic controller according to the situations that the vehicles meet. The different results between adaptive and constant weight schemes will be compared.

4.6 Conclusions

This chapter has described a set of behaviour-based rules for satisfying a simulation scenario. From the simulation results, we can conclude that these behaviour-based rule with constant priority weight values is a possible solution for multi-UUV applications. Simulation results with different priority weight values indicate that the priority weight is a fatal effect factor for decision of the next step velocity of a vehicle. In the next chapter, we will apply a fuzzy logic method to estimate the priority weights in real-time according to the situation that vehicles meet to replace the constant weights in this chapter.

Chapter 5

Fuzzy Logic Controlled Priority Weighted Behaviour-Based Method

5.1 The behaviour-based rules controlled by fuzzy logic priority weights

This chapter uses the Fuzzy Logic method to measure the priority weights for each behaviour-based rule on-line. Instead of using constant weights, the fuzzy logic method updates the weights value according to the situations that vehicles meet in real time.

Fuzzy logic is a method based in computational intelligence. It has been widely used in modern industrial and consumer product control systems. It is aimed at modelling the imprecise models of reasoning, such as common sense reasoning, for uncertain and complex processes (Kruse et al. 1994). Bajec used the fuzzy logic method as the basis of the Boid's decision making in computer animations simulations (Bajec et al. 2003). In this chapter we use the fuzzy logic method to estimate the weight values for the high priority rules according to the situation that the vehicles meet and to adapt the weight values for low priority rules according to the values given to the high priority rules. The decision of the next step steering direction is then calculated by velocity factors multiplied by relative priority weight values.

For the sake of simplicity, triangular shaped of the membership functions were selected. In general, the designer only has to decide the three points in the triangular function (Lin and Chang 1996).

5.2 Adjustment of behaviour rules by fuzzy logic and priority weights

In the previous chapter 4 we described the rules without fuzzy logic control. Weight values are decided uniquely as long as the vehicles implement their rules. For example, obstacle avoidance is referred to the highest priority rule and the weight value of this rule is given as 6 when vehicles locate an obstacle within danger zone. However, the weight value of this rule is fixed and cannot be adapted when the vehicles get closer to, or further from an obstacle. The generation process of the priority weight value for each rule is shown in Figure 5.1.

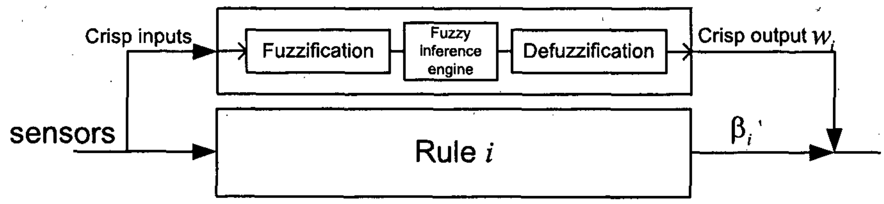


Figure 5.1: The generation process of the fuzzy logic controlled priority weights

In the previous chapter, there are six behaviour rules. In this chapter, we modify rules 4, 5 and 6 which are designed for avoiding danger. The priority weights of rules 1, 2 and 3 are calculated by the fuzzy logic method.

5.2.1 The fuzzy modification of rule 4: avoiding collision with teammates

In Figure 5.2, L_{mt} is the minimum tolerable distance between two vehicles. When a vehicle detects that there is one teammate within the danger zone, it will try to steer away from the position of the teammate. Sometimes, there are two teammates within the danger zone, in which case the vehicle will turn more away from the teammate which is the nearest to itself. From Figure 5.2, the vehicle on the right is the nearest to the central vehicle and so the central vehicle will steer away from the vehicle on the right.

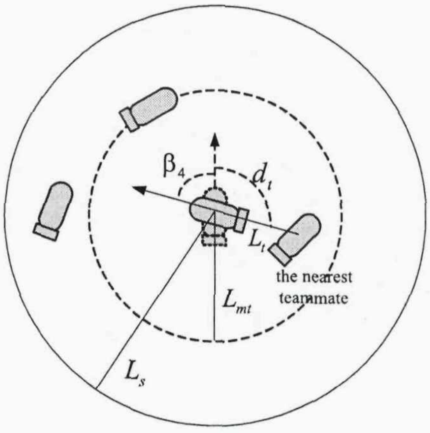


Figure 5.2: Rule 4: avoiding collision with teammates

To make a weight value adaptive, we use a fuzzy logic controller to compute the weight value in real time. Two linguistic input variables L_t, d_t are declared. In Figure 5.2, L_t denotes the distance to the nearest teammates. It has two linguistic values *short*

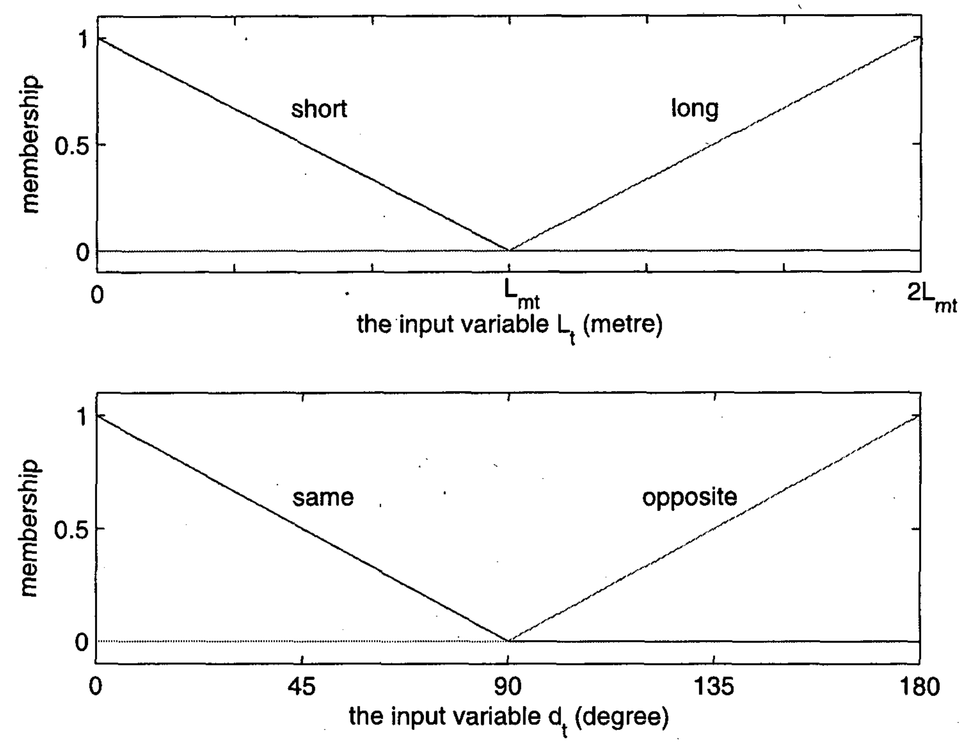


Figure 5.3: The membership function of two linguistic input variables L_t and d_t

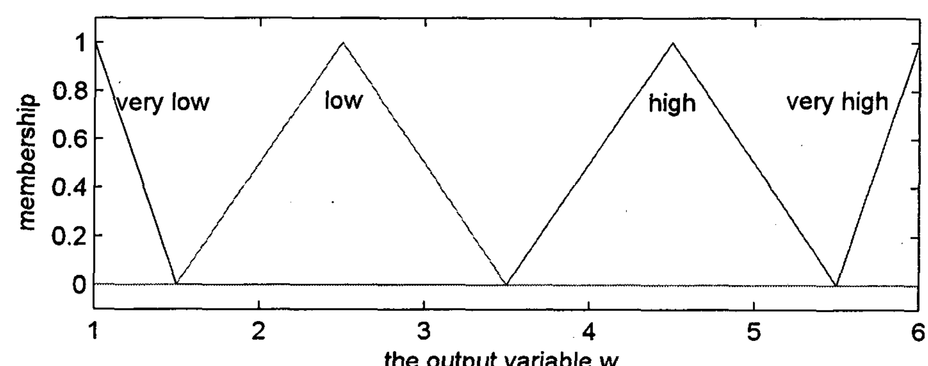


Figure 5.4: The membership function of the linguistic output variable w_4

and *long* (Figure 5.3). When $L_t \leq L_{mt}$, the vehicle is close to the teammate and the value of L_t is *short*; when $L_{mt} \leq L_t \leq 2L_{mt}$, the vehicle is far away from the teammate and the value of L_t is *long*. d_t denotes the angle difference between the current velocity direction and the direction along the current position towards the closest teammate. It has two linguistic values *same* and *opposite* (Figure 5.3). When $0 \leq d_t \leq \frac{\pi}{2}$, the vehicle is facing to the teammate and the value of d_t should be *same*; when $\frac{\pi}{2} \leq d_t \leq \pi$, the teammate is on the rear side of the vehicle and the value of d_t should be *opposite*. w_4 is the output variable declared with four linguistic values, *very low*, *low*, *high* and *very high* (Figure 5.4). According to the class of the priority rules in Figure 4.10, the value *very high* represents the highest weight value and its range is given as [5.5 6]; the value *high* and the value *low* represent values in the range [3.5 5.5] and [1.5 3.5] respectively; the value *very low* represents the lowest weight value and its range is given as [1 1.5].

The following fuzzy rules represent the function to decide the weight.

IF L_t *is short* and d_t *is same* THEN w_4 *is very high*;
IF L_t *is short* and d_t *is opposite* THEN w_4 *is high*;
IF L_t *is long* and d_t *is same* THEN w_4 *is high*;
IF L_t *is long* and d_t *is opposite* THEN w_4 *is low*.

The defuzzification of the data into a crisp output is accomplished by combining the results of the inference process and then computing the 'fuzzy centroid' of the area. The basic concept of centroid method is that center of mass of the result provides the crisp ouput value. The Figure 5.5 shows an example of centroid method. The following equation 5.2.1 used to calculate the crisp output.

$$z^* = \frac{\int \mu(z) \cdot dz}{\int \mu(z) dz} \tag{5.2.1}$$

where $\int$ denotes an algebraic integration.

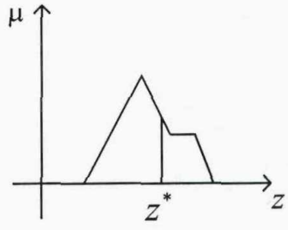


Figure 5.5: An example of centroid method

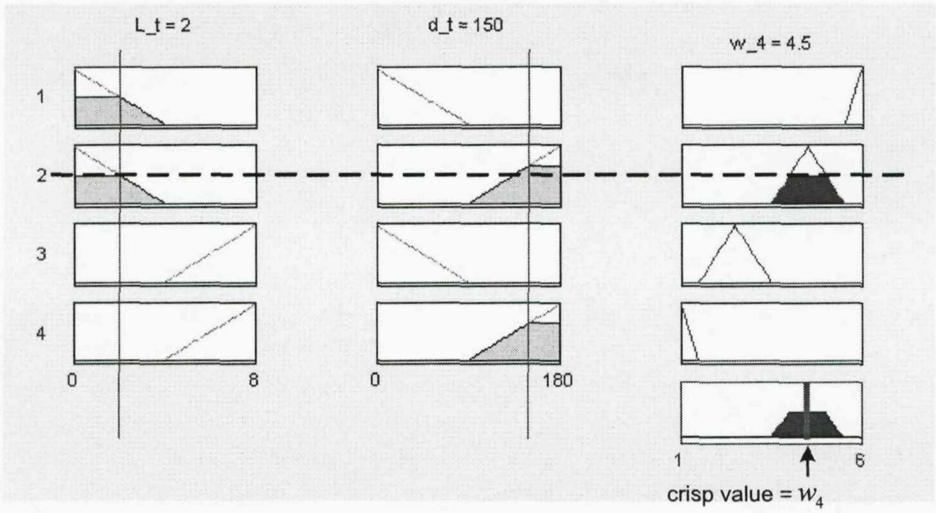


Figure 5.6: An example of w_4 calculation when $L_t = 2$ and $d_t = 150$

The Figure 5.6 shows the defuzzification process of the fuzzy logic controller F_4. In this example, w_4 is calculated by equation 5.2.2.

$$\begin{aligned} w_4 &= \frac{\int_{3.5}^{4}(x-3.5)xdx + \int_{4}^{5}0.5xdx + \int_{5}^{5.5}(-x+5.5)xdx}{\int_{3.5}^{4}(x-3.5)dx + \int_{4}^{5}0.5dx + \int_{5}^{5.5}(-x+5.5)dx} \\ &= \frac{0.4792+2.25+0.6458}{0.125+0.5+0.125} \\ &= 4.5 \end{aligned} \tag{5.2.2}$$

5.2.2 The fuzzy modification of rule 5: avoiding obstacles

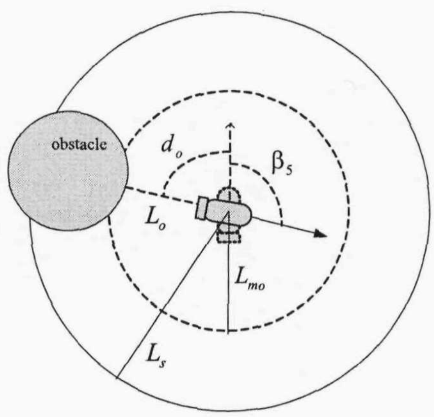


Figure 5.7: Fuzzy rule 5: avoiding obstacles. The dotted vehicle denotes the current orientation of the central vehicle. The overlapped solid vehicle denotes the desired orientation of the central vehicle.

Similarly, the weight values for rule 5 are adaptive controlled by the fuzzy logic controller.

Two linguistic input variables L_o and d_o are declared. In Figure 5.7, L_o is the minimum distance to the nearest obstacle boundary. It has two linguistic values *short* and *long*. The value *short* is given for L_o of between $[0\ L_{mo}]$. L_{mo} is the minimum tolerable distance between a vehicle and an obstacle. The value *long* is given when L_o is between $[L_{mo}\ 2L_{mo}]$. d_o is the angle difference between the current velocity direction and the direction from the position towards the nearest obstacle. It has two linguistic values *same* and *opposite*. The range of the value *same* is given as $[0\ \frac{\pi}{2}]$; the range of the value *opposite* is given as $[\frac{\pi}{2}\ \pi]$. The membership functions of the two input variables are shown in Figure 5.8. w_5 is the output variable declared with four linguistic values, *very low*, *low*, *high* and *very high*. The range of these values are the same as the description of w_4 in rule 4 and hence the membership functions are the same as those in Figure 5.4.

The following fuzzy rules represent the function to decide the weight value.

IF L_o *is short* and d_o *is same* THEN w_5 *is very high*;
IF L_o *is short* and d_o *is opposite* THEN w_5 *is high*;
IF L_o *is long* and d_o *is same* THEN w_5 *is high*;
IF L_o *is long* and d_o *is opposite* THEN w_5 *is low*;

5.2.3 Fuzzy rule 6: remaining in certain area

Since the algorithm of this rule is same as the algorithm for obstacle avoidance, the fuzzy logic controller design is the same except that the minimum tolerable distance L_{mb} is defined as between a vehicle and a team boundary. The value *short* of the input variable L_b is given as $[0\ L_{mb}]$. The value *long* is given as $[L_{mb}\ 2L_{mb}]$. The membership functions of the input variables and the output variable are similar to the membership functions in the Figures 5.3 and 5.4.

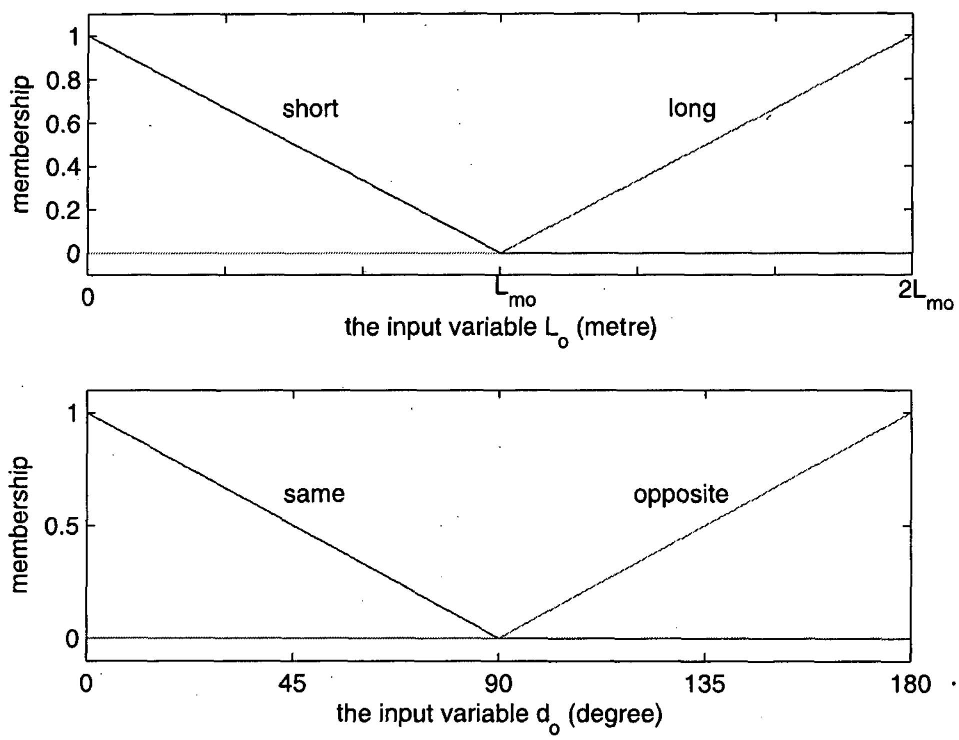


Figure 5.8: The membership function of the linguistic variable inputs L_o and d_o

5.3 Fuzzy priority weight values for low priority rules

The weight values of low priority rules are controlled by the weight values of the high priority rules. When the vehicles are in a high priority emergency situation, the weight values of high priority rules must be very high in order to ensure safety. When the vehicles do not meet any safety problem, the weight values of low priority rules increase to higher values than the weight values of high priority rules. This function allows the weights of low priority rules to adaptively change on-line to give increased priority to navigation and cooperation when collision is not a problem.

We declare three linguistic input variables with weight 4, weight 5 and weight 6 with 4 linguistic values *very low*, *low*, *high* and *very high*. The range of each value is the same as those in fuzzy rule 4 as is the membership function of the variable weight in Figure 5.4. As output, weights 1, 2 and 3 are declared as linguistic variables.

The following fuzzy rules represent the function.

IF *weight 4 is very high* or *weight 5 is very high* or *weight 6 is very high*
THEN *weight 1 is very low*, *weight 2 is very low*, *weight 3 is very low*;
IF *weight 4 is very low* and *weight 5 is very low* and *weight 6 is very low*
THEN *weight 1 is low*, *weight 2 is low*, *weight 3 is high*;
IF *weight 4 is high* or *weight 5 is high* or *weight 6 is high*
THEN *weight 1 is very low*, *weight 2 is very low*, *weight 3 is very low*;
IF *weight 4 is low* and *weight 5 is low* and *weight 6 is low*
THEN *weight 1 is high*, *weight 2 is high*, *weight 3 is high*;

5.4 Results and Discussion

We have described the behaviour-rules with weights controlled by the fuzzy logic method and how multiple vehicles operate within the system. The results of simulation are presented here to provide an indication of team performance.

Figure 5.9 shows the flow diagram of the adaptive system. Compared to the flow diagram Figure 4.15 in section 4.2.4, the fuzzy logic weights controller F_4,F_5 and F_6 in the dotted boxes are used to generate the weight values w_4, w_5 and w_6.

The parameters used in the simulations are those defined in Table 4.2. The minimum tolerable distance L_{mt}, L_{mo} and L_{mb} are used in the membership function design of the fuzzy weight controllers.

5.4.1 12 snapshots of a simulation to a target

First of all, we need to see whether the system is working for a typical mission scenario. Rather than showing the trajectory of the vehicles, we plot here 12 snapshots in figure 5.10 which can explain how the vehicles move towards the target position.

At time 0.1s, all vehicles depart from their initial positions. Before time 17.0s, all vehicles maintained a group and advanced to the target. At time 17.0s, the vehicle on the right hand side detected an obstacle in front. It then decided to turn right to avoid the obstacle. Other vehicles followed in this direction until time 22.9s. Then the vehicles detected another obstacle which posed a collision threat. Vehicles rotated to the left and found a safe path between two obstacles. At time 40.0s, the third obstacle was found and all vehicles turned away to avoid collision. At time 63.2s, all vehicles had successfully passed all obstacles. Finally, they reached the target at time

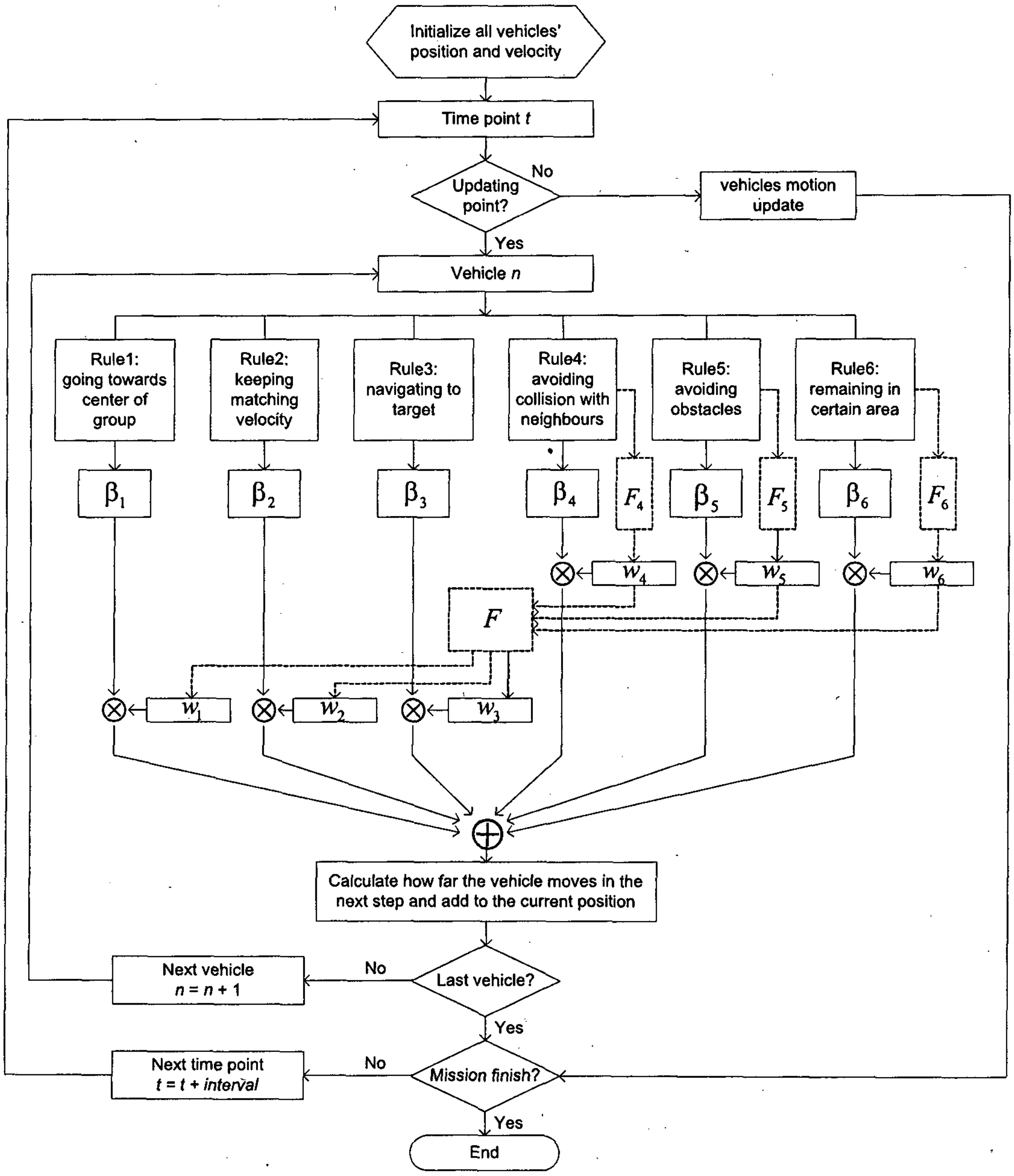


Figure 5.9: Adaptive system flow diagram

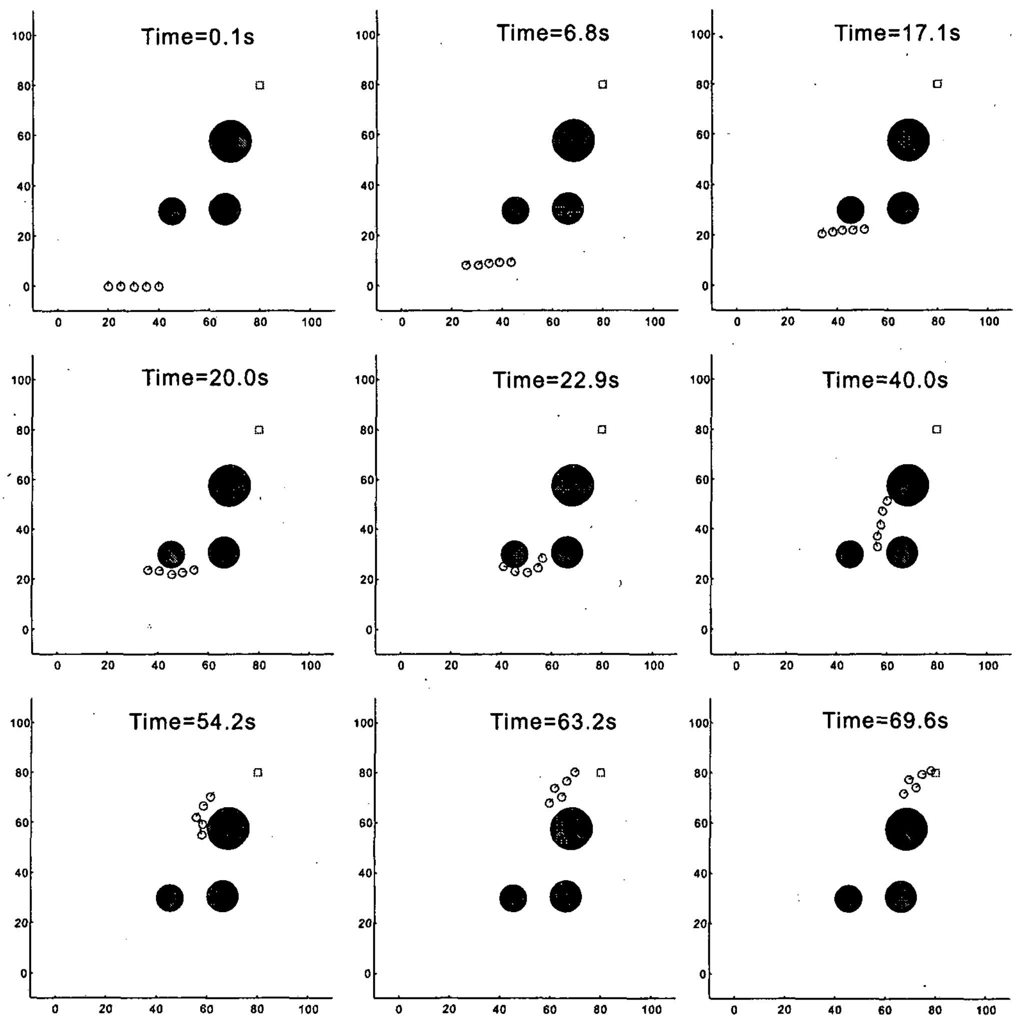


Figure 5.10: 9 snapshots at 9 time points

69.6s. All vehicles cooperated as a team throughout the entire trajectory. No one left the group or collided with a teammate.

Now let us have a look at how the weights update adaptively. We pick up vehicle 3 and plot the weights over the time of flight (Figure 5.11). From the plot of weight 5, we can see that the times of peaks are when the vehicle meets the obstacles. The Figure of weight 6 is empty because the vehicle has not been close to the team boundary. We can also see that the weights of rules 1,2 and 3 are low or very low when weights of rules 4,5 and 6 are high or very high; the weights of rules 1,2 and 3 are high and very high when weights of rules 4,5 and 6 are low and very low.

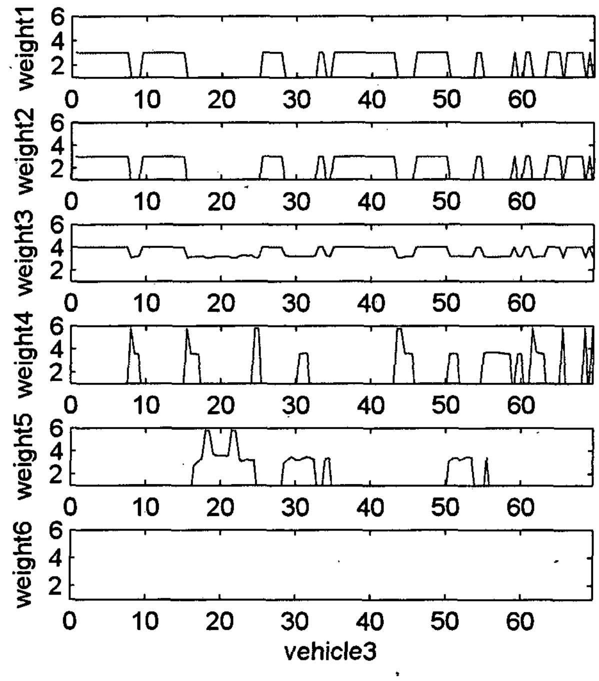


Figure 5.11: The weights of vehicle update over the time

Let us compare these results and those presented in the previous section 4.2.4. The travel time is 69.6 seconds when we used the fuzzy logic controller to adapt the weight values. The travel time in the simulation with constant weights is 73.0 seconds. Based on these simulations, we can expect that the fuzzy logic method has the ability to reduce travel time. In the real engineering system, this should benefit economic use of the onboard energy supply.

However, whether the mission can successfully conclude or not really depends upon what scenario is presented and what value of the parameters listed in the Table 4.2 are defined. For example, if the rotation speed is too slow, the vehicle may have collided on an obstacle before it rotates in avoidance.

Another key parameter is the desired heading direction updating frequency. A high frequency means that the vehicle has more chance to detect and avoid danger. Compared to the previous updating frequency of $F_{updating} = 10Hz$, we also simulated with $F_{updating} = 1Hz$ and the mission failed since a vehicle have collided with an obstacle. In terms of Figure 4.12, if $F_{updating}$ is slow, the vehicles would collide with obstacles before the new desired heading direction was calculated, and consequently the obstacle detection chance decreased. A higher updating frequency of the desired heading direction will lead to a reduced risk of a collision but to achieve a high frequency updating, a high speed processor and quick response sensors are required in practice.

In the current situation, decreasing the minimum tolerable distance L_{mt} between vehicles and the minimum tolerable distance L_{mo} between vehicles and obstacles maybe a possible solution to optimizing the travel time. Figure 5.12 shows the trajectories when different L_{mt} and L_{mo}. The following Table 5.1 lists the travel time with different parameters. From the Table, it only indicates that the travel time decreases when the minimum tolerable distance L_{mo} decreases because the vehicles are much closer to the line of sight to the target. The decreasing of L_{mt} also reduces the travel time because the smaller team size ensures that the vehicles do not separate to avoid the obstacle. The travel time when both are 4 metres is an exception which is longer than others. This is because one vehicle fled away from the team in order to avoid the last obstacle and took sometime to get back the team. As shown in Figure 5.12, when L_{mo} is shorter than 3 metres, the mission fails since the vehicles hit the obstacle. When L_{mt} is longer than 5 metres, the mission also failed. This is because rule 4 has much more effect on the next step decision than rule 5 when $L_{mt} \geq L_{mo}$ and this may allow the vehicles to collide with an obstacle. In practice, we may consider the configuration of these two parameters to minimize the travel time.

Table 5.1: The travel time results with different distance parameters. L_{mt} is the minimum tolerable distance between vehicles and L_{mo} is the minimum tolerable distance between vehicles and an obstacle. The travel time is the time when all vehicles arrive the target.

L_{mt}	L_{mo}	travel time
3m	3m	N/A (Mission failed.)
3m	4m	75.1s
4m	4m	80.5s
5m	4m	N/A (Mission failed.)
3m	5m	76.0s
4m	5m	78.3s
5m	5m	N/A (Mission failed.)

5.4.2 Comparison of simulations with and without a dynamic model of a vehicle

Dynamic models reflect the authenticity of the simulations since approaches with successful simulations within a dynamic environment have a greater chance of being implemented into real product designs. Simulations without dynamics are usually used to verify the approaches in computer graphic or in the ideal situation. However, although simulations with dynamics is still modelling and differs from reality, the aim

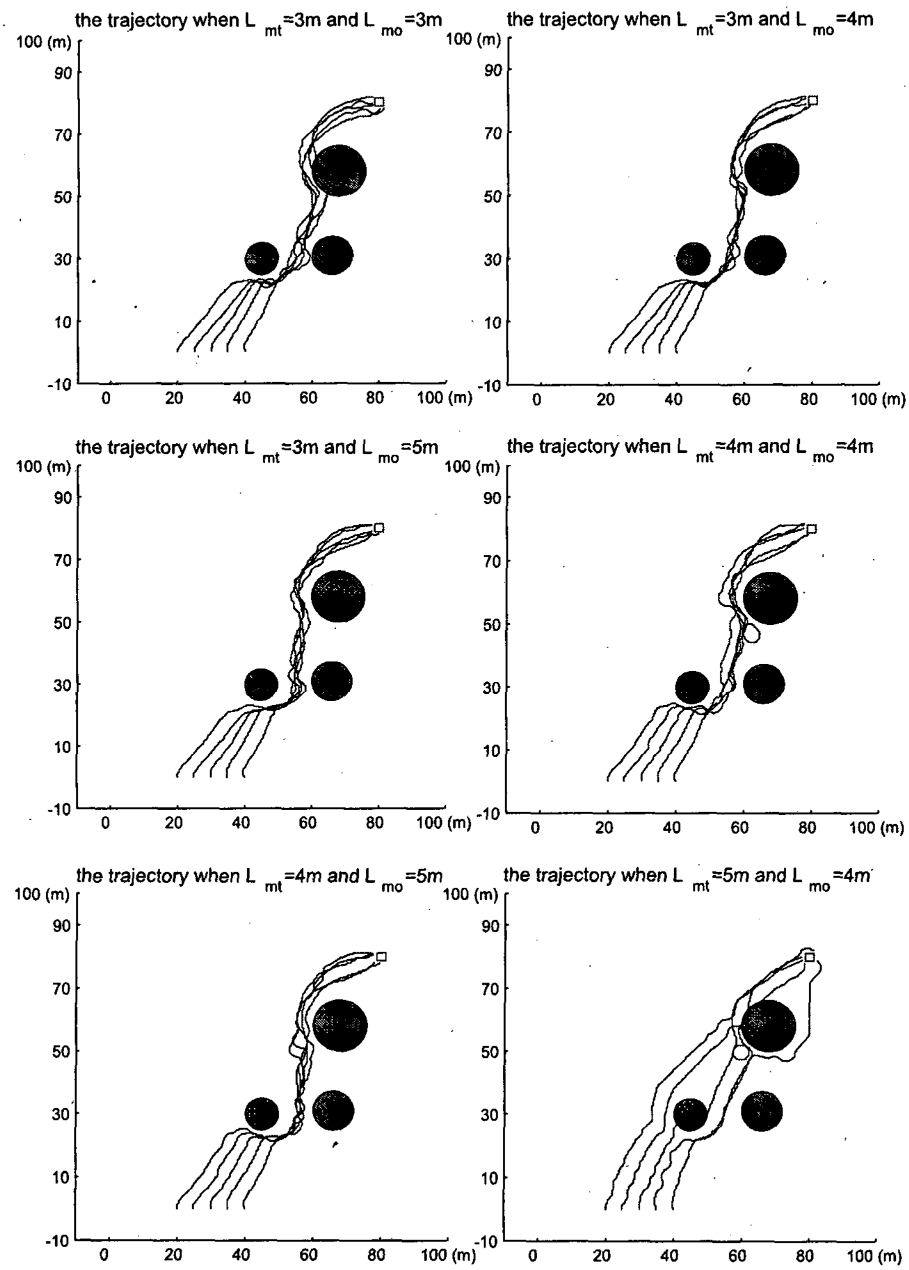
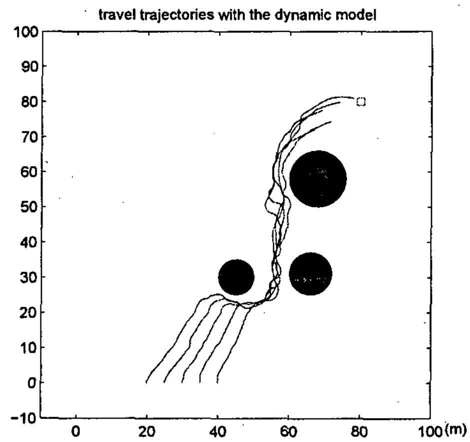


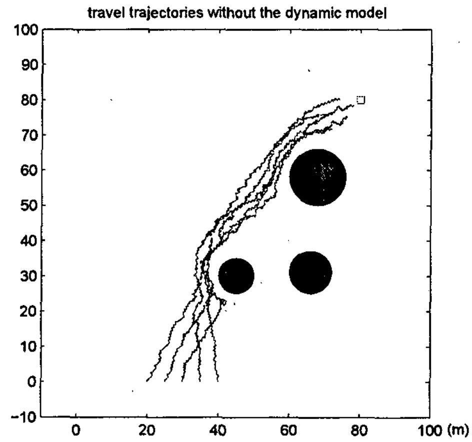
Figure 5.12: The trajectories with different L_{mt} and L_{mo}. L_{mt} is the minimum tolerable distance between the vehicles; L_{mo} is the minimum tolerable distance between the vehicles and an obstacle.

is to describe the dynamic model to be as close as to the real underwater vehicles as possible.

We respectively simulate missions with and without vehicle dynamics. The parameters in the simulations are same as those in Table 4.2. Figures 5.13(a) and 5.13(b) show the comparison of travel trajectories between simulations with and without a dynamic model of the SUBZERO vehicle. The trajectories in Figure 5.13(b) are much more unstable than the trajectories in Figure 5.13(a) because simulated vehicles without dynamics are always actuated to change their heading direction instantly based upon the decision from several behaviour rules. Simulated vehicles with dynamics must follow a dynamic change in heading direction. As a result, vehicles without dynamics spend a longer time (132.8s) than vehicles with dynamics (69.6s) at least in this simulation. The priority weighting scheme would have to reconsidered if vehicle dynamics were not significant. It is interesting that simulations of cooperative behaviour have largely ignored the effect of vehicle dynamics.



(a)



(b)

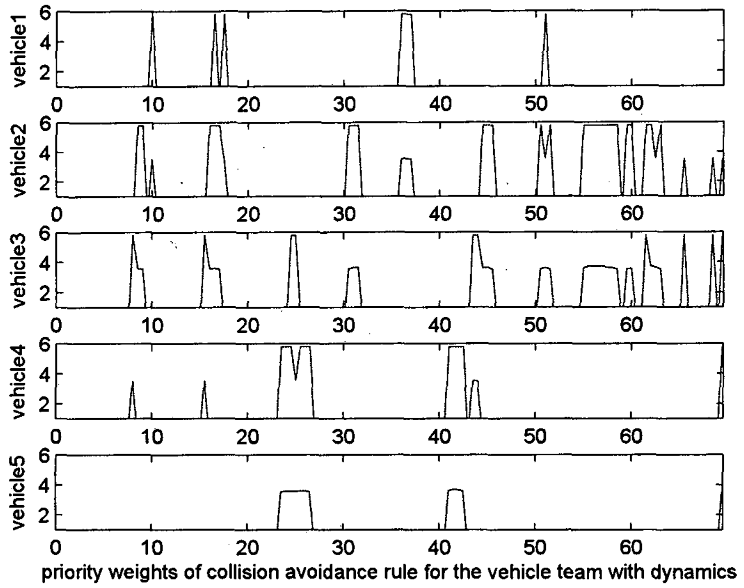
Figure 5.13: Travel trajectories between simulations with and without a dynamic model of SUBZERO

Figure 5.14(a) and 5.14(b) plot the priority weights w_4 of collision avoidance rules of 5 team members between simulations with and without a dynamic model of the vehicle. w_4 changes only when a vehicle finds itself too close to its neighbours. The peaks in both Figures indicate that the vehicle has a large possibility of colliding with its neighbours. We can see that the peaks appear not often in Figure 5.14(a). Compared to Figure 5.14(a), the peaks appear frequently in Figure 5.14(b). This comparison shows that vehicles without dynamics are frequently increasing the weight to keep away from neighbours although they are made closer at the same time by rule 1 going towards the centre of the group. Obviously vehicles with dynamics could save travel time by avoiding frequent changing of heading direction.

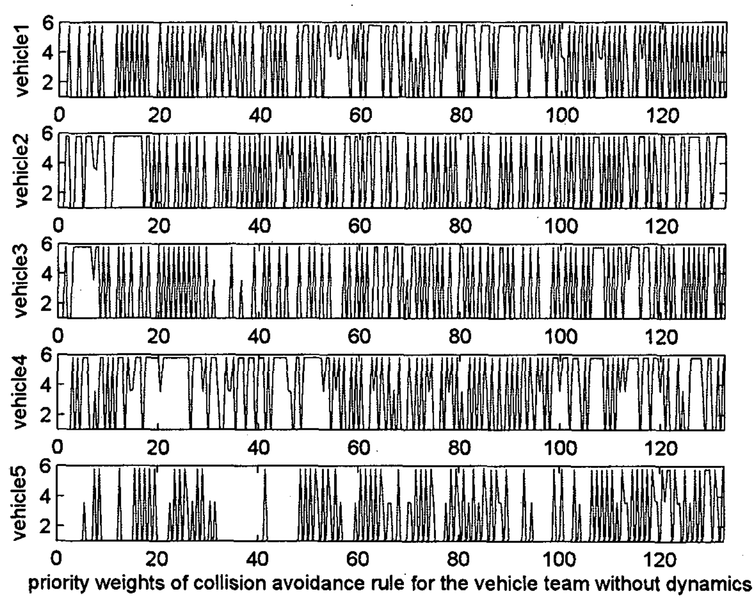
Figures 5.15(a) and 5.15(b) plot the distances from vehicles to centre of the team over time between simulations with and without a dynamic model of the vehicle. Table 5.2 shows the average distances. As shown in both Figures and Table, we can see that vehicles 1, 2, 4 and 5 without dynamics is closer to the centre of the team than vehicles with dynamics. This is because vehicles can change heading direction immediately to move towards centre of the team. Only vehicle 3 without dynamics is further from the center of the team than vehicle 3 with dynamics because the locations of vehicles with dynamics are nearly on a line and vehicle 3 is located closer the center of the line. Vehicles without dynamics are closer together, and hence have an increased risk of collision.

5.5 Conclusions

This chapter has described a set of behaviour-based rules with adaptive priority weights controlled by the fuzzy logic method. The results indicate that the fuzzy logic method could optimize the priority weights according to the current situation

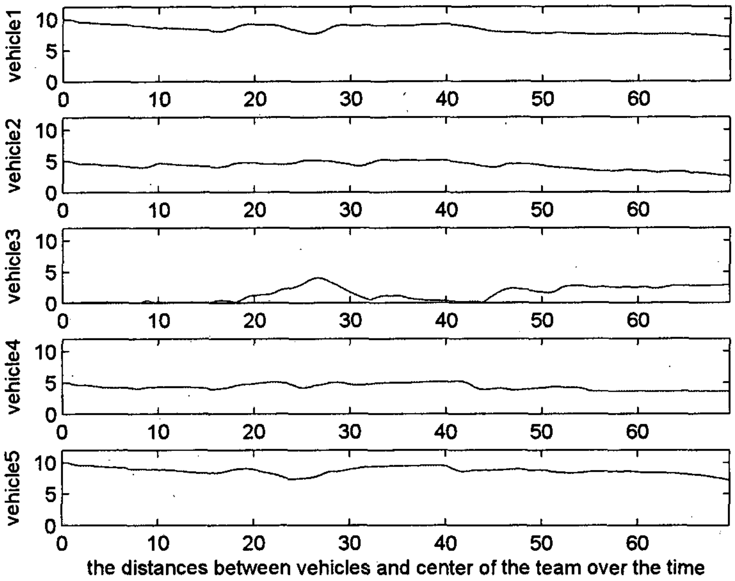


(a)

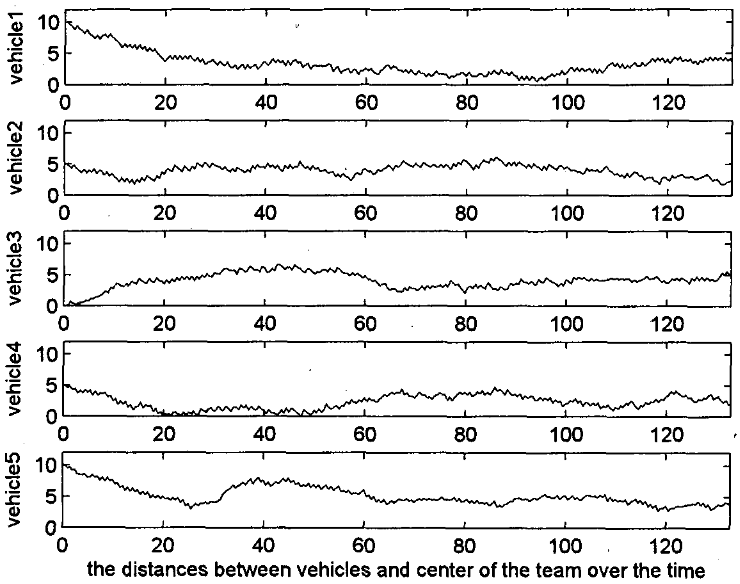


(b)

Figure 5.14: Priority weights of collision avoidance rules between simulations with and without dynamic model of a vehicle



(a)



(b)

Figure 5.15: The comparison of the average distances from vehicles to centre of the team between simulations with and without dynamic model of a vehicle

Table 5.2: The average distances from vehicles to centre of the team between simulations with and without dynamic model of a vehicle

Vehicles	**Distances to centre of the team (metres)**	
	with dynamics	**without dynamics**
1	8.39	**3.36**
2	4.24	**3.94**
3	**1.40**	4.03
4	4.30	**2.32**
5	8.68	**5.23**

and the behaviour-based algorithm is a possible solution for a multi-UUV cooperative mission. From simple scenario simulation, the results show that the different parameters have a significant effect on a mission. These parameters also affect the requirements for vehicle design and sensor specifications.

In the next chapter, in order to verify the suitability of the algorithm in different environments, we will apply this algorithm to deal with more complicated scenarios. Water flow is imposed on the environment to simulate a tidal stream.

Chapter 6

Team Cooperation of Multiple UUVs in a Water Flow Environment

This chapter continues using behaviour-based cooperation with the fuzzied weights method but extends the application into a water flow environment. The majority of this chapter has been accepted as a journal article '*Intelligent behaviour-based team UUVs cooperation and navigation in a water flow environment*' in the international journal 'Ocean Engineering' and has been published.

6.1 Water flow environment

In our simulations, we simulate a simple water flow environment which is inviscid and irrotational. The inviscid flow means that the flow has no resistance to shear stress (Symon 1971). The irrotational flow means the flow does not move in circular or helical motions. Wejchert and Haumann (Wejchert and Haumann 1991) produced 4 flow primitives (figure 6.1). Uniform flow is when the fluid velocity follows in straight lines; source flow is when fluid comes from a source and moves out from all directions;

sink flow is the opposite of source flow; in vortex flow fluid moves around in concentric circles. Most flow is constructed by a combination of these 4 flow primitives.

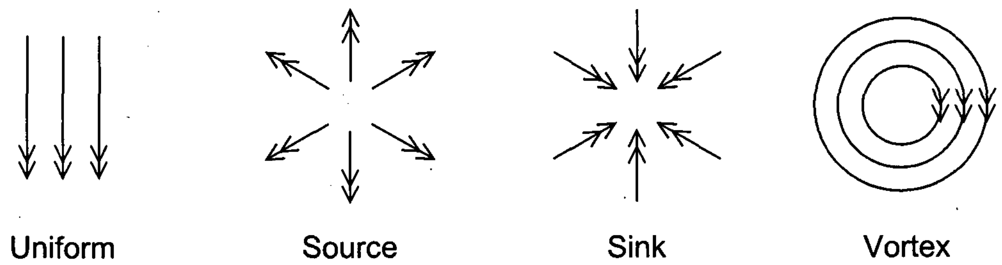


Figure 6.1: The flow primitives

In addition, the model constructed by Tu (Tu 1999) of a non-turbulent flow field with low computational cost, using a uniform flow field with sinusoidal strength and a source flow at each cylindrical obstacle, is used as a basis for generation of the water flow fields for this study. In our simulation, the water flow field in the tank only consists of uniform flow with constant strength and source flow from cylindrical obstacles in a 2D environment. The velocity field of the uniform flow is

$$v_{uniform} = a\hat{\mathbf{u}} \tag{6.1.1}$$

where a is the strength of the flow; $\hat{\mathbf{u}}$ is the unit vector indicating the orientation of the uniform flow.

The source flow velocity field at cylinder obstacles is defined as

$$v_{source} = \frac{a}{(d-r)^2}\hat{\mathbf{d}} \tag{6.1.2}$$

where d is the Euclidean geometry distance from the current position to the centre of the source, r is the radius of cylinder obstacle and $\hat{\mathbf{d}}$ is the unit vector pointing outward from the centre of the source (Tu 1999). The source flow is added to the uniform flow field in order to approximate the reflection effect of water flow around obstacles.

Figure 6.2 simulates the water flow field in a $100m \times 100m$ pool field with a cylindrical obstacle having a defined central location and radius. The strength of the flow is $a = 1m/s$ and the radius of obstacle is $r = 5m$.

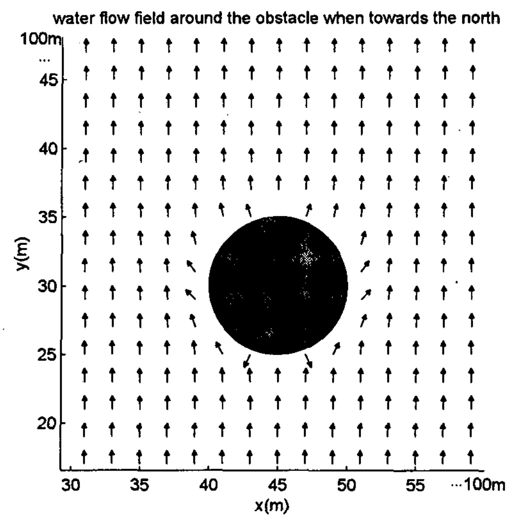
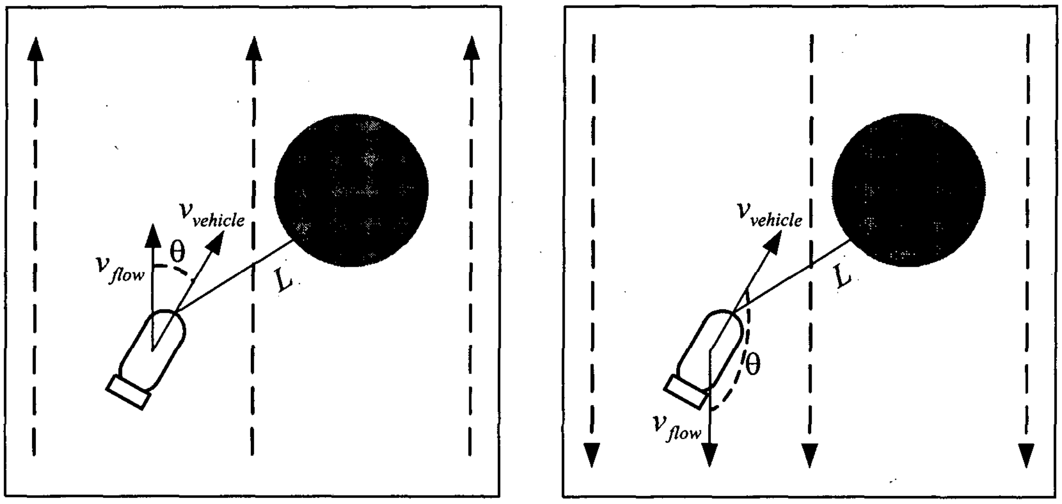


Figure 6.2: Water flow field in our simulation with a defined obstacle.

We can clearly see from Figure 6.2 that the water flow around the obstacles varies due to reflection effects from the obstacle.

In Figure 6.3(a), when velocity directions of the vehicle and the water flow correspond, the speed of the vehicle relative to an obstacle will be much higher. Therefore, we must increase the minimum tolerable distance between obstacles and vehicles in order to get enough time to rotate for avoidance. On the contrary, in Figure 6.3(b), when velocity directions of vehicle and water flow are opposite, the speed relative to obstacles will be decreased by resistance of the water flow and the minimum tolerable distance between



(a) vehicle velocity and water flow velocity are in the same general directions. $(0 < \theta < \frac{\pi}{2})$

(b) vehicle velocity and water flow velocity are in opposite directions. $(\frac{\pi}{2} < \theta < \pi)$

Figure 6.3: The relationship between the vehicle and water flow velocity directions, where v_{flow} is the water flow velocity, $v_{vehicle}$ is the vehicle velocity and L is the distance between the vehicle and the obstacle.

obstacles and vehicles need not be so high. Decreasing the distance between obstacles and vehicles may reduce transit time and save energy on vehicle manoeuvres.

In order to decide upon the minimum tolerable distance to the obstacles according to the vehicle and flow situations, a fuzzy logic controller F_w is designed with one input variable d_w which is the angle difference between the water flow direction and the direction of vehicle towards the danger. The input variable has two linguistic values *same* and *opposite* in Figure 6.4. The output variable is the minimum tolerable distance L_m with the range $\left[L \quad \frac{V_w+V_U}{V_U}L \right]$, where L is the minimum tolerable distance in waterflow-free environment, V_w is the speed of water flow, V_U is the speed of the vehicle. It has two linguistic values *high* and *low* which are distinguished by the centre point $L_c = \frac{V_w+2V_U}{2V_U}L$ in Figure 6.4 . The following fuzzy rules represent the function.

IF d_w is *same* THEN L_m is *long*;
IF d_w is *opposite* THEN L_m is *short*.

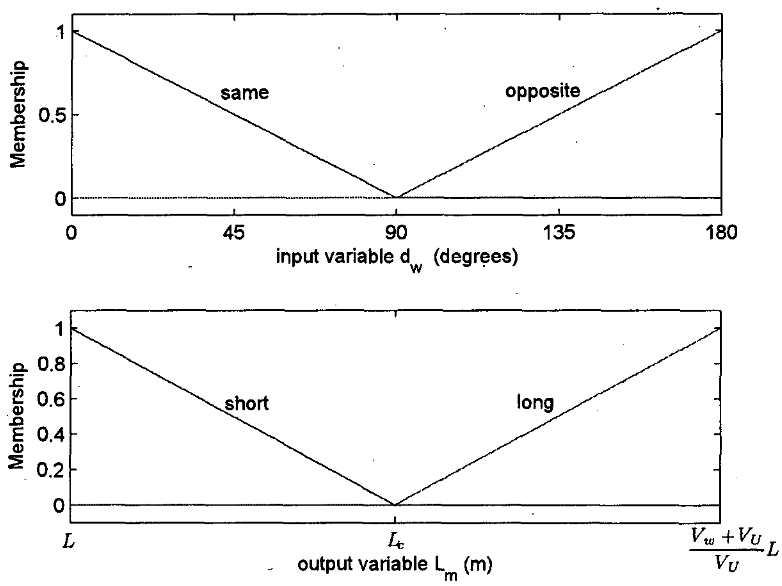


Figure 6.4: The memberships of the fuzzy controller F_w

6.2 Behaviour-based rules with fuzzy logic

As same as the behaviour-based method described in the previous chapter, each behavioural rule i is applied for generating a steering angle factor β_i to build the next step desired steering angle. However, due to the effect of waterflow, the obstacle avoidance rule and collision avoidance rule are modified to avoid the danger caused by waterflow. The LOS guidance rule is modified to minimize the trajectories of travel. In the following sections, the modified behavioural rules are described and the steering angle factors are formulated. The next step desired steering angle is calculated by the combination of all steering angle factors multiplied by their relative priority weights. The angle β_i is ensured in the proper quadrant. Before we describe the rules in a waterflow environment, here we must assume that vehicle can measure the waterflow velocity in order to calculate the modified LOS angle to offset the effects of the waterflow. In practice, this could be done by for example an 'acoustic doppler velocimeter (ADV)'. An ADV is mounted on the nose of the UUV and measures three-dimensional water velocity using sound waves (Sontek 1997). ADV is an active sonar that transmits acoustic signals and then receives their echoes reflected by sound scatterers in the water, like plankton or other floating particles. From the measured frequency shift between emissions and echoes, flow velocity is deduced based on the Doppler principle (Zhang 2001).

In this chapter, we choose the following five behaviour rules:

1. Rule 1: tracking the centre of the team.
2. Rule 2: matching the velocity with the teammates.
3. Rule 3: guidance rule.
4. Rule 4: avoiding collision with teammates.
5. Rule 5: avoiding obstacles.

Compared with the behaviour rules in chapter 4, rule 6 is not included in this chapter because remaining in certain boundary is not necessary within an open ocean with tidal flow. We classify these behaviour rules to three levels as shown in Figure 6.5 . The rules which possibly avoid danger have the highest priority weights. The guidance rule is assigned to the medium priority level since it is the main task of the mission. The collective behaviour rules have the lowest priority weights to maintain the vehicles within the team.

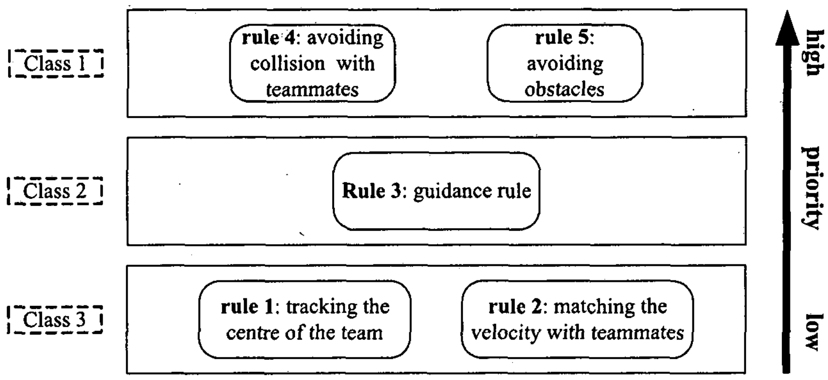


Figure 6.5: Three priority classes of the behaviour rules

Next, we not only describe the behaviour rules, but also explain how to use the fuzzy logic method to generate priority weights for the behaviour rules on-line.

6.2.1 Cooperative behaviour-based rules

The cooperative behaviour-based rules include:

- Rule 1: tracking the centre of the team.
- Rule 2: matching the velocity with the teammates.

These rules remain same as rule 1 and 2 in the previous chapter since waterflow does not affect the steering angle factor in both rules. Please refer to rule 1 and 2 in section 4.2 for more detail.

6.2.2 Rule 3: the modified line of sight guidance rule

Based on the description in section 4.2.3, in a waterflow-free environment, the LOS angle to the target is defined as:

$$\theta_{TU} = \arctan\left(\frac{y_t - y_U}{x_t - x_U}\right) \tag{6.2.1}$$

The steering angle factor β_3 is equal to θ_{TU}:

$$\beta_3 = \theta_{TU} \tag{6.2.2}$$

When water flow is present, the vehicles drift away from the line of sight direction

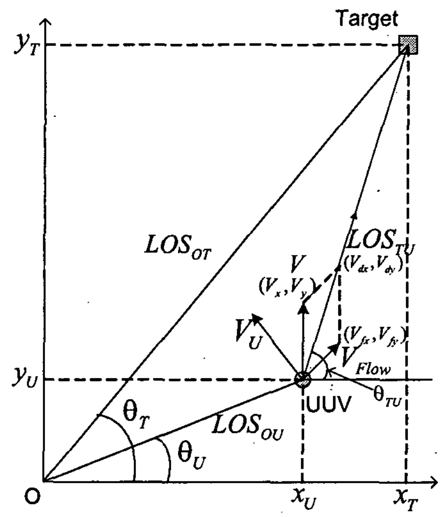


Figure 6.6: The modified LOS guidance rule in a water flow environment

if the water flow is in a different direction. For this reason, the modified line of sight method based on geometry is suggested. The desired velocity V_d with reference to the coordinate is defined as:

$$V_{dx} = V_d \cdot \cos\theta_{TU} \tag{6.2.3}$$

$$V_{dy} = V_d \cdot \sin\theta_{TU} \tag{6.2.4}$$

where (V_{dx}, V_{dy}) is the coordinate of V_d.

The velocity with reference to the coordinate is the sum of the vehicle velocity and the water flow velocity.

$$V_x + V_{fx} = V_{dx} \tag{6.2.5}$$

$$V_y + V_{fy} = V_{dy} \tag{6.2.6}$$

where (V_x, V_y) is the coordinate of the vehicle velocity and (V_{fx}, V_{fy}) is the coordinate of the water flow.

For the vehicle velocity

$$V_x = V_{dx} - V_{fx} \tag{6.2.7}$$

$$V_y = V_{dy} - V_{fy} \tag{6.2.8}$$

The steering angle factor can be obtained as follows:

$$\begin{aligned} \beta_3 &= \arctan\left(\frac{V_y}{V_x}\right) \\ &= \arctan\left(\frac{V \sin\left(\arctan\left(\frac{y_T - y_U}{x_T - x_U}\right)\right) - V_{fy}}{V \cos\left(\arctan\left(\frac{y_T - y_U}{x_T - x_U}\right)\right) - V_{fx}}\right) \end{aligned} \tag{6.2.9}$$

6.2.3 Rule 4: avoiding collision with teammates

Rule 4 ensures that vehicles do not collide when they are close. In Figure 6.7, L_{mt} is the minimum tolerable distance between two vehicles. When a vehicle detects that there is a teammate within the intolerable distance, it will try to steer away from the position of the teammate. Sometimes, there are two teammates within the danger zone, in which case the vehicle will turn more away from the teammate which is the nearest to itself. From Figure 6.7, the vehicle on the right is the nearest to the central vehicle and so the central vehicle will steer away from the vehicle on the right. The steering angle factor β_4 can be written as:

$$\beta_4 = -\arctan\left(\frac{py_{nearest} - py_{current}}{px_{nearest} - px_{current}}\right) \tag{6.2.10}$$

where $(px_{closest}, py_{nearest})$ is the coordinate of the position of the teammate which is the nearest to the current vehicle.

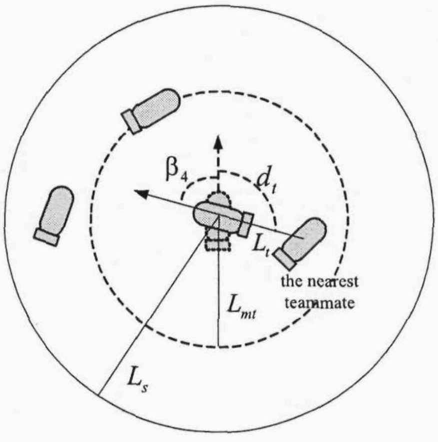


Figure 6.7: The description of rule 4: avoiding collision with teammates

To make weight values adaptive, we use a fuzzy logic controller F_4 to estimate the weight values in real time. At first, we must use the fuzzy controller F_w in section 6.1 to estimate the minimum tolerable distance between vehicles L_{mt}. Two linguistic input variables d_t, L_t are declared. In Figure 6.7, d_t is the angle difference between current velocity direction and direction along the position towards the closest teammate. It has two linguistic values *same* and *opposite* (Figure 6.8). L_t is the distance to the nearest teammate. It has two linguistic values *high* and *low* (Figure 6.8) which are distinguished by the L_{mt}. w_4 is the output variable declared with four linguistic values, *very low*, *low*, *high* and *very high* (Figure 6.9). It presents the priority weight value according to the classes in Figure 6.5. The following fuzzy rules represent the function to decide the weight.

IF L_t *is short* and d_t *is opposite* THEN w_4 *is very high*;
IF L_t *is short* and d_t *is same* THEN w_4 *is high*;
IF L_t *is long* and d_t *is opposite* THEN w_4 *is low*;
IF L_t *is long* and d_t *is same* THEN w_4 *is very low*.

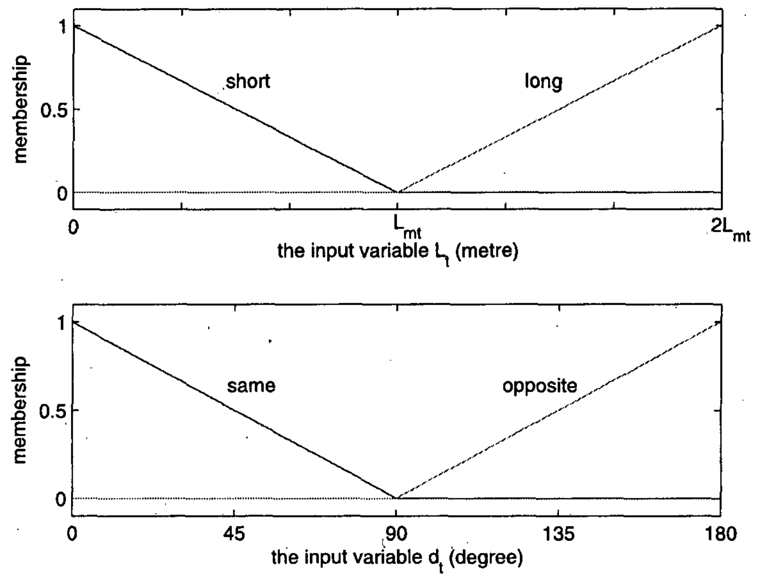


Figure 6.8: The membership of the input variables

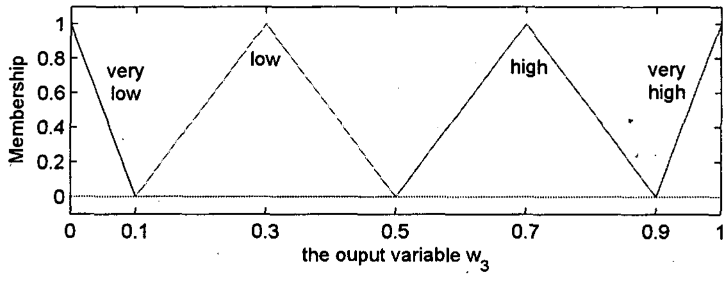


Figure 6.9: The membership of the output variable w_4

6.2.4 Rule 5: avoiding obstacles

Rule 5 does a similar thing to rule 4. As shown in Figure 6.10, the vehicle will steer away from an obstacle when the distance L_o between vehicle and obstacle is smaller than the minimum tolerable distance L_{mo}. The desired steering direction is directly against the obstacles. L_{mo} is also decided by the fuzzy controller F_w.

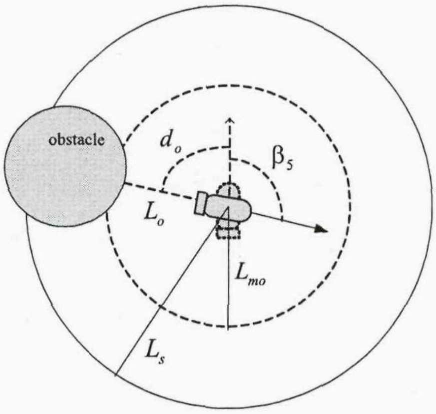


Figure 6.10: The description of rule 5: avoiding obstacles

The steering angle factor β_5 is computed by equation 6.2.11.

$$\beta_5 = -\arctan\left(\frac{py_{obstacle} - py_{current}}{px_{obstacle} - px_{current}}\right) \tag{6.2.11}$$

where $(px_{obstacle}, py_{obstacle})$ is the coordinate of the position of obstacle.

Similarly, the weight value for rule 5 is adaptive estimated by another fuzzy logic controller F_5 which has the same structure as the one shown in rule 3. Two linguistic input variables d_o and L_o are declared. d_o is the different angle between the current velocity and the direction towards the centre of the obstacle. L_o is the distance to the nearest obstacle. The priority weight w_5 is the output variable. The following fuzzy rules represent the function to decide the weight.

IF L_o *is short* and d_o *is opposite* THEN w_5 *is high*;
IF L_o *is short* and d_o *is same* THEN w_5 *is very high*;
IF L_o *is long* and d_o *is opposite* THEN w_5 *is low*;
IF L_o *is long* and d_o *is same* THEN w_5 *is low*.

6.3 The combination of all steering angle factors

As soon as the priority weights are generated for rules 4 and 5, weights for the other rules must be assigned and the fuzzy controller F is customized to generate other rules' weights. This is achieved by the fuzzy logic controller F which is the same as the one in section 5.3 and then the five steering angle factors are multiplied by their relative weights and are combined together to calculate the next desired steering angle by equation 4.3.1 with 5 behaviour rules.

6.4 Simulation implementation

6.4.1 The working flow of team cooperation control in the simulation

Figure 6.11 shows the working flow diagram of team cooperation mission control within the water flow environment. All vehicles are initialised with positions and velocities. All vehicles update their motions at updating time points in terms of the behaviour-based rules and the situation that they meet. The water flow velocity is added to the current vehicle velocities when they update their motions and when they need to decide their next step velocities. The mission ends when the vehicles reach the target. The fuzzy controllers mentioned in the previous sections, F_w,F_4,F_5 and F,

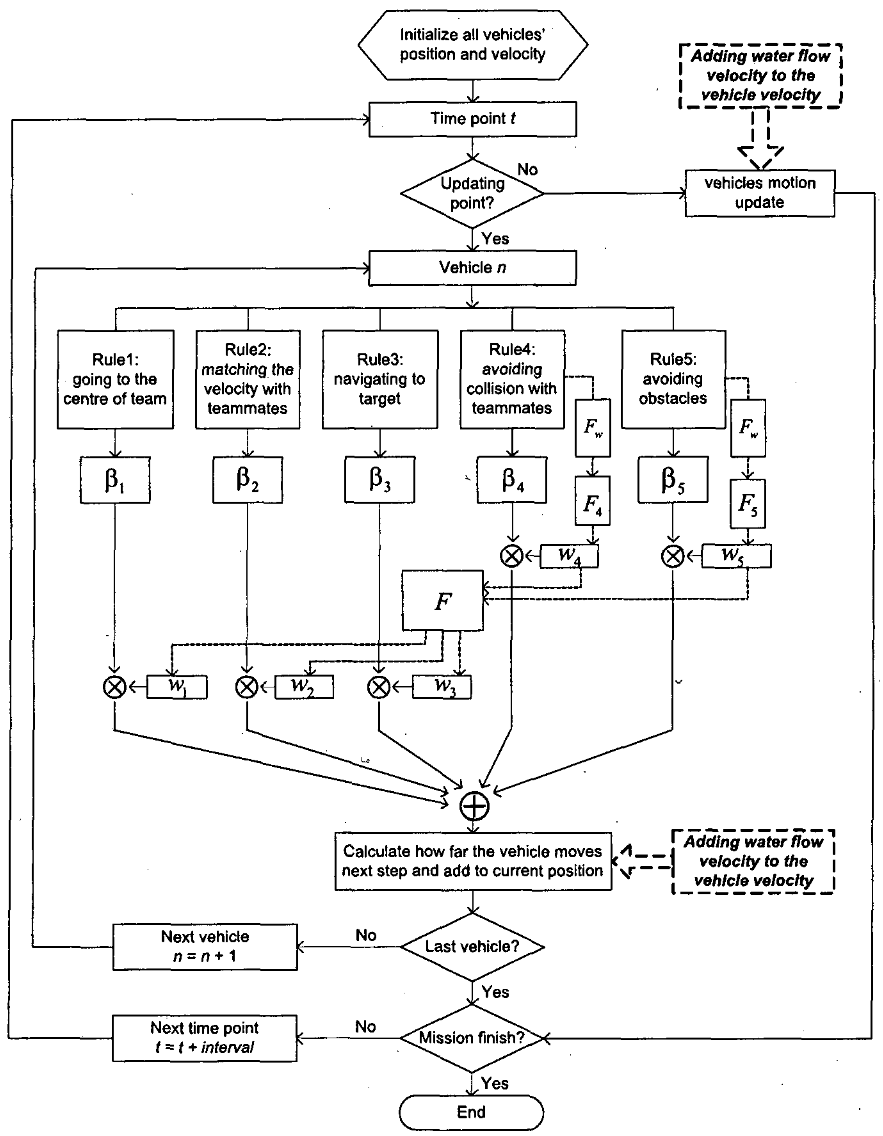


Figure 6.11: The working flow of team cooperation control in the simulation

are shown in this Figure. Compared with the working flow in Figure 5.9 in chapter 5, two F_w controllers are used before F_4,F_5 and determine the inputs for them. For example, F_w generates the minimum tolerable distance to the neighbour and uses it as the input of the fuzzy logic controller F_4. The Matlab environment is used as the simulation platform. The vehicle's motion updating step time is 0.1 second.

6.4.2 Scenario description

The dynamic model of a small vehicle, SUBZERO III, is used to represent each vehicle in the team. Please refer to section 4.4 in chapter 4 for model details.

The simulation mission scenario is simply designed to explain how the algorithm works. The water flow speed is defined as $a = 0.5m/s$ which is slower than the vehicle speed, but changes in direction in different simulations. The area is limited to a $100 \times 100m^2$ square pool. A team of vehicles depart from the starting line and then travel cooperatively and avoid obstacles until they reach the target. There are several parameters which need to be initialised. Parameters V, $V_{initial}$, $P_{initial}$, ω and $F_{updating}$ refer to the descriptions in Table 4.2. The following Table 6.1 lists parameters L_{mt} and L_{mo}.

Table 6.1: Parameters predefined in the simulation

Parameters	Descriptions	Value	Unit
L_{mt}	The minimum distance between two vehicles	3	m
L_{mo}	The minimum distance between vehicle and obstacles	5	m

We assume that the position of the start point and initial velocity of each vehicle are initialised by the operator before the mission starts. According to the assumption, each vehicle knows the direction of the water flow vector at each time point. The range of neighbouring vehicles is predefined. The position and velocity vectors of the neighbouring vehicles are also known but the position of obstacles and targets would have to be measured in practice.

6.5 Simulation Results

6.5.1 8 snapshots of a simulation to a target

First of all, we need to see whether the system is working in a typical scenario. Rather than showing the trajectories of simulation, we again plot 8 snapshots in Figure 6.12 which can illustrate how the vehicles move towards the target position. There is only one obstacle between the starting line and target. This position of the obstacle is such that the vehicles are to meet and avoid it in most of the situations. the meaning of the symbols in all Figures is the same as the symbols in chapter 4.

The water flow direction is towards the east. All vehicles departed from their initial positions and, up to 18.5s, all vehicles remained as a group and advanced towards the target. At time 18.5s, the vehicles detected an obstacle in front and then decided to turn right in avoidance. At 20.2s, the vehicles steered away from the obstacle. At 25.3s, the vehicles steered to the position which is further than the minimum tolerable distance from the obstacle L_{mo}. They then started to track the target again at time 29.5s and finally, one vehicle reached the target at time 56s. At time 10.6s and time 56s, we can see that the vehicles' orientation differs from the normal LOS. This is caused by the modified LOS guidance rule which accounts for the waterflow. The simulation illustrates that the vehicles are successful in achieving the mission without any collisions in this case.

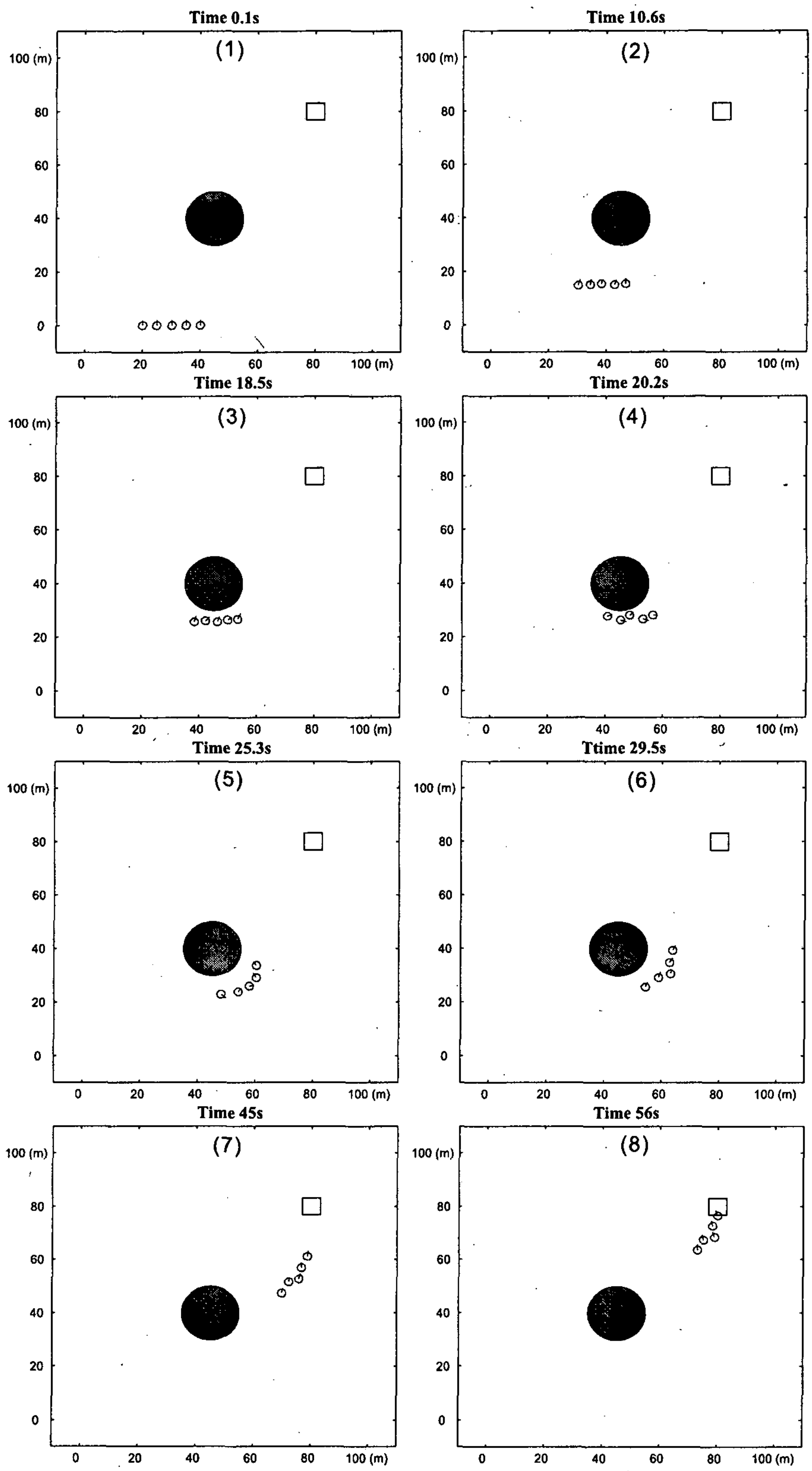


Figure 6.12: 8 snapshots in a simulated journey to a target. These snapshots are numbered by time order. The waterflow is towards the east.

6.5.2 Simulation trajectories under different situations

In the following simulations, we investigate the algorithms under different situations. This is not only to help us to test the feasibility of the cooperation algorithm, but also to indicate the advantages of the modified LOS guidance rule. The trajectories and the travel time are compared between the simulations with the normal and modified LOS guidance rules. The travel time is the average of time for the whole team.

Simulation trajectories with no obstacles

When the vehicles are in the obstacle-free environment, most of challenges come from avoiding collision while adapting to compensate for tidal flow direction. Figures 6.13 and 6.14 show the simulation trajectories with the normal and modified LOS guidance rules within the obstacle-free environment. Each guidance rule is simulated with four different water flow directions from the east, north, west and south. As shown in Figure 6.13, we can see that, the trajectories are slightly different with the line of sight orientation. This is caused by the water flow resistance. By using the modified LOS guidance rule, the vehicles get closer to the line of sight orientation although the directions of vehicles are not always directly towards the target. Table 6.2 compares the travel time with the two guidance rules under the different flow situations. It is

Table 6.2: Comparison of the travel times with the normal and modified LOS guidance rules within an obstacle-free environment. t_n is the travel time with the normal LOS guidance rule. t_m is the travel time with the modified LOS guidance rule. The bold letters denote the shorter times. (0 obstacle, 4 waterflow directions)

	t_n	t_m
East	67.48s	**57.50s**
North	56.52s	**56.32s**
West	90.68s	**80.32s**
South	**91.94s**	93.70s

shown that the modified LOS guidance rule reduces the travel time. However, when the water flow is towards the south, the vehicles spent slightly less time by the normal LOS guidance rule than the modified normal LOS guidance but it can be seen that the vehicles reached the nearest side of the target when the normal LOS guidance rule was applied.

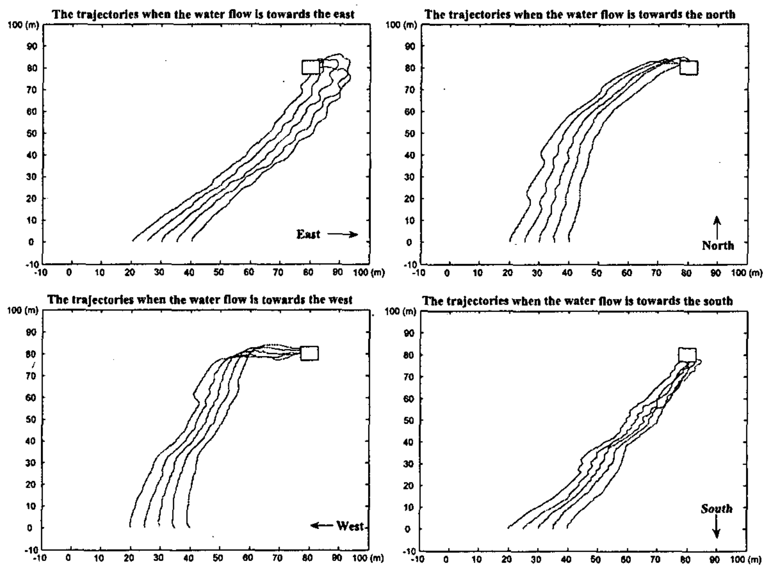


Figure 6.13: The simulation trajectories with the normal LOS guidance rule within an obstacle-free environment. Waterflow direction is shown in each case.

Simulation trajectories with only 1 obstacle

By adding an obstacle in the environment, the mission becomes more complicated and the vehicles must synchronously face the challenge of avoiding the obstacle in addition to the tidal stream. Figures 6.15 and 6.16 show the trajectories with the normal and modified LOS guidance rules respectively when the water flow is in 4 different directions. It is clear that all missions were successful but it is interesting to note that, with a tide moving from the west, the modified LOS guidance rule took

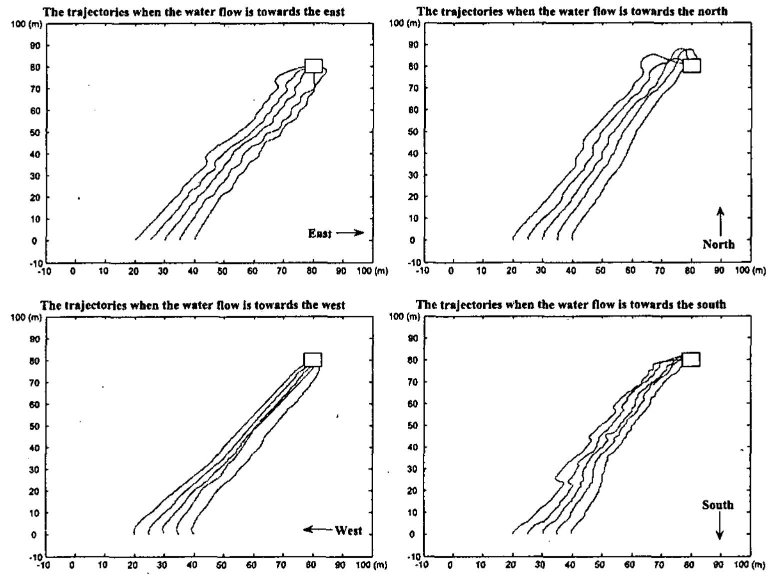


Figure 6.14: The simulation trajectories with the modified LOS guidance rule within an obstacle-free environment. Waterflow direction is shown in each case.

the team on a shorter route around the obstacle whereas the normal LOS rule allowed drift to occur in the course initially. From Table 6.3, we can see that the modified LOS guidance also reduced the travel time under all situations.

Table 6.3: Comparison of the travel time with the normal and modified LOS guidance rules. t_n is the travel time with the normal LOS guidance rule. t_m is the travel time with the modified LOS guidance rule. (1 obstacle, 4 waterflow directions)

	t_n	t_m
East	64.58s	**61.08s**
North	60.06s	**57.42s**
West	103.22s	**89.16s**
South	100.36s	**98.30s**

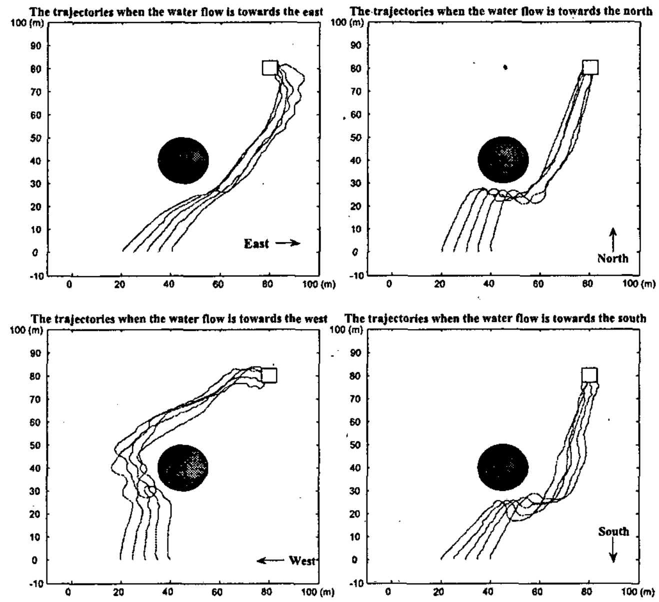


Figure 6.15: The simulation trajectories with the normal LOS guidance rule with 1 obstacle and different waterflow directions.

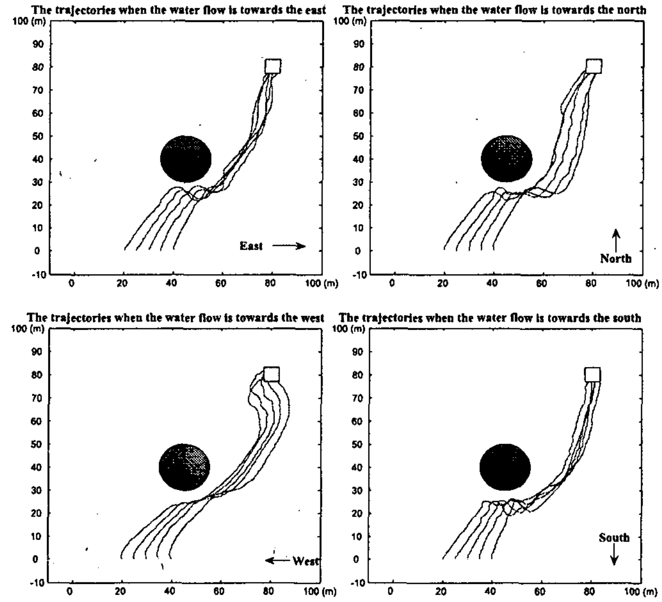


Figure 6.16: The simulation trajectories with the modified LOS guidance rule with 1 obstacle and different waterflow directions.

Simulation trajectories with 3 obstacles

Three obstacles considerably complicate the mission. Figures 6.17 and 6.18 show the trajectories with the normal and modified LOS guidance rules when the four water flow directions are simulated. However, the modified LOS guidance rule reduced the travel time only when the water flow was towards the east and the south. In this complicate environment, the modified LOS guidance rule cannot always reduce the travel time and took longer when the flow was from the North and West. For example, sometimes, the water flow pushes the vehicles not only away from the line of sight, but also away from the obstacles which are in the line of sight. The travel time is less when the vehicles do not spend as much time avoiding the obstacles.

Table 6.4: Comparison of the travel time with the normal and modified LOS guidance rules. t_n is the travel time with the normal LOS guidance rule. t_m is the travel time with the modified LOS guidance rule. (3 obstacles, 4 waterflow directions)

	t_n	t_m
East	86.86s	**77.30s**
North	**64.04s**	72.56s
West	**100.10s**	104.24s
South	118.26s	**112.72s**

6.6 Conclusion

This chapter has presented a behaviour-based approach with adaptive fuzzy logic assignment priority weights to investigate the feasibility of a team of UUVs cooperating in a water flow environment. From the simulation results, we can conclude that the approach has potential for further investigation for controlling a team of real cooperative UUVs. The modified LOS guidance rule provides the ability to reduce the

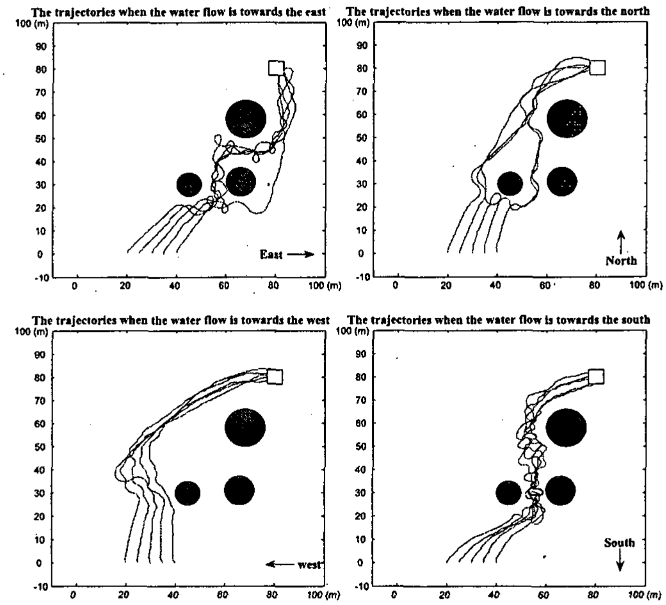


Figure 6.17: The simulation trajectories with the normal LOS guidance rule with 3 obstacles environment and different waterflow directions.

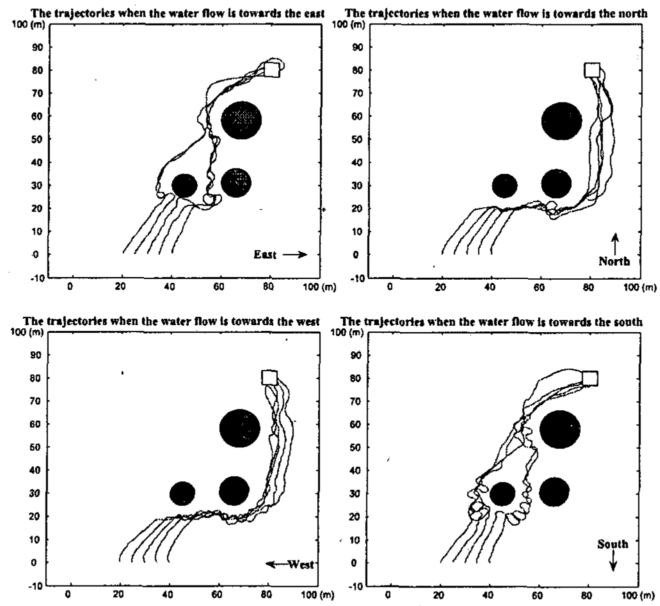


Figure 6.18: The simulation trajectories with the modified LOS guidance rule with 3 obstacles environment and different waterflow directions.

travel time although the travel time is mainly affected by the position of the obstacles when the environment becomes complicated. The results may provide clues to help minimize the trajectory of a mission and decrease the energy consumption which is always desirable in UUV deployment. In future work, extending work into a 3-D simulation scenario should be addressed.

Chapter 7

Formation Control of a Team of UUVs

In this chapter, we present a review of the current control issues and strategies for formation control of a group of unmanned autonomous vehicles/robots. Formation control has broad applications and has become an increasingly active research topic in recent years. Based on the behaviour-based method of previous chapters, we provide behaviour-based solutions for line and circle formation control under different simulation scenarios in this chapter.

7.1 Introduction

Formation behaviour is common in nature, such as geese flying as 'V' formation when migrating. The first benefit is that it saves their energy. The second benefit is that it is easy to keep track of every bird in the group. In general, animals not only benefit by minimizing the chance of being eaten by predators, but also by combining group members' senses to maximize the chance of detecting both predators and food. In chapter 2, we have introduced that robotics researchers and engineers have realized that formation control can bring positive effects for multiple robots or vehicles applications and have drawn from these biological studies to develop formation control approaches for both simulated agents and robots or vehicles.

In recent years, formation control has been widely applied for coordination of multiple robots, unmanned air/underwater vehicles and even satellites, aircraft and spaceships (Beard et al. 2001). In general, moving in formation has many advantages over conventional systems, for example, it can reduce the system cost, increase the robustness and efficiency of the system while providing redundancy, reconfiguration ability and structural flexibility for the system. A team of low-cost vehicles with specified formation could cover an area quickly, e.g. for pollution detection and clearance and benefit from the advantages of maintaining a desired spatial pattern that can reduce the requirement of communications between the vehicles. For example, vehicles with a circle formation can use broadcast communications to transmit and receive the information (Cheng et al. 2006). The team size can then be determined to be less than the sensor range. Each vehicle in a line formation can act as a router for the nearest teammates. When some vehicles are out of sensor range of other vehicles, information can still be exchanged by routing. This significant advantage may broaden the operating range of the team. However, when vehicles within a team cannot detect the shape of an obstacle in the flight path, they must keep together rather than being separated into two teams while passing the obstacle; otherwise the vehicles may not be able to reform again if some are out of sensor and routing range.

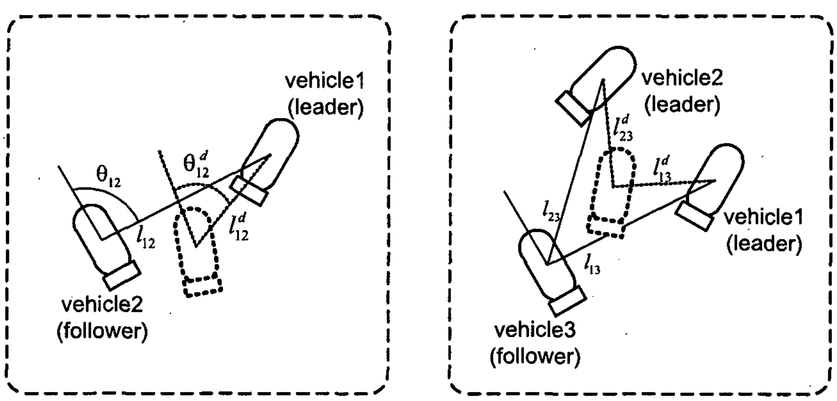
Three methods to formation control of multiple vehicles have been developed over the years:

- leader-follower strategy
- virtual structure approach

- behaviour-based strategy

Each method has its own advantages and disadvantages and here we give a short review of these methods.

In the *leader-follower* method, one or more vehicles are considered as leaders and the rest of vehicles in the team act as followers. At the beginning of the mission, the trajectories are given to the leaders. The leaders follow the trajectories as the reference and the other vehicles keep a distance between the leaders and themselves according to the formation shape. An advantage of the leader-follower approach is that it is easy to understand and implement (Cowan et al. 2003). However, a disadvantage is that there is no explicit feedback to the formation since there is only local sensor-based information available for each vehicle (Ghommem et al. 2007).



(a) Notion for the $l-\theta$ controller (b) Notion for the $l-l$ controller

Figure 7.1: Two types of feedback controllers.

There are two types of feedback controllers for maintaining formations of multiple robots. The first one is the $l-\theta$ controller and the second one is the $l-l$ controller (Chen and Wang 2005). As shown in Figure 7.1(a), the objective of the $l-\theta$ controller is to maintain a desired length l_{12}^d and a desired relative angle θ_{12}^d between the leader and the follower for two vehicles (Desai et al. 1998). The controller can be designed to converge l_{12} and θ_{12} to the desired value. Figure 7.1(b) shows the $l-l$ controller which considers the relative position of three mobile vehicles. In the Figure, vehicle3 is controlled to follow two leaders, vehicle1 and vehicle2. The objective is to maintain the desired length l_{13}^d and l_{23}^d between the follower and its two leaders. A $l-l$ controller was also designed by Desai et al. (1998). Dasai applied both types control algorithms to get good numerical simulations and successful experiments using a testbed consisting of three robots (Das et al. 2002). Li studied a fuzzy algorithm to track and maintain close relative spacing to the leader of a group of Unmanned Air Vehicles (UAVs) (Li et al. 2005). However, the leader-follower method has a fatal shortcoming that the follower is dependent on the leader and they would lose control when the leader fails.

The concept of *virtual structure* was first introduced by Tan and Lewis (1996) and is aimed at spacecraft or small satellite formation flying control (Beard et al. 2000). The virtual structure approach treats the entire formation as a single virtual structure that acts as a single rigid body. The control law for a single vehicle is derived by defining the dynamics of the virtual structure and then translating the motion of the virtual structure into the desired motion for each vehicle. The advantages of the virtual structure approach are that it is fairly easy to prescribe the coordinated behaviour for the group and the formation can be maintained very well during manoeuvres, i.e. the virtual structure can evolve as a whole in a given direction with some given orientation, and can maintain a rigid geometric relationship among multiple vehicles. However, if the formation has to maintain the exact same virtual structure at all times, the potential applications are limited especially when the formation shape is time-varying or needs to be frequently reconfigured.

The *behaviour-based* approach uses a set of desired behaviour rules for each member in the group (Balch and Arkin 1995; 1998). Each behaviour rule generates a vector representing the desired behaviour response to sensory input. Possible behaviour rules include collision avoidance, obstacle avoidance, navigation and formation keeping. The advantages of this approach are: it is natural to design control strategies when vehicles have multiple competing objectives, and an explicit feedback is included through communication between neighbours. The disadvantages are that the group behaviour cannot be explicitly defined, and it is difficult to analyze the approach mathematically and guarantee the group stability. McDowell and Chen used machine learning techniques for creation and maintenance of loose formations of autonomous vehicles for the purpose of traveling from a mission staging area to an area of interest (McDowell et al. 2002). They applied a genetic algorithm(GA) to find relevant parameters for a feed-forward neural network that controls heading and speed changes of UUVs. However, the method has not been tested under some emergency situations within a complicated environment, such as where obstacle avoidance is necessary.

In this chapter, we develop the behaviour-based method described in the previous chapters for formation control of a group of UUVs by adding an additional formation control behaviour rule. The fuzzy logic controllers are used to estimate the priority weights for the rule base. Compared with existing work on formation control, our approach applies artificial intelligent techniques (fuzzy logic) to estimate priority weights for rules according to the situation the vehicles meet, in particular when formation control and emergency vehicle or obstacle avoidance happen simultaneously.

7.2 Line formation control

In line formation pattern, the vehicles are spaced with a certain distance between teammates which are all heading in the same direction. Vehicles dispersedly lie on a line. This leads to the increasing of the team size. The longer distance between two neighbouring teammates, the larger team size.

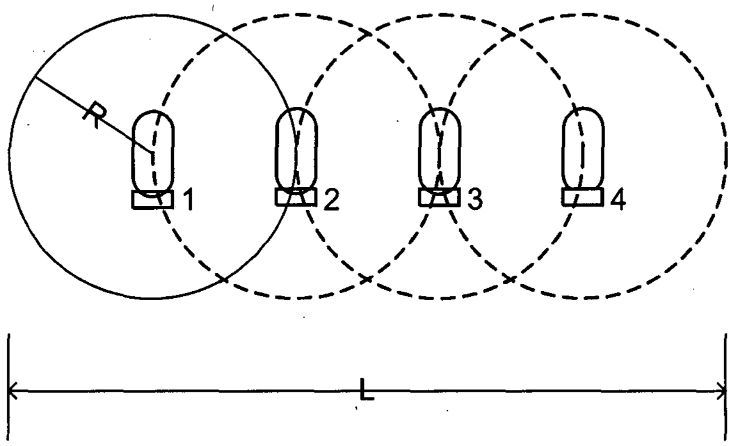


Figure 7.2: The communications scheme between vehicles in a line formation pattern. R is the communication range of a single vehicle. L is the maximum team size.

Each vehicle can act as a router to transmit the information between its teammates and itself. As shown in Figure 7.2, the transmitters of vehicle 1 cannot reach vehicle 3, but vehicle 1 can exchange information with vehicle 2 which is in its range. Then vehicle 2 can route the information from vehicle 1 to vehicle 3. At the same time, vehicle 3 can pass information to vehicle 2 and vehicle 1 and hence information from each vehicle can be shared across the entire team. Some applications could benefit from this feature, for example, map searching in a pollution survey or mine detection in the military field. A larger team size enlarges the searching area and makes the mission quicker and more efficient compared to the cost of the mission with a single large ship.

7.2.1 Behaviour rules for line formation control

The behaviour control algorithm includes of the following rules:

1. Rule 1: Keeping formation
2. Rule 2: Avoiding obstacles
3. Rule 3: Moving to a target

Compared the collective behaviour rules in previous chapters, the keeping formation rule not only must avoid collision between teammates, but also maintain the vehicles in a line formation.

Rule 1: Keeping formation

When the vehicles are cruising with the same speed value, they will keep the same distance as they depart if they have the same heading direction. This rule allows a vehicle to determine the heading direction of its neighbours and to follow the same direction. In Figure 7.3, vehicle 2 is the closest to the obstacle and it must steer angle β_1' to avoid obstacle. Vehicle 2 sends this information to tell other vehicles in the

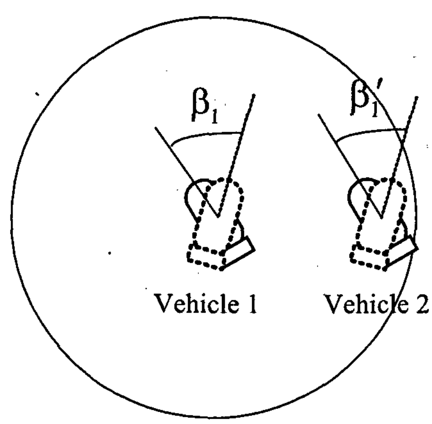


Figure 7.3: Rule 1: keeping formation

line formation pattern. After vehicle 1 receives this information, it follows the same steering angle as vehicle 2's. Then the steering angle factor β_1 is

$$\beta_1 = \beta_1' \tag{7.2.1}$$

In order for the team to learn the heading direction, it is assumed here that the vehicles can receive the heading information from neighbours. In this case of line formation, a vehicle can only receive information from the near neighbours on both sides.

Rule 2: Avoiding obstacles

Each vehicle use the same obstacle avoidance rule which is described in chapter 5. However, since rule 1 is also used here, vehicles have a less chance of reaching the minimum tolerable distance between a vehicle and an obstacle. When a vehicle is first facing the danger of colliding with an obstacle, it will make a decision to steer to an angle factor β_2 away from the obstacle. According to rule 1, the teammates will then follow the same heading with the same priority weighting. Obviously this steering direction will make the other vehicles much further away from the obstacle which may result in an increased travel time since all vehicles need to steer away the target, even if some of them are not within danger range of the obstacle.

The priority weight value w_2 for the rule is generated by the fuzzy logic controller. The design of the fuzzy logic controller and generation process of weight value w_2 can be referred to the description of fuzzy rule 5 in section 5.2.2.

Rule 3: Moving to a target

The basic navigation rule is based on the LOS method. However, in order to keep the same heading across the teammates, the heading direction is determined by the vehicle which is the closest to the target. Other vehicles then follow this heading direction and collectively move to the target. The steering angle factor β_3 can refer back to section 4.2.3 in chapter 4.

In terms of rule 1, the vehicles must have the same heading direction. Therefore, each vehicle in a team can be regarded and treated as an individual body. The motion decision is made by the single vehicle according to the behaviour rules with fuzzy logic controlled priority weights. The single vehicle could be the one which is the closest to the obstacle or the one which is the closest to the target. The motion decision is then transferred into each vehicle.

Relationship between rules and the priority weights

The relationship between avoiding obstacle and navigation rules is represented by the priority weights w_2 and w_3 respectively and these are adapted by the fuzzy logic controllers. The membership function can be determined from Figure 5.4 in chapter 5. The following fuzzy rules represent the function.

IF w_2 *is very high* THEN w_3 *is very low*;
IF w_2 *is high* THEN w_3 *is low*;
IF w_2 *is low* THEN w_3 *is high*;
IF w_2 *is very low* THEN w_3 *is high*;

The motions of vehicles at several stages of a typical mission

We describe several stages to explain the vehicles' motions at several important time points as shown in Figure 7.4 using 5 vehicles in a typical mission for illustration.

Stage 1 Five vehicles cooperate as a team and depart with initial velocity direction which is towards to the North.

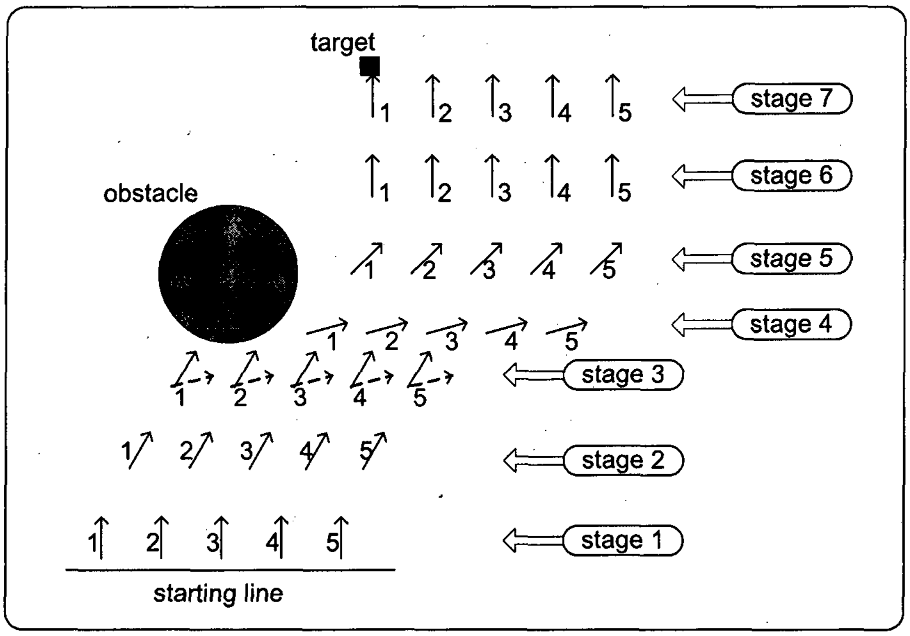


Figure 7.4: Line formation control stages

Stage 2 Five vehicles maintain a horizontal line formation and navigate to the target. At this stage, vehicle 2 is the nearest to the target. Other vehicles follow the same heading direction as vehicle 2.

Stage 3 Vehicle 2 is the first to travel into the minimum distance between a vehicle and an obstacle. The obstacle avoidance rule and fuzzy logic controllers generate a steering angle for vehicle 2 to avoid the obstacle. Other vehicles follow vehicle 2 to steer the same angle. It can be seen that the new steering angle helps vehicle 1 steer away from the obstacle in advance.

Stage 4 Five vehicles keep travelling away from the obstacle.

Stage 5 Vehicles detect the distance to obstacle larger than the minimum tolerable distance to an obstacle and start tracking the target. Vehicle 1 is the closest to the target and so other vehicles steer the angle set by vehicle 1 steers.

Stage 6 Five vehicles track the target.

Stage 7 The mission ends when vehicle 1 reaches the target.

7.2.2 The behaviour control rules when one vehicle is lost in line formation

So far the vehicles keep the same heading direction with the same forward speed. However, there is a chance that one or more vehicles may be lost in formation. If vehicles remain at the same heading direction, there will be an increased gap between the teammates on each side of the lost vehicle. In order to fill the gap and get back into formation, the teammates on both sides must converge to the original formation grouping.

To regroup back to original formation, we make the following assumptions:

- The vehicles' speeds are the same. Vehicles only change heading direction to reform as a team.
- To ensure that the vehicles remain on a line their positions on the y-axis in the coordinate system should be the same.
- At the initial stage, the distance d_f between nearby teammates in line formation is defined. The vehicles keep measuring the distances d_l and d_r to the nearest vehicles on both sides. If the distance d_l or d_r is longer than d_f, the left or right teammate might be lost. In order to be able to detect the nearest teammates, the sensor range d_s of a vehicle must be twice as larger as the predefined distance d_f between teammates($d_s \geq 2d_f$).

The behaviour rules consist of the formation control, the obstacle avoidance and the navigation. The formation control rule differs from the rule described in the previous section since there is now a vehicle that has failed or is lost during the mission and will be described shortly by an illustration of the motions at several stages. The obstacle avoiding and navigation rules duplicate the rules described in the previous section.

However, since the positions of vehicles in line formation need to be accurate, the relationships among the navigation, the obstacle avoidance and the formation control rules are affected. When the vehicles are maneuvering to converge back into formation, the navigation rule is disabled. Otherwise the navigation rule will change the y-axis position of the vehicles. When any vehicle is in danger of colliding with an obstacle, the formation control rule is disabled. The vehicles must use the obstacle avoidance to keep away from the obstacle first and then implement the formation control to maintain the formation pattern.

Assuming that only one vehicle fails during the mission, Figure 7.5 describes the mission stages to explain how the vehicles move in formation.

Stage 1 Vehicles are moving in a line formation towards the target in the northeast.

Stage 2 Vehicle 4 suddenly breaks down due, say a failed battery, and the vehicles in the line are then separated into two groups.

Stage 3 vehicle 3 on the left finds the nearest vehicle on the right is vehicle 5 and the distance between them is larger than d_f. Synchronously the vehicle 5 finds the nearest vehicle on the left is vehicle 3 and also the distance between them is larger than d_f. In order to keep in line formation, the y-axis position of vehicles should be always the same. Vehicles intend to steer in a vertical direction.

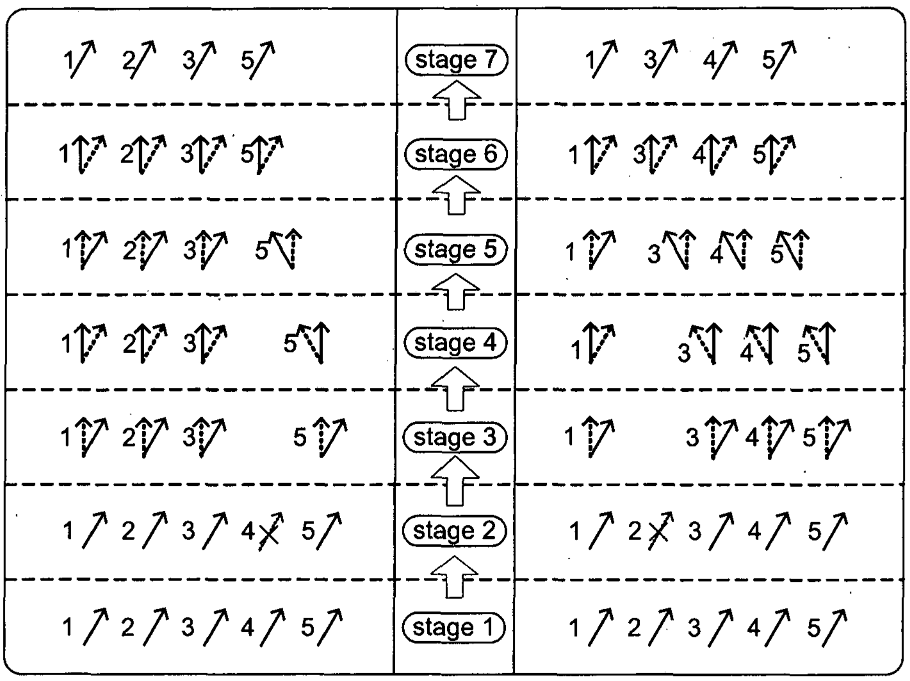


Figure 7.5: Line formation control stages when one vehicle fails in formation. The target is located on the northeast of vehicles. The dotted line arrows with a cross denote the failed vehicles. The real line arrows denote the current heading directions of vehicles. The dotted line arrows denote the intended heading direction of vehicles. The left sets show the case when vehicle 4 fails. The right sets show the case when vehicle 2 fails.

Stage 4 When vehicles reach the vertical heading direction, three of them on the left intend to steer to right and vehicle 5 on the right intends to steer to left.

Stage 5 when the distance d between vehicle 3 and 5 is equal or close to $\frac{3}{2}d_f$, vehicles intend to steer back to vertical direction.

Stage 6 Vehicles recover the spacing distance and converge back to one team.

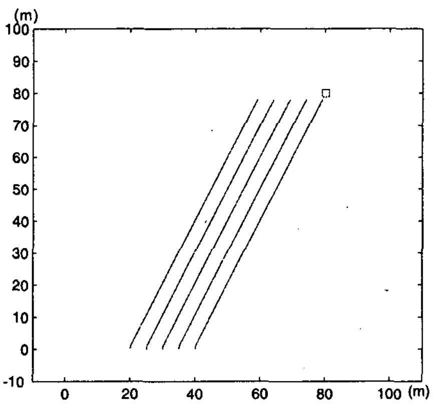
Stage 7 After recovering, vehicles start to track the target again.

Similarly, the stages on the right show how vehicles respond when vehicle 2 fails. In this case, the entire team is splited into two teams by the failed vehicle 2.

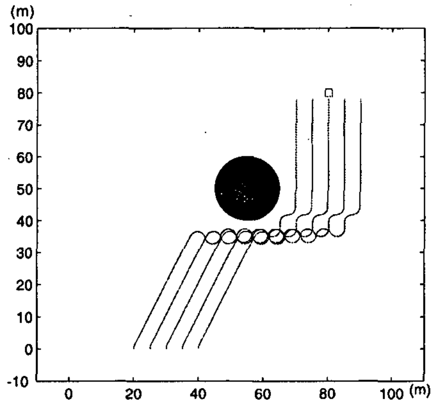
7.2.3 Simulations results

The simulation is under the mission scenario that a group of vehicles start from the starting line, navigate to a target and maintain a line formation during the process. The distance d_f between two nearby teammates is specified here as 5 metres. Other parameters in the simulation are presented in Table 4.2 in chapter 4.

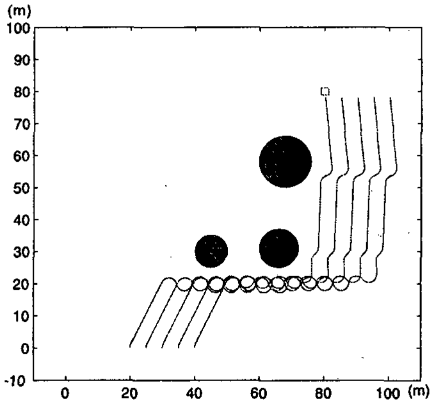
First, we verify that 5 vehicles can keep formation under the line formation control algorithm presented in section 7.2.1. We simulate the situations in which vehicles are in a tidal free environment with non, 1 and 3 obstacles. As shown in Figure 7.6(a), (b) and (c), we can clearly see that the vehicles succeed in keeping the line formation in these different simulations. Next, we simulate the situation that the vehicles converge to the original formation after one vehicle fails in a no obstacle environment as described in section 7.2.2. In Figure 7.7(a), vehicle 2 fails at time point 16s. Vehicle 1 and 3 are not able to converge back to original distance between two nearby neighbours. Compared with Figure 7.7(a), Figure 7.7(b) shows vehicles 1 and



(a) Simulation trajectories with no obstacle



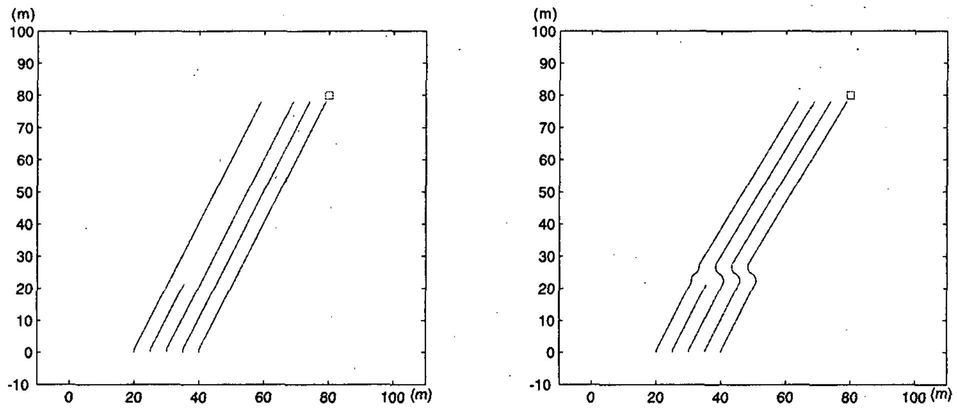
(b) Simulation trajectories with 1 obstacle



(c) Simulation trajectories with 3 obstacles

Figure 7.6: Line formation simulation trajectories within different numbers of obstacles

3 converge back to the original distance between two neighbours. After convergence, we can see that the 4 vehicles, 1,3,4 and 5, maintain the original line formation.



(a) The simulation trajectories with the normal method.

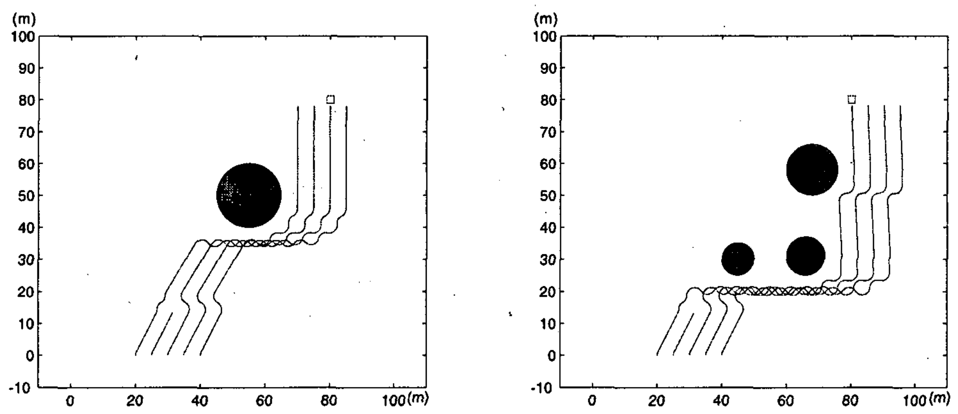
(b) The simulation trajectories with the modified method.

Figure 7.7: Comparison between simulation trajectories with normal and modified method

Figure 7.8(a) and (b) show that the vehicles are still able to avoid obstacles and keep formation at the same time after they converge back to the original formation pattern.

However, since a mechanical failure may occur when the vehicles try to manoeuvre, we have to consider the sequence of rules for formation convergence and avoiding obstacles. We specify that the vehicles should steer away the obstacle first and when they are all outside the minimal tolerable distance to the obstacle, they start to converge back to their original formation pattern. Figures 7.9 and 7.10 plot 18 snapshots of the status of vehicles at 18 time points.

Snapshot 1 5 vehicles depart from the starting line. The distance between two neighbouring teammates is 5m.



(a) The simulation trajectories with modified method with 1 obstacle.

(b) The simulation trajectories with the modified method with 3 obstacles

Figure 7.8: Simulation trajectories within different numbers of obstacles

Snapshot 2 Before vehicles face the obstacle danger, they keep the formation and cooperate to the target.

Snapshot 3 At time 26.3s, vehicle 5 has arrived within the minimum tolerable distance to the obstacle. Vehicles start to steer away the obstacle.

Snapshot 4 At time 27.0s, vehicle 2 fails due to some reason and it does not exist. Other vehicles realize the situation and consider to converge back to original formation pattern. In 3 dimensional environment, vehicle will drop to the sea floor if it fails. Here we assume vehicle 2 sank and did not block other vehicles.

Snapshot 5 Since vehicles are still within the minimum tolerable distance to the obstacle, they keep steering away from the obstacle rather then converge back to original formation.

Snapshot 6 At time 30.9s, all vehicles have escaped the danger of obstacle. They start to converge back to original formation.

Snapshot 7 At time 32.3s, vehicle 1 is steering towards the position of vehicles 2,3 and 4. Vehicles 2,3 and 4 steer towards the position of vehicle 1.

Snapshot 8 At time 33.3s, the distance between vehicle 1 and 3 is nearly 7.5m which is close to $\frac{3}{2}d_f$ (d_f is the distance between two neighbours in formation). Vehicle 1 starts to steer back to vertical. Vehicles 2,3 and 4 also start to steer back to vertical.

Snapshot 9 At time 34.0s, the distance between vehicle 1 and 3 is 5.049 which is roughly equal to d_f. Here, vehicles 1,3,4 and 5 have converged back to their original formation pattern. A new team has formed from these 4 vehicles.

Snapshot 10 The new team start to implement the main task of navigating to the target.

Snapshot 11 Vehicles steer towards the target.

Snapshot 12 At time 42.6s, vehicles meet the obstacle again and have to steer away for the second time.

Snapshot 13 Vehicles steer away the obstacle.

Snapshot 14 Vehicles keep steering away the obstacle.

Snapshot 15 When the danger of the obstacle has passed, the vehicles start to navigate to the target.

Snapshot 16 Vehicles steer towards the target.

Snapshot 17 Vehicles move forward to the target.

Snapshot 18 Vehicles arrive at the target.

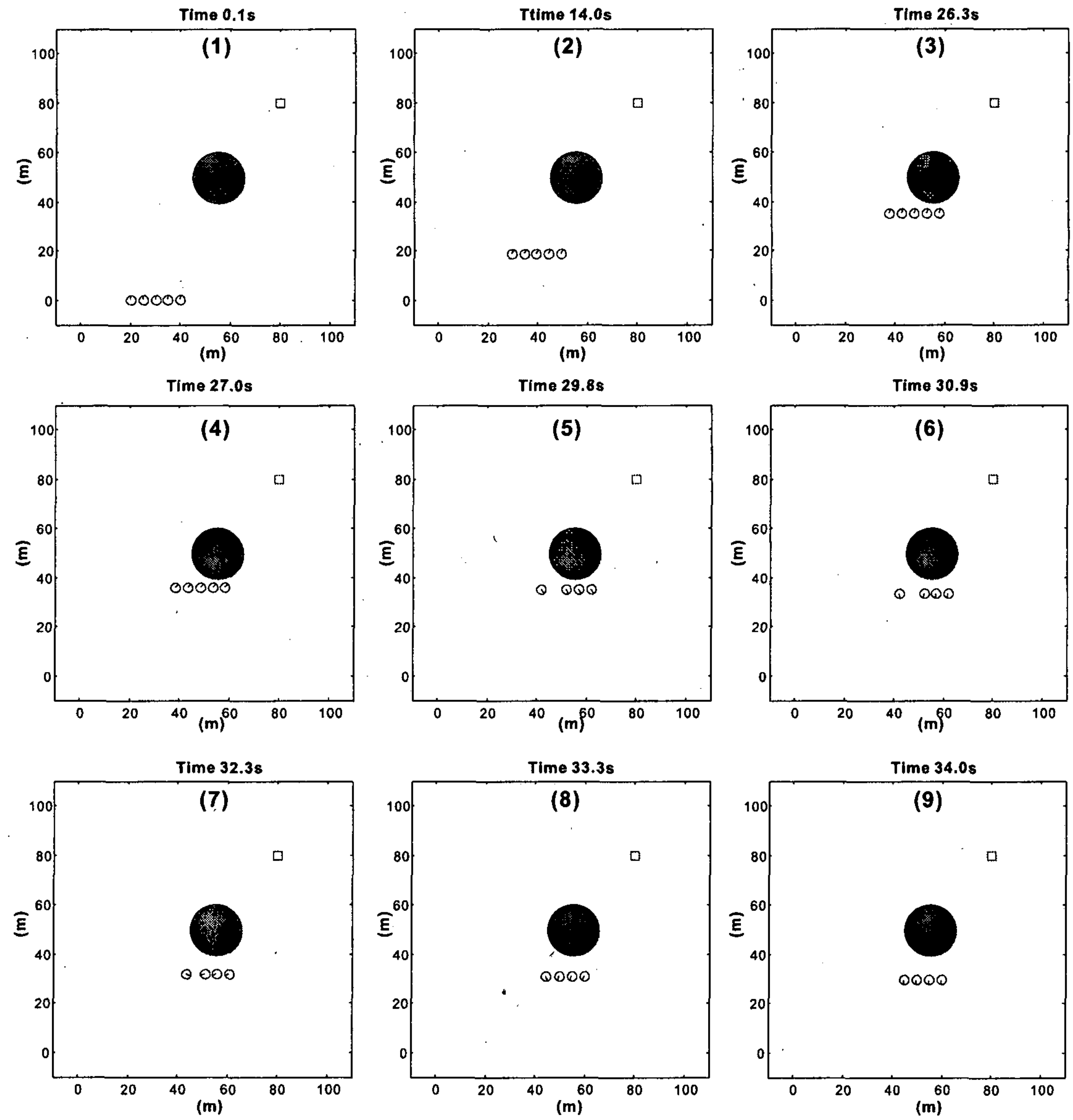


Figure 7.9: Snapshots 1-9 when one vehicle fails during the time that vehicles are steering away from the obstacle

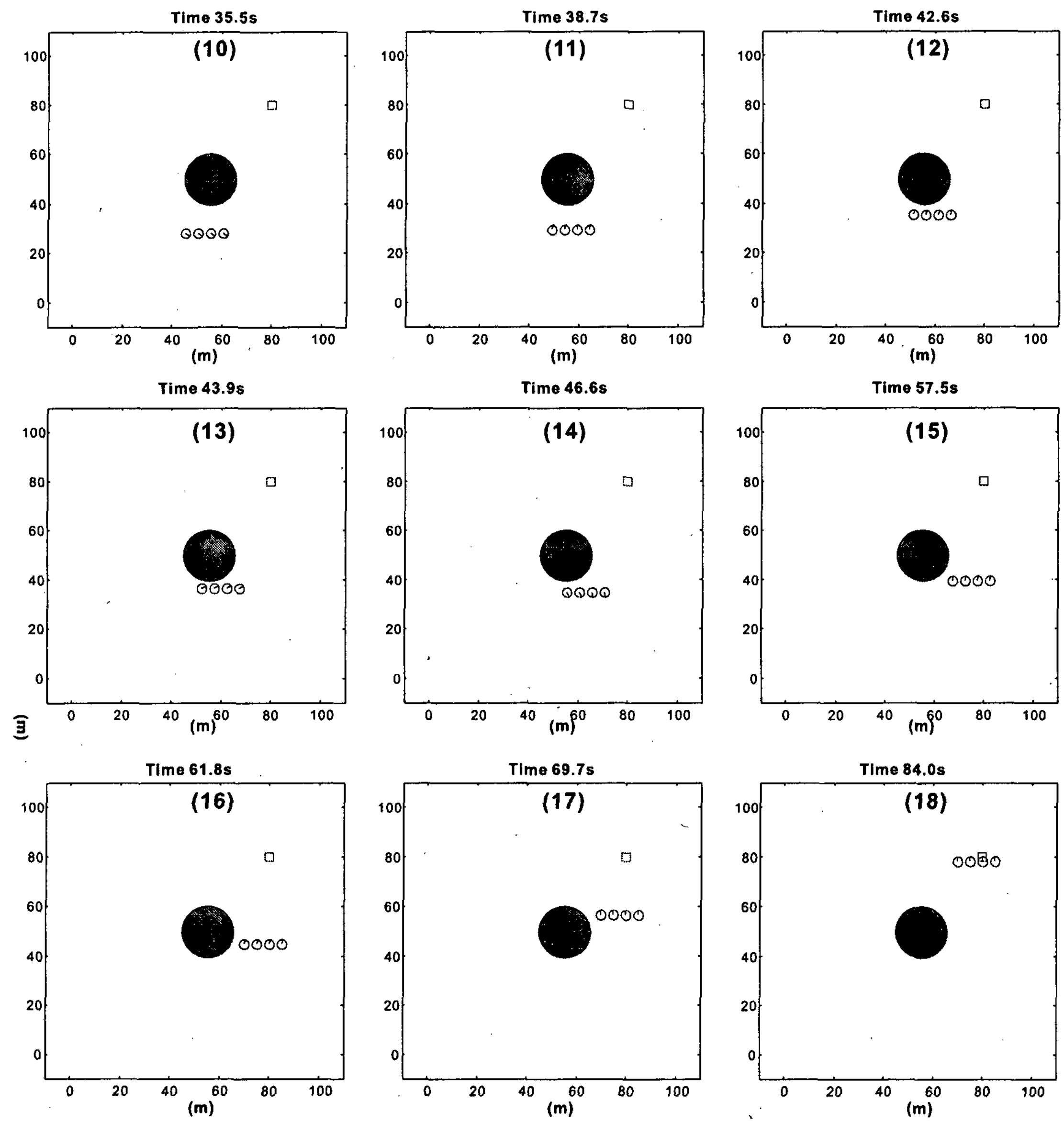
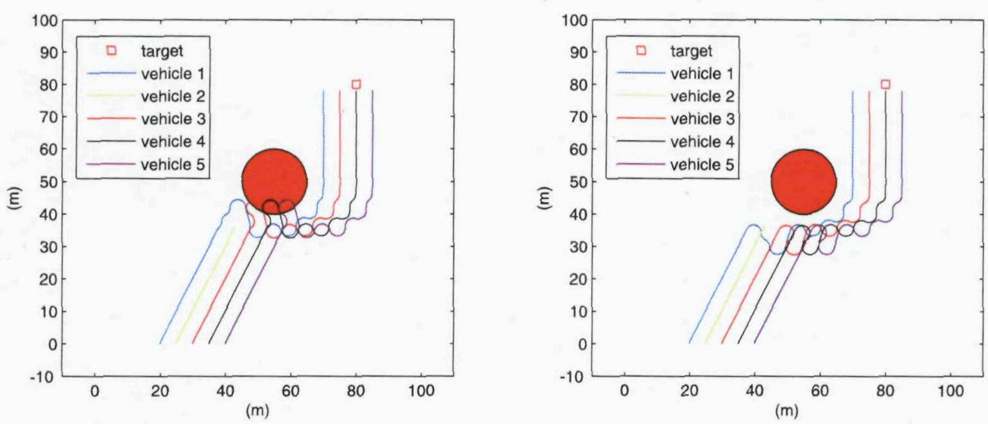


Figure 7.10: Snapshots 10-18 when one vehicle fails during the time that vehicles are steering away from the obstacle

Snapshots 3-9 mainly describe the motions of vehicles when one vehicle fails during the time that vehicles are within the danger range of obstacle. If vehicles still execute the convergence behaviour before avoiding obstacles, vehicles would crash with the obstacle. As shown in Figure 7.11(a), the trajectories of vehicles 2,3 and 4 cross the obstacle boundary which means that these vehicles have crashed into the obstacle. On the contrary, if we execute avoiding obstacles behaviour before the convergence behaviour, vehicles can not only avoid the obstacle, but also converge back to their original formation pattern safely. Simulation trajectories in Figure 7.11(b) confirm this behaviour.



(a) Convergence behaviour is a higher priority than collision avoidance
(b) Convergence behaviour is a lower priority than collision avoidance

Figure 7.11: Simulation trajectories when using different rule priorities when vehicle 2 fails during the time that vehicle are within the danger of obstacle.

7.3 Circle formation control

7.3.1 Circle formation problem statement

In a circle formation pattern, all vehicles do not locate on the circle but remain within the circle. The circle is predefined according to the sensor range of a vehicle, and typically the formation circle range would be less than the sensor range to ensure that vehicles in the circle can communicate with each other. For example, in Figure 7.12, L is the radius of the formation circle and R is the radius of the vehicle's sensor range. The largest distance between two teammates is up to $2L$. R must be 2 times larger than L ($2L \leq R$). Each vehicle can broadcast information to all vehicles within the circle. A smaller size of circle can decrease the sensor's transit distance so that it can increase the reliability of communications between vehicles. In some cases, sensor range may be dramatically decreased due to the complicated underwater environment and then the team circle size could be reduced to ensure that vehicles in the circle can communicate directly and effectively.

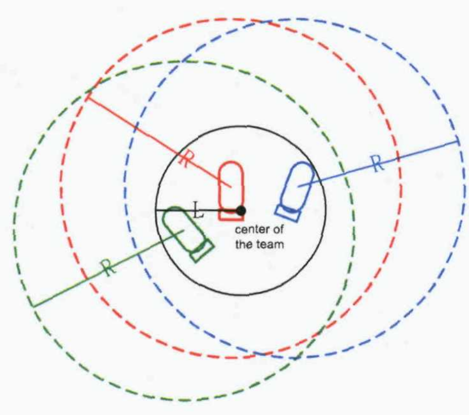


Figure 7.12: Three vehicles in a circle formation. R is the radius of the vehicle's sensor range. L is the radius of the formation circle. The centre point of circle is the average position of all team members.

For all vehicles to remain in a predefined circle range, we have two problems to solve:

1. When vehicles tend to go outside the circle, or are outside the circle, they must be able to turn back in to the formation circle.

2. When vehicles meet obstacles, they must keep within their circle and not split into two teams. This is because vehicles may not be able to communicate with each other after breaking into different teams.

In addition to 2L≤R noted above, we assume $L_{mo} > L_{mt}$, where L_{mo} is the minimum tolerable distance between vehicle and obstacle and L_{mt} is the minimum tolerable distance between vehicles. This assumption ensures that vehicles are closer to their neighbours than to obstacles. Therefore the chance of team separation is decreased when vehicles manoeuvre around obstacles.

7.3.2 Behaviour rules for circle formation control

The behavioural control algorithm consists of the following rules:

1. Rule 1: matching velocity of the teammates

2. Rule 2: navigating to the target

3. Rule 3: avoiding collision between teammates

4. Rule 4: avoiding obstacles

5. Rule 5: maintaining a circle formation

We can see from the list that Rules 1-4 are the same as the rules we described in the previous chapter. The reason we do not use tracking the centre of team rule is because vehicles will be kept in the team range by the circle formation rule. These rules will maintain vehicles in a team and cooperatively navigating to the target. Rule 5 (maintaining a circle formation) is an additional rule which aims to ensure that vehicles keep in a predefined circle range. The details of rules 1-4 and the generation process of fuzzy priority weights are described in the previous chapters.

Rule 5: keeping circle formation

This rule is very similar to the remaining within an area rule. The circle range can be viewed as an invisible boundary. In Figure 7.13, when vehicles reach close to the circle boundary, they use steer angle β_5 to move back towards the centre of the circle.

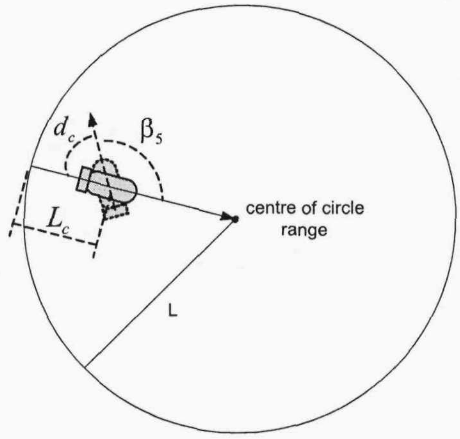


Figure 7.13: Rule 5: keeping circle formation

The steering angle factor β_5 can be calculated by equation 7.3.1.

$$\beta_5 = \arctan(\frac{py_{centre} - py_{current}}{px_{centre} - px_{current}}) \tag{7.3.1}$$

where β_5 is the steering angle factor calculated from rule 5, $(px_{centre}, py_{centre})$ is the average position of all teammates in the reference frame of coordinates.

In order to control how vehicles move away the circle boundary and back to the centre, we use again the fuzzy logic controller to estimate the priority weight value according to the distance from vehicle to circle range L_c and the angle difference d_c between the current velocity direction and the direction away the centre of the circle as would be calculated from the vehicle heading sensor and an estimate of the circle centre computed from the position of teammates.

Two linguistic input variables L_c and d_c are declared. L_c has two linguistic values *short* and *long*. The value *short* is given as $[0\ L_{mc}]$. L_{mc} is the minimum tolerable distance between a vehicle and the circle boundary. The value *long* is given as $[L_{mc}\ L]$. L is the radius of circle. d_c has two linguistic values *same* and *opposite*. The range of value *same* is given as $[0\ \frac{\pi}{2}]$; the range of value *opposite* is given as $[\frac{\pi}{2}\ \pi]$. The membership function of the two input variables is similar to the membership function shown in Figure 5.8. w_5 is the output variable declared with four linguistic values, *very low*, *low*, *high* and *very high*. The membership functions of them are similar to the membership function in Figure 5.4.

The following fuzzy rules represent the function to decide upon the weight value.

IF L_c *is short* and d_c *is same* THEN w_5 *is very high*;
IF L_c *is short* and d_c *is opposite* THEN w_5 *is high*;
IF L_c *is long* and d_c *is same* THEN w_5 *is high*;
IF L_c *is long* and d_c *is opposite* THEN w_5 *is low*;

Combination of fuzzy behaviour rules

We introduce an extension of the method by generating a new priority classification scheme. A priority weight value is assigned to each behaviour rule. Basically, as shown in Figure 7.14, the rules are classified to class and subclass level. Each level of classification presents a fuzzy logic generation process. Collision and obstacle avoidance rules are still in the higher priority, class 1, than rules 1,2 and 5 in class 2 since rules 3 and 4 take responsibility to avoid danger of colliding with obstacles and neighbours. In this class level, the priority weight value sets are calculated by the fuzzy logic controller F1 (see Figure 7.14). Furthermore, the rules within class 2 are classified again into 3 subclasses and their priority weights will be calculated by the fuzzy logic controller F2. In subclass level, the circle formation control rule has the highest priority to ensure that vehicles stay within communication range of one another. The navigation rule is in subclass 2 since it is in charge of the mission goal. Matching velocity is in the lowest subclass 3.

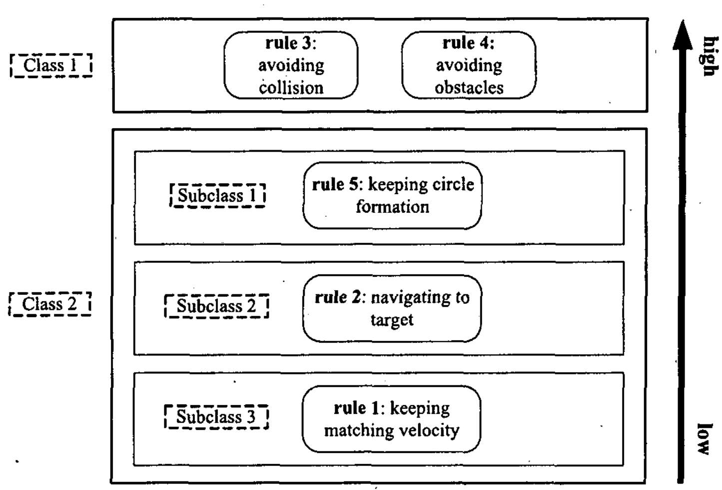


Figure 7.14: The classification of behaviour rules in terms of priorities

Two generation process results are combined together to get final priority weight values. The generation process is shown in figure 7.15.

In the fuzzy logic controller F1, w_3 and w_4 are linguistic input variables; w_1, w_2 and w_5 are linguistic output variables. The membership function of each priority weight w_i is referred to the description in the previous chapter.

The following fuzzy rules represent the function to decide the weight values.

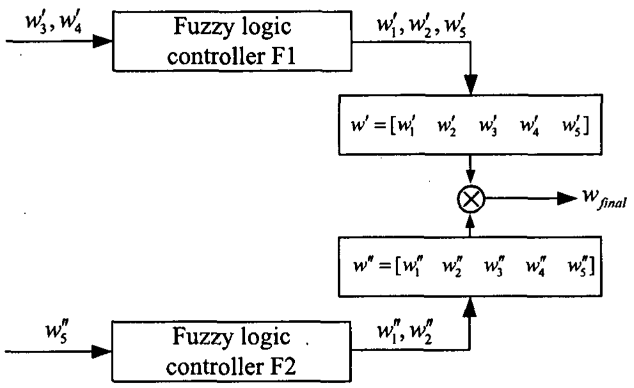


Figure 7.15: The generation process of priority weights by two fuzzy logic controllers

IF *weight 3 is very high* or *weight 4 is very high*
THEN *weight 1,2,5 are very low*;
IF *weight 3 is very low* and *weight 4 is very low*
THEN *weight 1,2,5 is low*;
IF *weight 3 is high* or *weight 4 is high*
THEN *weight 1,2,5 is very low*;
IF *weight 4 is low* and *weight 5 is low*
THEN *weight 1,2,5 is high*;

After this fuzzy logic controller F1, we get the first set of priority weight by combining the inputs and outputs of F1. The weight set is expressed as vector 7.3.2:

$$w' = [\ w'_1 \quad w'_2 \quad w'_3 \quad w'_4 \quad w'_5\ ] \tag{7.3.2}$$

In the fuzzy logic controller F2, w_5 is the linguistic input variable; w_1 and w_2 are linguistic output variables.

The following fuzzy rules represent the function.

IF *weight 5 is very high*
THEN *weight 1 is very low, weight 2 is low*;
IF *weight 5 is very low*
THEN *weight 1 is high, weight 2 is very high*;
IF *weight 5 is high*
THEN *weight 1 is very low, weight 2 is low*;
IF *weight 5 is low*
THEN *weight 1 is low, weight 2 is high*;

The vector of priority weights is expressed as:

$$w'' = [\ w_1''\ \ w_2''\ \ w_5''\ ] \tag{7.3.3}$$

Since w_1', w_2' and w_5' in w' are the same, we use w'' as proportion of them to multiply with w'. Therefore, each weight value is vector 7.3.3 is scaled down to:

$$w'' = [\ {}^{w_1''}\!/_{\max(w'')}\ \ {}^{w_2''}\!/_{\max(w'')}\ \ {}^{w_5''}\!/_{\max(w'')}\ ] \tag{7.3.4}$$

Then we manually set $w_3 = 1$ and $w_4 = 1$ and add in the vector w''. This setting will not give effects to w_3' and w_4' when multiplication.

$$\begin{aligned} w'' &= [\ w_1''\ \ w_2''\ \ w_3''\ \ w_4''\ \ w_5''\ ] \\ &= [\ w_1''\ \ w_2''\ \ 1\ \ 1\ \ w_5''\ ] \end{aligned} \tag{7.3.5}$$

Finally, the final weight vector is equal to w' multiplied by w'':

$$w_{final} = w' * w'' \tag{7.3.6}$$

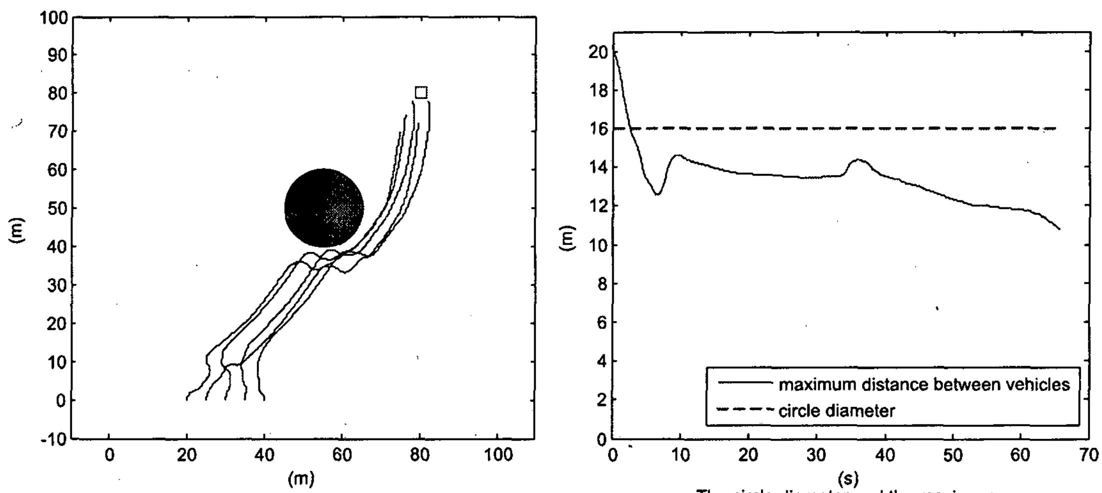
For example, when the vehicle is facing an obstacle, F1 generates weights w' such as [0.2 0.2 0.1 0.8 0.2] in which here rule 4 has the highest priority. In the meantime, this vehicle also needs to implement rule 5 to maintain a circle formation. Then F2 generates weights w'' as, for example, [0.1 0.1 0.8]. According to equation 7.3.4, w'' is scaled to [0.125 0.125 1]. Next according to equation 7.3.5, w'' is extended to [0.125 0.125 1 1 1]. According to the equation 7.3.6, w_{final} is [0.025 0.025 0.1 0.8 0.2]. From w_{final}, we can see that obstacle avoidance rule has the highest priority and formation keeping rule has the second highest priority.

The five steering angle factors are then multiplied by their relative weights and are combined together to calculate the next desired steering angle by equation 4.3.1.

7.3.3 Simulation results

The approach is again tested in a simulation scenario without the tidal flow in which vehicles depart from their initial positions and navigate to the target whilst maintaining a predefined circle formation. The circle range can be adjusted in the simulations in order to verify how small a circular range the vehicles can maintain. In addition, obstacles can be added to the simulation to complicate the environment. The initial positions of vehicles are defined as a line and as random positions in a circular range. This allows us to examine how different initial positions affect the approach. The minimum tolerable distance to the circle boundary L_{mc} is specified here as 4m and is used in design of the fuzzy logic controller for maintaining the circle formation. Other parameters are given in the Table 4.2 in chapter 4, together with a description of the dynamics of a UUV, SUBZERO, which forms the basis of each UUV in the team.

First, 5 vehicles depart as a line and the distance between two neighbouring vehicles is 5m. Simulation results with different formation circle radius within an 1 obstacle environment are shown in Figures 7.16 and 7.17. The circle radius is 8m in Figure 7.16. Since the initial size of team is 20m, the vehicles first must regroup within the circle. From the right subfigure, we can see that vehicles took about 2s to regroup within the circle diameter. As vehicles reached the minimum distance between them at time 6s, they then increased the distance between each other because the collision avoidance rule was the main contribution. At time 37s, distance between vehicles increased because they met an obstacle and the obstacle avoidance rule priority is raised. Subsequently, the maximum distance between vehicles became less and less. From the left subfigure, we can see the simulation trajectories which visually show how the vehicles progressed to the target. Then we decrease the circle radius to 7m

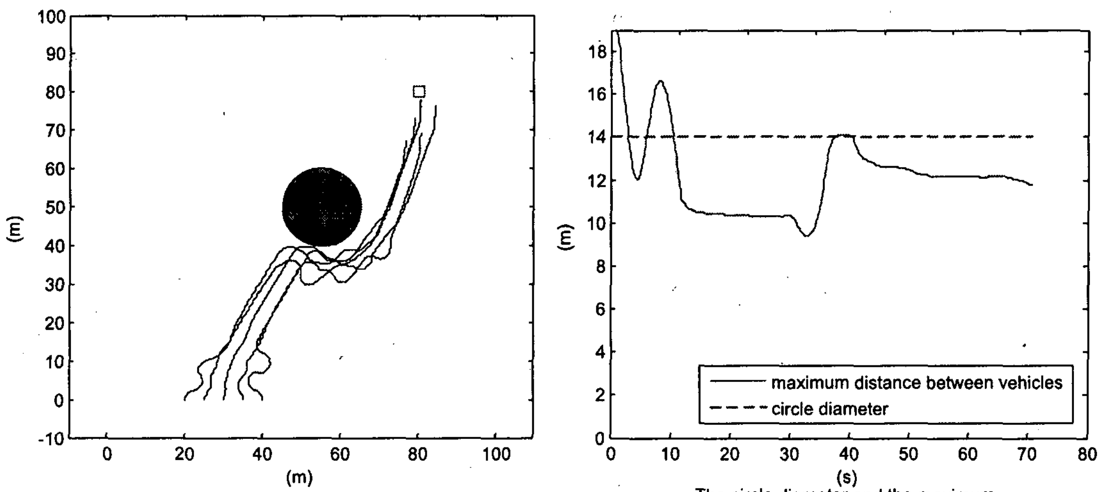


(a) Simulation trajectories when circle radius is 8m

(b) The circle diameter and the maximum distance between vehicles over time

Figure 7.16: Simulation results when circle radius is 8m within a 1 obstacle environment. Vehicles depart as a line.

and the simulation results are shown in Figure 7.17. Although we can see that vehicles succeeded in achieving the goal, the maximum distance between vehicles rose twice during the time around 10s and 38s. The reason for that is because the collision and obstacle avoidance rules navigated vehicles out of the circle range. From the above simulation results, we can conclude that the circle radius cannot be defined as small as we may wish, but depends on various parameters, such as the minimum distance between two neighbouring vehicles, vehicle forward speed and the minimum distance between a vehicle and an obstacle. For example, in the simulations case above, the minimum circle radius which can maintain the vehicles in the circle range is roughly 8m. A larger size of the circle will give much more chance to maintain vehicles in the circle formation. However, as the consequences, larger circle size causes more communication difficulties.

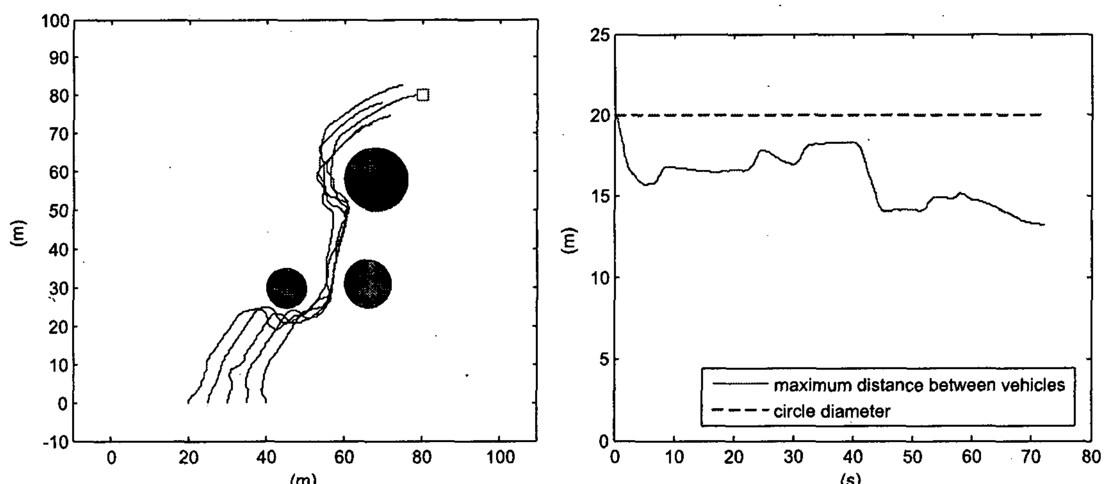


(a) Simulation trajectories when circle radius is 7m

(b) The circle diameter and the maximum distance between vehicles over time

Figure 7.17: Simulation results when circle radius is 7m within a 1 obstacle environment. Vehicles depart as a line.

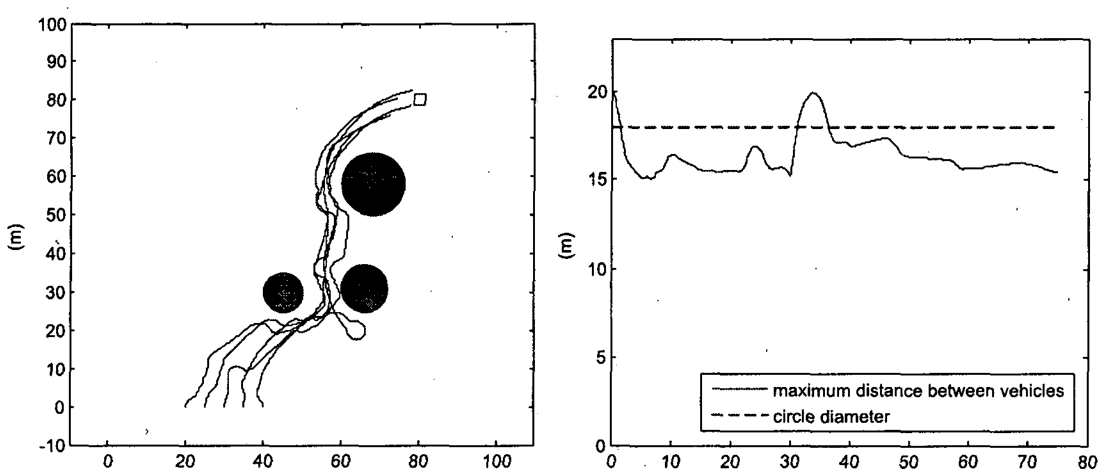
Next, we add more obstacles into the environment and obtain simulation results shown in Figures 7.18 and 7.19. In Figure 7.18, the circle radius is defined as 10m. We can see that the solid line of maximum distance between vehicles is very close to the dashed line of circle diameter. This is because that obstacle avoidance rule mainly contributes to the steering decision as soon as vehicles are facing the danger of obstacles. In Figure 7.19, we can see that there were vehicles out of range at the peak about 34s. Therefore, in this case, the minimum circle radius which can ensure vehicles in the circle range is 10m. Compared to the simulation results within a 1 obstacle environment, we can see that the minimum circle radius must increase when more obstacles present. From the comparison, we can conclude that the complexity



(a) Simulation trajectories when circle radius is 10m

(b) The circle diameter and the maximum distance between vehicles over time

Figure 7.18: Simulation results when circle radius is 10m within a 3 obstacle environment. Vehicles depart as a line.



(a) Simulation trajectories when circle radius is 9m

(b) The circle diameter and the maximum distance between vehicles over time

Figure 7.19: Simulation results when circle radius is 9m within a 3 obstacle environment. Vehicles depart as a line.

of environment is another major influencing factor on circle formation control.

After we simulate vehicles departing from a line, we give the initial positions of vehicles as random positions which are close to each other, as shown in Figure 7.20. In this case, vehicles are already within the circle range. They do not need to regroup as the vehicles with initial line positions. Figures 7.21 and 7.22 show the simulation

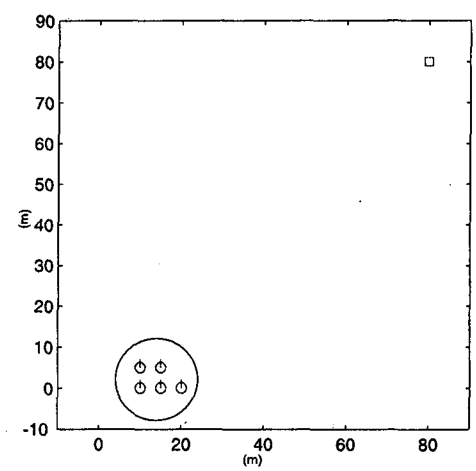
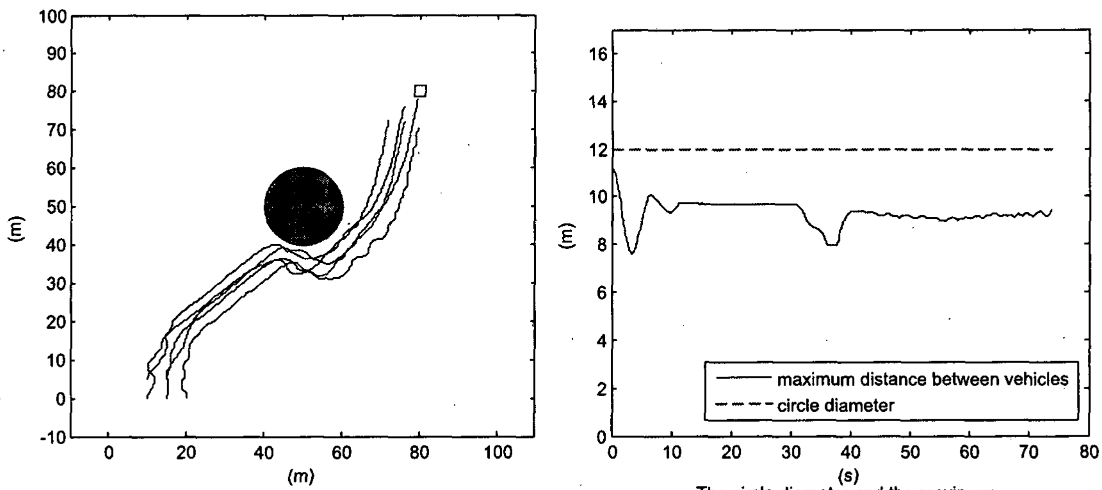


Figure 7.20: Vehicles depart from their random positions which are already within the circle.

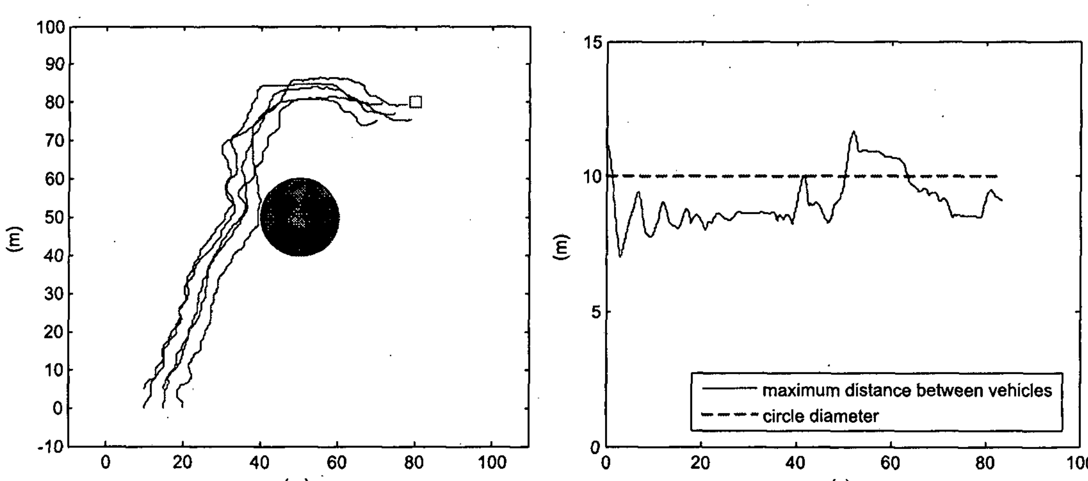
results with different circle radii within a 1 obstacle environment when vehicles depart from random positions. In Figure 7.21, the circle radius was 6m. Vehicles succeeded in keeping within the circle range and reached the target. In Figure 7.22, the circle radius was 5m. Vehicles failed to keep everyone within the circle range at time 50-70s. However, if we compare results with Figures 7.16 and 7.17, we can see that under the same environment the size of circle range can be smaller when initial positions of vehicles are close to each other. In this case, the minimum circle radius is 6m. From this point, we can conclude that the initial positions of vehicles can affect the feasibility of circle formation control.



(a) Simulation trajectories when circle radius is 6m

(b) The circle diameter and the maximum distance between vehicles over time

Figure 7.21: Simulation results when circle radius is 6m within a 1 obstacle environment. Vehicles are already within the circle range when departure.

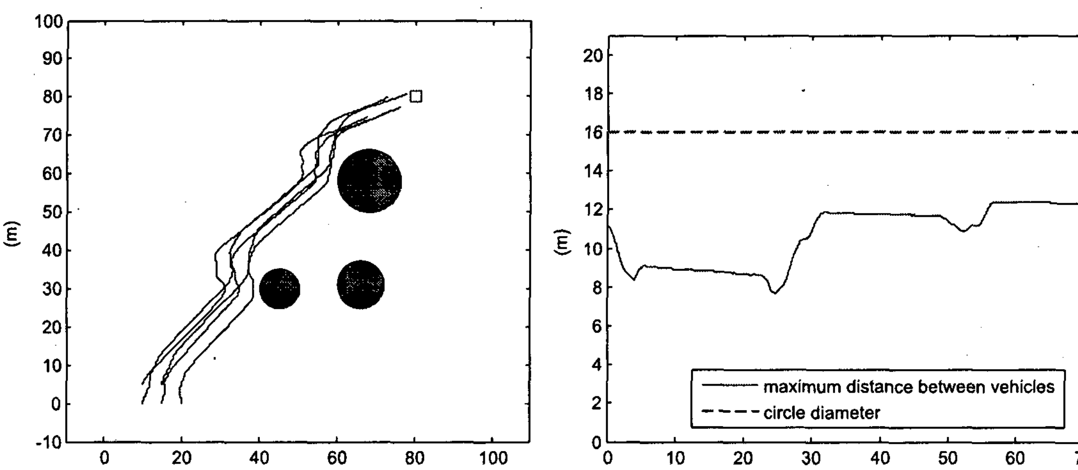


(a) Simulation trajectories when circle radius is 5m

(b) The circle diameter and the maximum distance between vehicles over time

Figure 7.22: Simulation results when circle radius is 5m within a 1 obstacle environment. Vehicles are already within the circle range when departure.

Finally, we simulate the situation that vehicles are already within the circle range when departure as shown in Figure 7.20 in a 3 obstacles environment. The simulation results in figures 7.23 and 7.24 show the successful simulations. In this case, the minimum circle radius is 6m. The reason why the minimum circle radius is quite small is because the vehicles did not meet any obstacles during their entire trajectories and hence did not have to act to steer away. However, we can deduce that the minimum radius will be greater than 6m if they have to steer to avoid an obstacle. From another point of view, this tells us that if we have no advance knowledge about the environment, we must consider the presence of obstacles in the environment when defining the circle radius.

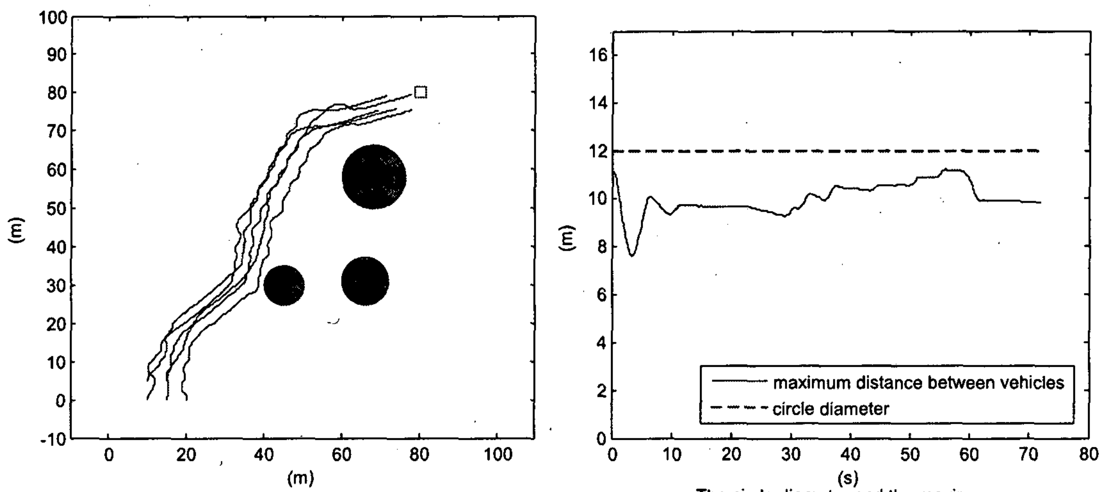


(a) Simulation trajectories when circle radius is 8m

(b) The circle diameter and the maximum distance between vehicles over time

Figure 7.23: Simulation results when circle radius is 8m within a 3 obstacle environment. Vehicles are already within the circle range when departure.

To sum up, the circle diameter is affected by the obstacle number and the vehicles' initial positions. If the obstacle number increases, the circle diameter must be increased. If the vehicles' initial positions are already within the circle, the circle diameter could



(a) Simulation trajectories when circle radius is 6m

(b) The circle diameter and the maximum distance between vehicles over time

Figure 7.24: Simulation results when circle radius is 6m within a 3 obstacle environment. Vehicles are already within the circle range when departure.

be decreased. Besides these parameters, vehicle size and speed , the minimum distance between two neighbouring vehicles and the distance between a vehicle and an obstacle could also affect the circle diameter.

7.4 Conclusion

The approach has shown promise for formation control of a cooperative UUV team. However, the mission is affected by several parameters, such as circle diameter, initial positions, vehicle size and speed, the minimum distance between two neighbouring vehicles, the minimum distance between a vehicle and an obstacle and the complexity of environment, etc. In a real UUV system, these parameters would be determined by sensors and other mechanical designs considerations and we must consider these parameters when we design a control system for a team of cooperative UUVs.

Chapter 8

Conclusions and Future plan

8.1 Conclusions

This thesis has investigated an intelligent, behaviour-based approach to control a team of collective autonomous underwater vehicles. The approach implies the distributed control strategies. Inspired by social animal societies and their collective behaviour, this approach exploits vehicle-vehicle and vehicle-environment interactions to develop goal-oriented and emergent collective behaviour.

The main contributions of this thesis are:

1. Literature reviews of social animal societies and their collective behaviours and a comparison between single vehicle and multiple cooperative vehicle applications.
2. A behaviour-based approach to control a team of cooperative unmanned underwater vehicles. The vehicle's steering decision is made by combing a set of behaviour-based rules with their relative priority weights. We apply the fuzzy

logic method to adapt the relative priority weight values according to the situation the vehicles meet.

3. Validation of the intelligent behaviour-based approach in different simulations, such as applications in a waterflow environment and in formation control.

In chapters 2 and 3, by discussing and comparing the design, cost, reliability of the current and possible applications of a single UUV and a team of UUVs, we can conclude that a team of UUVs could be more efficient than a single UUV in some missions, such as environmental monitoring and pollution surveillance. From the comparisons, we can clearly see that using a team of cooperative UUVs can not only bring more advantages than using a single UUV, but also overcome some of the disadvantages. In practice, the design of a team of UUVs could be much easier. Each vehicle may not need to carry any high specification sensors and complex control systems, but just duplicate a simplified system since all vehicles are the same. Even if each vehicle is small and low cost, the entire system cost could be possibly high when vehicle numbers increase. However, the risk of mission failure by losing vehicles could decrease when vehicle numbers increase and mission achievement is more likely. From another point of view, we cannot deny the necessity of using a single UUV for some applications. We should make the choice between a single UUV and a team of UUVs in terms of the requirement of the mission.

Following the UUV introduction, we have reviewed ants, bees and fish collective behaviours and compared them with cooperative actions of a team of UUVs. From this, we concluded that we can learn from animal collective behaviours and apply bio-inspired behaviour-based methods for control a team of UUVs. In order to link biological behaviours and team UUVs cooperative actions together, we have compared the requirements of the communication mode when both collective animals and cooperative UUVs are working as a team, in particular between fish and UUVs since they both operate in a complicated underwater environment. From the description of fish schooling and the comparison between it and a cooperative UUV team, we can conclude that the advantages of fish schooling may also benefit the mission implementation of a team of cooperative UUVs, and a fish school could provide a clue for controlling a team of UUVs by behaviour-based rules.

In chapter 4, we have used a behaviour-based method with fuzzy logic controlled priority weights. The basic method is inspired from **Boids** approach. We have demonstrated that behaviour-based method inspired from **Boids** can not only use for computer animation simulations, but also for cooperative control mechanism for a team of cooperative vehicles. By simulating a mission scenario, we can see that the behaviour-based rules can satisfy the requirements of a mission even with the dynamic model of SUBZERO. By comparing results with different weights setting, we can conclude that high priority rules with high weight values and low priority rules with low weight values can give a steering direction decision to ensure vehicles achieve the goal without collision with obstacles and neighbours. By analyzing the simulation results with different numbers of vehicles, we can give the practical design a suggestion that the minimum tolerable distance to obstacles must increase when the number of vehicles increases.

However, although the constant priority weight settings can avoid obstacles and neighbours all the time, they cannot be adapted to fit changes in situations. In chapter 5, we have applied an artificial intelligent method, fuzzy logic, to estimate the priority weight according to the situation that the vehicles meet on-line. The simulation results under a mission scenario have shown that fuzzy logic controllers have the abilities to make intelligent decisions for emergent behaviours. From the variation of the weights over the entire travel time, we can see that the weights are adapted to fit the situation by the fuzzy logic controllers. For example, the weight for the obstacle avoidance rule becomes greater when the vehicles approach an obstacle. The comparison results with different setting of parameters L_{mt} and L_{mo} have shown that the travel time could be minimized by the adjustment of these parameter settings. Consequently, in practice, parameter optimization would provide some beneficial solutions for economic use of an onboard energy supply. In addition, by comparing the results with and without a dynamic model of a vehicle, we can conclude not only that the control systems work with dynamics, but also that the dynamics can help to save travel time.

In chapters 6 and 7, we have extended the method to fit the cases when vehicles are in a tidal flow environment and when vehicles are in a formation pattern. The verification in different situations showed that the methods are adaptive to fit the variable requirements under complicated situations.

First, we have verified the behaviour method for control a team of underwater vehicles in a very common environment with water flow effects. The simulation results have shown that the modified line of sight navigation method can offset the effect of waterflow on the velocity direction of the vehicles and decrease the travel time, compared with normal line of sight navigation. In addition, the behaviour-based method has been successfully used for simulations in an environment with obstacles.

Second, we have assessed the behaviour method for line and circle formation control a team of vehicles. The particular behavioural rules are designed to position the vehicles within the specified formations. Under the assumptions of the specific geometric configuration of the circle and line formations, and the relationships between the teammates in the formations, we have simulated the cooperative mission scenarios with both formation patterns and the results showed that the approach has potential to help the cooperative vehicles maintain a specified formation.

Both case studies have demonstrated that the behaviour-based method with fuzzy logic intelligent decision making can be adaptively used for different situations and under different conditions. Through the simulations, the results have given some indications as to how to set the parameters. In practice, this would be of considerable benefit for applications involving the design of control schemes for teams of real UUVs.

However, since the results are based on computer simulations, we have to consider the system design if the real system of multiple UUVs and environment do not match the assumptions. We list two examples:

- The simulation results are really dependent on the parameter settings and the simulation environment which, in practice, would be variable. For example, the velocity and direction of water flow in a waterflow environment varies under different climatic and geographical conditions, e.g. tide flows change with time. The vehicles must be able to quickly respond to changes in situation to ensure that they achieve their goal. Adapting the behaviour-based method and the intelligent decision making to fit variable situations would be a further challenge for the future.
- Sensor detecting system and communications technologies could be another challenge for real situations. In the previous simulations, we have not considered the sensor systems and communication and assumed that the vehicles can gather any local information they need, such as obstacle positions in the local environment and teammate velocities and positions. However, due to the constraints in a underwater environment, sensing and communication systems will become complicated and sometimes even unreliable. We must consider how vehicles make decisions when the local information is poor. This problem will be further addressed in the future work.

To conclude, we expect that the cooperative approach developed here will contribute to reinforcing the weak link between studies of social animals, collective autonomous underwater vehicles and artificial intelligent algorithms. In particular, we believe that the behavioural rules with priority weight modelling have considerable potential for cooperative underwater vehicle applications.

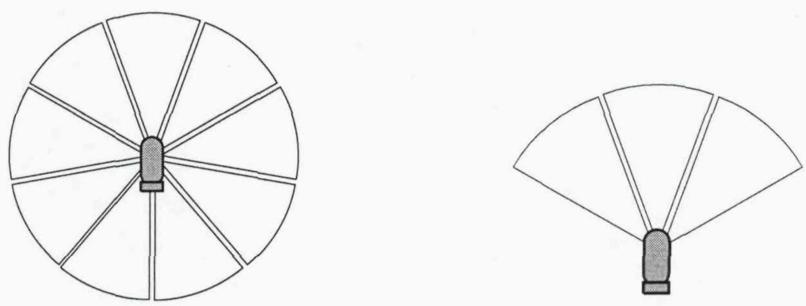
8.2 Suggestions for future work

8.2.1 Communications

In previous simulations, communications are assumed to be perfect and the vehicles can get all the information they need. Vehicles can know their positions at any time and also the positions of their teammates. Sensors are assumed to be able to detect the velocity of teammates and of the vehicle itself. However, sensors and communications have practical limitations in the complex underwater environment, such as time delays, sensing range limit, power cost and size. Finding the best communication equipment and a communication network strategy is a difficult task and will be addressed in future studies using SUBZERO.

Communication equipment

Communication modes can be classified into two categories. The first one is the sensors which only collect local information and do not exchange information with other vehicles, such as sonar and underwater vision cameras. These sensors collect information independently. However, sonar and vision sensors have strong directional characteristic. In all the simulations, detecting sensor system in Figure 8.1(a) are assumed to have no blind area. In future work, we will consider how to design the sensors on the vehicle to overcome the problems posed by blind areas. Multiple sensors or sensor-arrays can help to enhance the covering range. Furthermore, vehicles may not able to carry more sensors due to the limitations in their power supply. We need to investigate how the control algorithms work when the sensors cannot cover all directions, such as the sensor in Figure 8.1(b) which only covers a 120 degree angle. In the previous simulations, the effect of different sensor ranges has been discussed and the minimal sensor range has been suggested by the simulation results. In practice, we would have to ensure that sensors can satisfy such requirements.



(a) Sensor-array covers a 360 degree angle in simulation

(b) Sensor covers only a 120 degree angle in practice

Figure 8.1: Sensor covering range in simulation and in practice

Communication between vehicles is also necessary in order to exchange the information. Digital acoustic communication has been developed and is becoming used in autonomous underwater vehicles. In the previous simulations, we assumed that a vehicle can exchange information with its teammates and know exactly where the teammates are and their velocities. However, acoustic sensors cannot be easily deployed underwater due to some major challenges (Akyildiz et al. 2004), such as:

- Limited bandwidth
- Multi-path and fading underwater channel

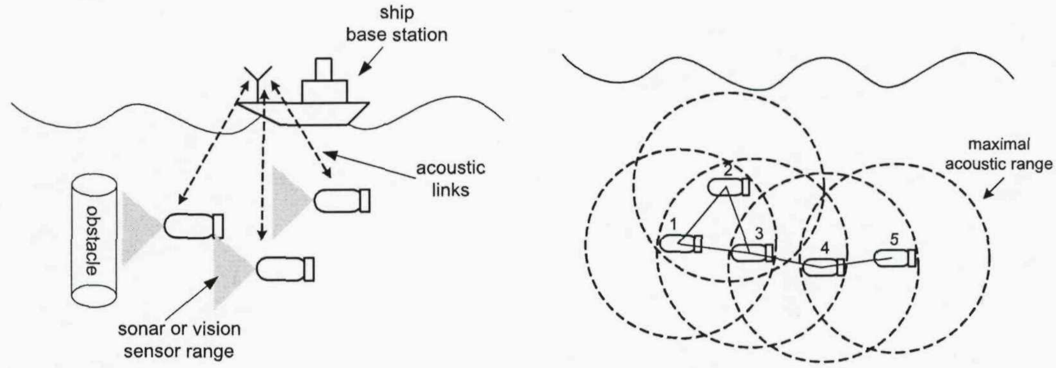
- Propagation delay in underwater
- High bit error rates and temporary losses of connectivity
- Limited power supply

It will be necessary in future work to determine whether the control algorithm works satisfactorily when acoustic communication has such challenges. For example, if there are high bit error rates and temporary losses of connectivity, vehicles may lose information when they need to decide upon the next step. They must have other communication methods to offset the problem, such as sonar and vision sensors. Moreover, algorithms maybe need to be adjusted to make them robust and reliable when there is transmission time delays or bit errors.

Communication network strategy

Communication network strategy may offset the limitation of the communication sensors and reduce the technical requirement of the sensors. For example, communication network could cover a larger range than a single communication sensor. Even if the available communication equipment has a shorter detecting range, we can still achieve the desired detecting range by using multiple such sensors to build a communication network. There are two network types which can be used for different missions. We will suppose that these two network strategies will be used in future simulations.

One strategy (Figure 8.2(a)) is to use a ship or other equipment on the surface as the base station. All information from the vehicles are routed by the base station and sent to the appropriate destination. The sonar and vision sensors take responsibility for collecting the information about obstacles and targets with a uncertain position, such as a moving target. This type of network is good when a mission is only implemented within a relatively small area. The ship or other base station equipment must have the ability to cover this area.



(a) Ship as base station transmits information for all vehicles

(b) Ad-hoc network strategy.

Figure 8.2: Two types of communication network strategies

Another strategy is to used an Ad-hoc network (Sozer et al. 2000), as shown in Figure 8.2(b), in which communication does not require a centralized station for routing vehicle information. In other words, each vehicle can act as a router, if needed. When some vehicles are out of acoustic range from other vehicles, routing might be the only enabling solution. This significant advantage may broaden the range of the team. Figure 8.2(b) shows a scenario with five vehicles with their maximal acoustic range circles. Vehicles 1,2 and 3 are all reachable by each other, and point-to-point communication would suffice. Vehicle 5 must require vehicle 4 to route messages from 1,2 and 3. Routing of communications should be investigated in the next stages of the project.

8.2.2 Implementation on real UUVs

In chapter 4, we have described a small UUV, SUBZERO, which forms the basic model of each UUV in the simulations. However, this does not mean that the method developed can be used directly on real UUVs. There are several problems to be considered when we design the executable method for real UUVs.

- Based on the discussion in the last section, we must choose suitable sensors and communication equipments from the current market. In the mechanical design, we need to consider the architecture of sensor system and communication equipment on the real UUV. For example, in the description of the obstacle avoidance rule in the previous chapters, the distance from a vehicle to an obstacle is calculated from the centre of the vehicle to the edge of the obstacle. In practice, however, the distance should start from the position of the sensor which can detect the edge of the obstacle. Hence the structure and layout of sensors and communication equipments could be a large issue for the implementation of the rules.

- In chapter 2, we discussed the benefits of low-cost, small UUVs for underwater applications. In particular, in a situation where a few vehicles are lost due to poor communication, the mission still can continue to execute effectively by the rest vehicles. The overall system cost and risk of the mission are minimised by a distributed system. However, fully assembling adequate detecting and communication equipment, and the behaviour-based rule controllers, into a small UUV and seamlessly work with dynamic control modules would be another significant issue to be paid attention in vehicles design.

References

M. Adioui, O. Arino, W. Smith, and J. Treuil. A mathematical analysis of a fish school model. *Journal of Differential Equations*, 188(2):406–446, 2003.

M. A.Kovacina, D. Palmer, G. Yang, and R. Vaidyanathan. Multi-agent control algorithms for chemical cloud detection and mapping using unmanned air vehicles. In *Proceedings of the 2002 IEEE International Conference on Intelligent Robots and Systems(IROS)*, October 2002.

I. Akyildiz, D. Pompili, and T. Melodia. Challenges for efficient communication in underwater acoustic sensor networks. *ACM Sigbed Review*, 1(2), 2004.

R. Allen. Bioinspired ideas for uuv control and sensing. *The 7th Unmanned Underwater Vehicle Showcase, The National Oceanography Centre*, September 2005.

M. Asada, H. Kitano, I. Noda, and M. Veloso. Robocup: Today and tomorrow - what we have learned. *Artificial Intelligence*, 2(110):193–214, June 1999.

R. Babb. Scientific surveys in the deep ocean using auv: the autosub project. *AUV's 93*, 7(3), June 1993.

I. L. Bajec, M. Mraz, and N. Zimic. Boids with a fuzzy way of thinking. *Proceeding of ASC 2003*, pages 58–62, 2003.

T. Balch and R. Arkin. Behavior-based formation control for multirobot teams. *IEEE Trans. Robot. Automat.*, 14:926–939, Dec. 1998.

T. Balch and R. Arkin. Motor schema-based formation control for multiagent robot teams. In *Proceedings of the First International Conference on Multiagent Systems (ICMAS'95)*, 1995.

J. Batlle, P. Ridao, M. Carreras, and V. Ila. Toward cooperative underwater robots for ecological surveillance of marine and underwater environments. In *3rd Workshop on European Scientific and Industrial Collaboration, WESIC 2001*, Enschede, The Netherlands, 27-29 June 2001.

O. Bayazit, J.-M. Lien, and N. M. Amato. Roadmap-based flocking for complex environments. *Proceedings of the 10th Pacific Conference on Computer Graphics and Applications*, 2002.

R. Beard, J. Lawton, and F. Hadaegh. A feedback architecture for formation control. In *Proceedings of the American Control Conference*, volume 6, pages 4087–4091, June 2000.

R. Beard, J. Lawton, and F. Hadaegh. A coordination architecture for spacecraft formation control. *IEEE Trans. Contr. Syst. Technol.*, 9:777–790, Nov. 2001.

C. Bechaz and H. Thomas. Gib portable tracking systems - the underwater gps solution. *Hydro International 2002*, 2002.

F. Belkhouche, B. Belkhouche, and P. Rastgoufard. Line of sight robot navigation toward a moving goal. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS*, 36(2):255–267, April 2006.

J. Bellingham, C. Goudey, T. Consi, J. Bales, D. A. amd J.J. Leonard, and C. Chryssostomidis. A second generation survey auv. *Proceedings of the IEEE Oceanic Engineering Society Conference Autonomous Underwater Vehicles (AUV) 94*, pages 148–155, July 1994.

Y. C. T. B. Bishop. Evaluation of robot swarm control methods for underwater mine countermeasures. In *System Theory, 2004. Proceedings of the Thirty-Sixth Southeastern Symposium on*, pages 294–298, 2004.

E. Bonabeau and G. Theraulaz. Swarm smarts. *Scient. Am.*, 282:72–79, 2000.

Brancart and P. Claude. The evolution of the darpa/arpa uuv program. *International Advanced Robotics Programme (IARP): Mobile Robots for Subsea Environments*, May 1994.

R. Brown and J. Jennings. A pusher/steerer model for strongly cooperative mobile robots cooperation. In *IEEE/RSJIROS*, pages 562–568, 1995.

B. Bullnheimer, R. F. Hartl, and C. Strauss. An improved ant system algorithm for the vehicle routing problem. *Annals of Operations Research: Nonlinear Economic Dynamics and Control (Dawid, Feichtinger, and Hartl, Eds.)*, 1999.

S. Camazine, J.-L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraulaz, and E. Bonabeau. *Self-Organization in Biological Systems*. Princeton University Press, 2003. ISBN 0-691-01211-3.

Y. U. Cao, A. S. Fukunaga, and A. B. Kahng. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4:7–27, 1997.

M. Carreras, J. Batlle, P. Ridao, and G. Roberts. An overview on behaviour-based methods for auv control. In *5th IFAC MCMC, Alborg (Denmark)*, pages 141–146, August 2000.

R. Cassinis, G. Bianco, A. Cavagnini, and P. Ransenigo. Strategies for navigation of robot swarms to be used in landmines detection. In *Advanced Mobile Robots, 1999. (Eurobot '99) 1999 Third European Workshop on*, pages 211–218, 1999.

Y. Chen and Z. Wang. Formation control: A review and a new consideration. *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference*, pages 3181–3186, Aug. 2005.

L. Cheng, Y. Wang, and Q. Zhu. A communication-based multirobot rigid formation control system: design and analysis. *Int. J. Modelling, Identification and Control*, 1(1):13–22, 2006.

E. Coiras, Y. Petillot, and D. Lane. Automatic rectification of side-scan sonar images. *Proceedings of the International Conference Underwater Acoustic Measurements: Technologies and Results Heraklion, Crete, Greece*, June 2005.

P. Collar and S. McPhail. Autosub - an autonomous unmanned submersible for ocean data collection. *IEE Electronics and Communication Engineering Journal*, 7(3):105–114, June 1995.

N. Cowan, O. Shakerina, R. Vidal, and S. Sastry. Vision-based follow-the-leader. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 2, pages 1796–1801, 2003.

A. K. Das, R. Fierro, V. Kumar, J. P. Ostrowski, J. Spletzer, and C. J. Taylor. A vision-based formation control framework. *Robotics and Automation, IEEE Transactions on*, 18(5):813–825, 2002.

A. Dawson, M. Downer, G. Patterson, P. Pattni, and D. Warrender. Subzero: A low cost remotely operated underwater vehicle. *Group design report, Faculty of Engineering and Applied Science, University of Southampton*, 1992.

H. de Vries and J. C. Biesmeijer. Modelling collective foraging by means of individual behaviour rules in honey-bees. *Behav Ecol Sociobiol*, 44:109–124, 1998.

J. Desai, J. Ostrowski, and V. Kumar. Controlling formations of multiple mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 4, pages 2864–2869, May 1998.

S. Doctor, G. K. Venayagamoorthy, and V. G. Gudise. Optimal pso for collective robotic search applications. *IEEE Congress on Evolutionary Computation*, pages 1390–1395, 2004.

B. Donald, J. Jennings, and D. Rus. Designing control laws for cooperative agent teams. In *IEEE ICRA*, pages 1896–1903, 1994.

M. Dorigo, V. Maniezzo, and A. Colorni. Ant system: Optimization by a colony of cooperating agents. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS-PART B CYBERNETICS*, 26, 1996.

Z. Feng, R. Stansbridge, D. White, A. Wood, and R. Allen. Subzero iii - a low cost underwater flight vehicle. *1st IFAC workshop on guidance and control of underewater vehicles*, 2003.

G. Ferri, E. Caselli, V. Mattoli, A. Mondini, B. Mazzolai, and P. Dario. A biologically-inspired algorithm implemented on a new highly flexible multi-agent platform for gas source localization. In *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006. BioRob 2006.*, pages 573–578, February 2006.

D. Fox, W. Burgard, H. Kruppa, and S. Thrun. Collaborative multi-robot exploration. *Autonomous Robots*, 3(8):325–344, 2000.

J. Fricke. Down to the sea in robots. In *Technology Review*, pages 46–55. Association of Alumni and Alumnae of MIT, Cambridge Massachusetts, October 1994.

J. Ghommem, F. Mnif, G. Poisson, and N. Derbel. Nonlinear formation control of a group of underactuated ships. In *OCEANS 2007 - Europe*, pages 1–8, 2007.

K. Hattori, Y. Narita, Y. Kashimori, and T. Kambara. Self-organized critical behavior of fish school and emergence of group intelligence. *Neural Information Processing, 1999. Proceedings. ICONIP '99. 6th International Conference on*, 2:465–470, 1999.

D. Hristu-Varsakelis and C. Shao. Biologically-inspired optimal control: learning from social insects. *International Journal of Control*, 77(18):1545–1566, 2004.

H. Hu, I. Kelly, D. Keating, and D. Vinagre. Coordination of multiple mobile robots via communication. In *Proceedings of SPIE'98, Mobile Robots XIII Conference, Boston*, pages 94–103, November 1998.

T. Iwai and M. Hisada. *Fishes-Illustrated Book of Gakken (in Japanese)*. Gakken, 1998.

J. Joshua and B. Gary. Parallel simulation of uav swarm scenarios. *Proceedings of the 2004 Winter Simulation Conference*, 2004.

S. Kalantar and U. R. Zimmer. Contour shaped formation control for autonomous underawter vehicles using canonical shape descriptors and deformable models. In *OCEANS 2004*, Kobe, Japan, 2004.

A. Kamphuis and M. H. Overmars. Motion planning for coherent group of entities. In *IEEE International Conference on Robotics and Automation*, San Diego,CA, 2004a. IEEE Press.

A. Kamphuis and M. H. Overmars. Finding paths for coherent groups using clearance. In *Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, 2004b.

J. Kennedy and R. Eberhart. Particle swarm optimization. *Proceeding of IEEE Int'1. Conference on Neural Networks (Perth, Australia)*, 4:1942–1948, 1995.

R. Kruse, J. Gebhardt, and F. Klawon. *Foundation of Fuzzy Systems.* Wiley, 1994.

C. Kube and E. Bonabeau. Cooperative transport by ants and robots. *Robotics and Autonomous Systems*, 30:85–101, 2000.

R. Lea, R. Allen, and S. Merry. A comparative study of control techniques for an underwater flight vehicle. *International Journal of Systems Science*, 30(9):947–964, Aug. 1999.

Y. Li, B. Li, Z. Sun, and Y. Song. Fuzzy technique based close formation flight control. In *Industrial Electronics Society, 2005. IECON 2005. 31st Annual Conference of IEEE*, Nov. 2005.

Z.-C. Lin and H. Chang. Application of fuzzy set theory and back-propagation neural networks in progressive die design. *Journal of Manufacturing Systems*, 1996.

Y. Liu and K. M. Passino. Swarm intelligence: Literature overview. Website: www.ece.osu.edu/ passino/swarms.pdf, March 2000.

P. Lučić and D. Teodorović. Transportation modeling: an artificial life approach. *Tools with Artificial Intelligence, 2002. (ICTAI 2002). Proceedings. 14th IEEE International Conference on*, pages 216–223, 2002.

R. Madhavan, K. Fregene, and L. E. Parker. Distributed cooperative outdoor multi-robot localization and mapping. *Autonomous Robots*, 17:23–39, 2004.

J. E. Manley. The role of risk in auv development and deployment. In *Proceedings of OCEANS07 IEEE Aberdeen Conference and Exhibition*, June 2007.

R. L. Marks, T. McLain, D. W. Miles, S. M. Rock, G. A. Sapilewski, H. H. Wang, M. J. Lee, and R. C. Burton. Monterey bay aquarium research institute/stanford aerospace robotics laboratory joint research program summer report 1992. Technical report, Monterey Bay Aquarium Research Institute, Pacific Grove California, 1992.

R. L. Marks, M. J. Lee, and S. M. Rock. Using visual sensing for control of an underwater robotic vehicle. *International Advanced Robotics Programme (IARP): Mobile Robots for Subsea Environments*, page 13, May 3-6 1994.

R. L. Marks, S. M. Rock, and M. J. Lee. Real-time video mosaicking of the ocean floor. In *Proceedings of the IEEE Oceanic Engineering Society Conference Autonomous Underwater Vehicles (AUV) 94*, pages 21–27. Cambridge Massachusetts, July 19-20 1994.

M. Mataric, M. Nilsson, and K. Simsarian. Cooperative multi-robot box-pushing. In *IEEE/RSJIROS*, pages 556–561, 1995.

P. McDowell, J. Chen, and B. Bourgeois. Uuv teams, control from a biological perspective. In *Oceans '02 MTS/IEEE*, volume 1, pages 331–337, 2002.

W. Naeem, R. Sutton, S. Ahmad, and R. Burns. A review of guidance laws applicable to unmanned underwater vehicles. *The Journal of Navigation*, 56:15–29, 2003.

I. Noda, S. Suzuki, H. Matsubara, M. Asada, and H. kitano. Robocup-97: The first robot world cup soccer games and conferences. *AI Magazine*, 3(19):49–59, Fall 1998.

T. OBOSHI, S. KATO, A. MUTOH, and H. ITOH. A simulation study on the form of fish schooling for escape from predator. *Forma*, 18:119–131, 2003.

Orbital. Swarm algorithms advanced techniques for multi-agent coordination. 2003.

J. Ouellette. Science goes exploring under the sea. *American Institute of Physics*, pages 20–23, 2002.

B. Partridge. The structure and function of fish schools. *Scienfitfic American*, 246: 90–99, 1982.

B. L. Partridge and T. Pitcher. the sensory basis of fish schools: relative role of lateral line and vision. *Journal of Comparative Physiology*, 135:315–325, 1980.

F. Rêpoulias and E. Papadopoulos. Planar trajectory planning and tracking control design for underactuated auvs. *Ocean Engineering*, 34(11-12):1650–1667, Aug. 2007.

C. Reynolds. Website http://www.red3d.com/cwr/boids/.

C. W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics, 21(4) (SIGGRAPH '87 Conference Proceedings)*, pages 25–34, 1987.

N. Sannomiya, H. Nakamine, and K. Matuda. Application of system theory to modeling of fish behavior. *Decision and Control, 1990. , Proceedings of the 29th IEEE Conference on , 5-7 Dec. 1990*, 5:2794–2799, 1990.

J. Sasaki, J. Ota, E. Yoshida, D. Kurabayashi, and T. Arai. Cooperating grasping of a large object by multiple mobile robots. In *IEEE ICRA*, pages 1205–1210, 1995.

T. Schmitt, R. Hanek, M. Beetz, S. Buck, and B. Radig. Cooperative probabilistic state estimation for vision-based autonomous mobile robots. *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION*, 18(5):670–684, October 2002.

W. Seah, H. X. Tan, Z. Liu, and M. Ang. Multiple-uuv approach for enhancing connectivity in underwater ad-hoc sensor networks. In *Proceedings of MTS/IEEE OCEANS, 2005*, volume 3, pages 2263–2268, 2005.

S. Sen, M. Sekaran, and J. Hale. Learning to coordinate without sharing information. In *Proceeding of AAAI*, pages 426–431, 1994.

E. Shaw. The schooling of fishes. *Scientific American*, 206:128–138, 1962.

Sheetal and G. K. Venayagamoorthy. Unmanned vehicle navigation using swarm intelligence. *Intelligent Sensing and Information Processing, 2004. Proceedings of International Conference*, pages 249–253, 2004.

Sontek. *Acoustic Doppler velocimeter (ADV) operation manual (Firmware version 4.0)*. SonTek, San Diego, CA, 1997.

J. B. D. Sousa and F. L. Pereira. Coordinated control strategies for networked vehicles: an application to autonomous underwater vehicles. 2003.

E. M. Sozer, M. Stojanovic, and J. G.Proakis. Underwater acoustic networks. *IEEE Journal of Oceanic Engineering*, 25(1), Jan 2000.

K. Stephens, B. Pham, and A. Wardhani. Modelling fish behaviour. *Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 71–78, 2003.

I. R. Stergios and A. B. George. Distributed multirobot localization. *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION*, 18(5):781–795, October 2002.

K. Symon. *Mechanics*. Addison-Wesley, 1971. ISBN 0-201-07392-7.

T. Takagi, Y. Moritomi, J. Iwata, H. Nakamine, and N. Sannomiya. Mathematical model of fish schooling behaviour in a set-net. *ICES Journal of Marine Science*, 61(7):1214–1223, 2004.

M. Tambe, M. Adibi, Y. Al-Onaizan, A. Erdem, G. Kaminka, S. Marsella, and I. Muslea. Building agent teams using an explicit teamwork model and learning. *Artificial Intelligence*, 2(110):215–239, June 1999.

K.-H. Tan and M. Lewis. Virtual structures for high-precision cooperative mobile robotic control. In *Proceedings of the 1996 IEEE/RSJ International Conference Intelligent Robots and Systems*, volume 1, pages 132–139, Nov. 1996.

P. Tarasewich and P. R. McMullen. Swarm intelligence: power in numbers. *Commun. ACM*, 45(8):62–67, 2002.

X. Tu. *Artificial Animals for Computer Animation: Biomechanics, Locomotion, Perception, and Behavior*. Springer Berlin / Heidelberg, 1999. ISBN 0302-9743.

X. Tu and D. Terzopoulos. artificial fishes:physics,locomotion,perception,behaviours. *Computer Graphics, SIGGRAPH*, pages 43–50, 1984.

P. Visscher. Animal behaviour: How self-organization evolves. *Nature*, 421:799–800, 2003.

Z. Wang, E. Nakano, and T. Matsukawa. Cooperating multiple behavior-based robots for object manipulation. In *IEEE/RSJIROS*, pages 2561–2566, 1994.

J. Wejchert and D. Haumann. Animation aerodynamics. *Computer Graphics*, 25(4): 58–62, July 1991.

S. B. Williams, G. Dissanayake, and H. Durrant-Whyte. Towards multi-vehicle simultaneous localisation and mapping. *Robotics and Automation, Proceedings. ICRA '02. IEEE International Conference*, 3:2743–2748, 2002.

X. Wu, Z. Feng, J. Zhu, and R. Allen. Line of sight guidance with intelligent obstacle avoidance for autonomous underwater vehicles. *Oceans'06 MTS/IEEE-Boston*, September 2006.

Y. Yonezawa and T. Kikuchi. Ecological algorithm for optimal ordering used by collective honey bee behavior. *Micro Machine and Human Science, 1996. , Proceedings of the Seventh International Symposium*, pages 249–256, 1996.

D. Zhang, G. Xie, J. Yu, and L. Wang. Adaptive task assignment for multiple mobile robots via swarm intelligence approach. *Robotics and Autonomous Systems*, 55: 572–588, 2007.

Y. Zhang. Acoustic doppler velocimeter flow measurement from an autonomous underwater vehicle with applications to deep ocean convection. *Journal of Atmospheric and Oceanic Tehnology*, 18:2038–2051, December 2001.

Appendix A

Mapping the velocity with Cartesian coordinate on polar coordinate system

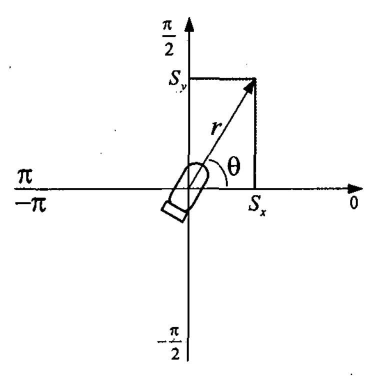


Figure A.1: Mapping the velocity with Cartesian coordinate on polar coordinate system, where (S_x, S_y) is the Cartesian coordinate of the vehicle's velocity and (θ, r) is the polar coordinate of the vehicle's velocity.

The velocities of vehicles represent as Cartesian coordinates. In order to be easy to compare velocity directions, the Cartesian coordinates of vehicles' velocities must convert to polar coordinates. Figure A.1 shows that how to map the velocity with Cartesian coordinate on polar coordinate system.

The two Cartesian coordinates x and y can be converted to polar coordinate r by equation A.

$$r = \sqrt{S_x^2 + S_y^2} \tag{A.0.1}$$

To determine the angular coordinate θ, the following two ideas must be considered: For $r = 0$, θ can be set to any real value; for $r \neq 0$, to get a unique representation for θ, it must be limited to an interval of size 2π.

To obtain θ in the interval $[-\pi, \pi]$, the following procedure may be used (**arctan** denotes the inverse of the tangent function):

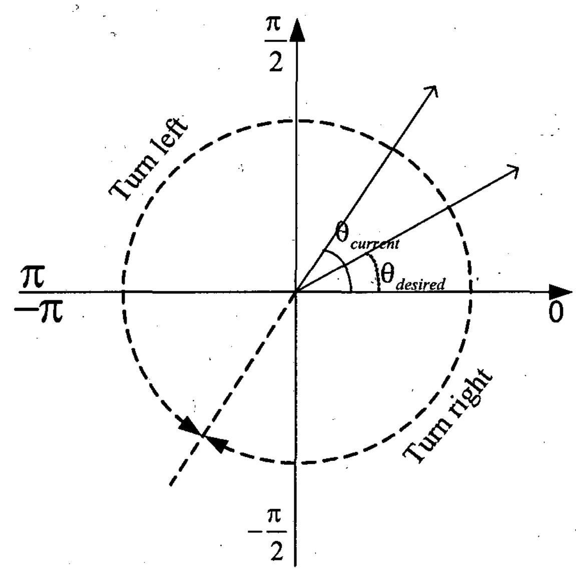
Procedure 3 Converting Cartesian coordinates to polar coordinates

if $S_x > 0$ and $S_y \geq 0$ **then**
    $\theta \leftarrow \arctan(S_y/S_x)$
end if
if $S_x < 0$ and $S_y \geq 0$ **then**
    $\theta \leftarrow \pi - \arctan(S_y/S_x)$
end if
if $S_x < 0$ and $S_y \leq 0$ **then**
    $\theta \leftarrow \pi + \arctan(S_y/S_x)$
end if
if $S_x > 0$ and $S_y \leq 0$ **then**
    $\theta \leftarrow \arctan(S_y/S_x)$
end if
if $S_x = 0$ and $S_y > 0$ **then**
    $\theta \leftarrow \pi/2$
end if
if $S_x = 0$ and $S_y < 0$ **then**
    $\theta \leftarrow -\pi/2$
end if

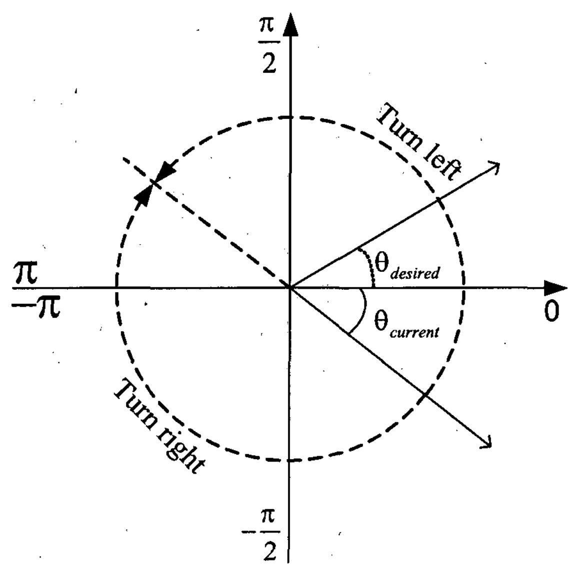
Appendix B

Dynamic manoeuvering model: steering control procedure

The mathematical procedure for steering control varies when the current velocity direction of the vehicle lies in a different quadrant. The figure B.1(a) and procedure 4 in Appendix A describe the details of the steering control procedure when the current velocity direction of the vehicle is in 1st or 2nd quadrant. When current velocity direction of the vehicle is in 3rd or 4th quadrant, figure B.1(b) and procedure 5 in Appendix A are used for steering direction control.



(a) Steering control when the current velocity direction of the vehicle is in 1st or 2nd quadrant.



(b) Steering control when the current velocity direction of the vehicle is in 3rd or 4th quadrant.

Figure B.1: Steering direction control when the current velocity direction of the vehicle is in different quadrant.

Procedure 4 Steering control when current velocity and direction of the vehicle is in the 1st or 2nd quadrant, where ω is the angular change velocity.

if $\theta_{current} < \pi$ and $\theta_{current} \geq 0$ **then**
    if $\theta_{desired} \leq \theta_{current}$ and $\theta_{desired} \geq -\pi + \theta_{current}$ **then**
        if $\theta_{current} - \theta_{desired} > \omega \times t$ **then**
            $\theta_{next} \leftarrow \theta_{current} - \omega \times t$
        end if
        if $\theta_{current} - \theta_{desired} \leq \omega \times t$ **then**
            $\theta_{next} \leftarrow \theta_{desired}$
        end if
    end if
    if $\theta_{desired} > \theta_{current}$ or $\theta_{desired} < -\pi + \theta_{current}$ **then**
        if $\theta_{desired} < -\pi + \theta_{current}$ **then**
            $\theta_{desired} \leftarrow 2\pi + \theta_{desired}$
        end if
        if $\theta_{desired} - \theta_{current} > \omega \times t$ **then**
            $\theta_{next} \leftarrow \theta_{current} + \omega \times t$
        end if
        if $\theta_{desired} - \theta_{current} \leq \omega \times t$ **then**
            $\theta_{next} \leftarrow \theta_{desired}$
        end if
    end if
end if

Procedure 5 Steering direction control when the current velocity direction of the vehicle is in the 3rd or 4th quadrant

if $\theta_{current} < 0$ **then**
  if $\theta_{desired} \leq \theta_{current}$ and $\theta_{desired} \geq -\pi + \theta_{current}$ **then**
    if $\theta_{current} - \theta_{desired} > \omega \times t$ **then**
      $\theta_{next} \leftarrow \theta_{current} + \omega \times t$
    end if
    if $\theta_{current} - \theta_{desired} \leq \omega \times t$ **then**
      $\theta_{next} \leftarrow \theta_{desired}$
    end if
  end if
  if $(\theta_{desired} \leq \pi$ and $\theta_{desired} > \pi + \theta_{current})$ or $(\theta_{desired} \geq -\pi$ and $\theta_{desired} < \theta_{current})$ **then**
    if $\theta_{desired} \leq \pi$ and $\theta_{desired} > \pi + \theta_{current}$ **then**
      $\theta_{desired} \leftarrow -2\pi + \theta_{desired}$
    end if
    if $\theta_{current} - \theta_{desired} > \omega \times t$ **then**
      $\theta_{next} \leftarrow \theta_{current} - \omega \times t$
    end if
    if $\theta_{current} - \theta_{desired} \leq \omega \times t$ **then**
      $\theta_{next} \leftarrow \theta_{desired}$
    end if
  end if
end if