The Design and Implementation of A Multi-Agent Architecture to Increase Coordination Efficiency in Multi-AUV Operations

Christopher Carson Sotzing

A dissertation submitted for the degree of Doctor of Philosophy

Heriot-Watt University

School of Engineering and Physical Sciences

January 2009

Abstract

This research addresses the problem of coordinating multiple autonomous underwater vehicle (AUV) operations. An intelligent mission executive has been created that uses multi-agent technology to control and coordinate multiple AUVs in communication deficient environments. By incorporating real time vehicle prediction, blackboard-based hierarchical mission plans and mission optimisation in conjunction with a simple broadcast communication system this system aims to handle the limitations inherent in underwater operations and intelligently control multiple vehicles. In this research efficiency is evaluated and then compared to the current state of the art in multiple AUV control. The research is then validated in real AUV coordination trials.

Results will show that compared to the state of the art the control system developed and implemented in this research coordinates multiple vehicles more efficiently and is able to function in a range of poor communication environments. These findings are supported by in water validation trials with heterogeneous AUVs.

This thesis will first present an in depth state of the art of the related research topics including multi-agent systems, collaborative robotics and autonomous underwater vehicles. The development and functionality of this research will then be explained followed by a detailed description of the experiments. Results are then presented both for the simulated and real world trials followed by a discussion of the findings.

Acknowledgements

I'd first like to thank Professor David M. Lane and Dr. Keith Brown for their support and guidance throughout this project. I'd also like to thank the staff and researchers in the Ocean Systems Laboratory, particularly the PhD students of G.86, who helped make this research far more rewarding and far more fun. Additionally this work was supported by a scholarship from Heriot-Watt University, without which none of this would have been possible. A big thanks goes out to all my friends who listened to the trials and tribulations of PhD life over (multiple) pints and helped keep me sane! Finally I'd like to thank my family for their love and support as I travelled far from home to avoid the real world for a few more years.

Declaration Statement

ACADEMIC REGISTRY
Research Thesis Submission

Name*:*	Christopher Carson Sotzing		
School/PGI:	School of Engineering & Physical Sciences		
Version: *(i.e. First, Resubmission, Final)*	First	Degree Sought (Award **and** Subject area)	Doctor of Philosophy

Declaration

In accordance with the appropriate regulations I hereby submit my thesis and I declare that:

1) the thesis embodies the results of my own work and has been composed by myself
2) where appropriate, I have made acknowledgement of the work of others and have made reference to work carried out in collaboration with other persons
3) the thesis is the correct version of the thesis for submission and is the same version as any electronic versions submitted*.
4) my thesis for the award referred to, deposited in the Heriot-Watt University Library, should be made available for loan or photocopying and be available via the Institutional Repository, subject to such conditions as the Librarian may require
5) I understand that as a student of the University I am required to abide by the Regulations of the University and to conform to its discipline.

* *Please note that it is the responsibility of the candidate to ensure that the correct version of the thesis is submitted.*

Signature of Candidate*:*		Date:	

Submission

Submitted By *(name in capitals):*	
Signature of Individual Submitting:	
Date Submitted:	

For Completion in Academic Registry

Received in the Academic Registry by *(name in capitals):*			
Method of Submission *(Handed in to Academic Registry; posted through internal/external mail):*			
E-thesis Submitted (***mandatory from November 2008***)			
Signature:		Date:	

Table of Contents

Tables and Figures

Figures

Tables

Glossary

ABE	Autonomous Benthic Explorer
ACTRESS	Actor-Based Robots and Equipments Synthetic System
ALI	Application Layer Interface
ARF	Augmented Reality Framework
AuRA	Autonomous Robot Architecture
AUSS	Advanced Unmanned Search System
AUV	Autonomous Underwater Vehicle
AUVC	Autonomous Underwater Vehicle Controller
CADCAC	Computer Aided Detection Computer Aided Classification
CCL	Compact Control Language
CEBOT	Cellular Robotic System
CURV	Cable Controlled Underwater Recovery Vehicle
DAI	Distributed Artificial Intelligence
DBB	Distributed Blackboard System
DRRS	Dynamically Reconfigurable Robotic System
FIPA	The Foundation for Intelligent Physical Agents
HOV	Human Occupied Vehicle
ITOCA	Intelligent Task-Oriented Control Architecture
KPI	Key Performance Indicator
KS	Knowledge Source
LMRS	Long Term Mine Reconnaissance System
MAS	Multi-Agent Systems
MCM	Mine Countermeasures
NASREM	NASA/NBS Standard Reference Model
NOAA	National Oceanic and Atmospheric Administration
REMUS	Remote Environmental Monitoring Units
RMM	Recursive Modelling Method
ROV	Remotely Operated Vehicle
SAUVIM	Semi-Autonomous Underwater Vehicle for Intervention Missions
SPURV	Self Propelled Underwater Research Vehicle
TCA	Task Control Architecture
UAV	Unmanned Air Vehicle
UGV	Unmanned Ground Vehicle
UUV	Unmanned Underwater Vehicle
UxV	Unmanned Vehicle
WHOI	Woods Hole Oceanographic Institute

List of Publications by the Candidate

[1] C.C. Sotzing and D.M. Lane, “Multi-AUV Coordination in Poor Communication Environments: The Design and Implementation of A Multi-Agent Architecture to Increase Coordination Efficiency in Multi-AUV Operations,” Sea Technology, Volume 49, Issue 11, 23-28, 2008.

[2] C.C. Sotzing, N. Johnson and D.M. Lane, "Improving Multi-AUV Coordination with Hierarchical Blackboard-Based Plan Representation," 27th Workshop of the UK Planning and Scheduling Special Interest Group, PlanSIG 2008, 110-117, Dec 11-12th 2008, Edinburgh.

[3] C.C. Sotzing and D.M. Lane. “Improving the Coordination Efficiency of Multiple AUV Operations Using Prediction of Intent.” Systems Engineering for Autonomous Systems Defence Technology Centre. July 2008.

[4] C.C. Sotzing, J. Evans and D.M. Lane. “A Multi-Agent Architecture to Increase Coordination Efficiency in Multi-AUV Operations,” Proceedings of IEEE Oceans Europe, 2007.

[5] J. Evans, C.C. Sotzing, P. Patrón and D.M. Lane. “Cooperative Planning Architectures for Multi-Vehicle Autonomous Operations.” Systems Engineering for Autonomous Systems Defence Technology Centre. July 2006.

Chapter 1

Introduction

1.1 Overview

For centuries humans have been seeking the aid of machines to help with tasks that would be otherwise too difficult, tedious or impossible. From the ancient Greek Antikythera mechanism from around 150 BC (widely described as the first computer) to modern assembly line industrial robots, the desire for automated tools to make life easier is growing every day. Though the tem "robot" wasn't coined until the 1920s, the ancestors of today's robots have been around for centuries. One of the earliest examples was Leonardo da Vinci's mechanical knight based on his research for the Vitruvian Man. Recently discovered in notes dating back to the 1490s, this early humanoid consisted of a suit of armour connected with ropes and pulleys that could recreate simple human behaviours including moving its limbs and head. Other early robots include the 18th century Japanese karakuri ningyō mechanical puppets, "Elektro" a humanoid robot built by Westinghouse for the 1939 New York Worlds Fair and "Unimate", the first industrial robot built by General Motors in 1961 for its assembly line. Robot technology has grown significantly in the past 30 years with more and more examples and applications seen every day.

Though there is still a considerable amount of research continuing in humanoid robots today, a major focus is devoted to mobile robotics, or autonomous vehicles. These robots take the form of all types of vehicles including cars, planes, boats, submarines and even spacecraft that can operate without the need for human control. Autonomous vehicles are especially useful in areas where a human presence is particularly costly or dangerous. Some of the most famous autonomous vehicles in recent history have been the exploration rovers that were sent to explore the surface of the planet Mars . The two vehicles, *Spirit* and *Opportunity,* captured the attention of the world with their ability to go where we as humans could not.

Mars however is not the only inhospitable, unexplored environment available to us; in fact it's not even the closest. For that we must look to our own planet, specifically to

71% of it. Despite being right in front of us it is often said that we know more about the surface of the moon than we do about our oceans. The challenge is that undersea exploration is a dangerous undertaking for humans. Aside from the obvious problem of respiration, humans weren't made for the extreme depth, temperature and other hazards of ocean existence. Robots provide an ideal platform for undersea work and this fact has spurred the evolution of many generations of underwater robots, culminating in the current technology of autonomous underwater vehicles (AUVs). Not just limited to deep water exploration AUVs are now being used in a wide variety of sectors from oceanography to offshore energy and defence.

As AUV technology (and robotics technology in general) matures there is a growing need for multiple vehicles to work together to solve more complex tasks. Unlike other robotic systems however AUVs must operate in environments that are in many ways far more inhospitable, and communication unfriendly. This makes the already complex task of coordination that much more difficult. The current state of the art in multi-AUV control is only a first step and therefore limited in its ability to coordinate vehicle actions. This research aims to address this problem and improve the ability of AUVs to work together underwater.

1.2 Research Objectives

Autonomous underwater vehicles are limited to relatively primitive acoustic communication, unlike the high-bandwidth, low latency communication protocols available to other forms of mobile robotics. Subsequently as AUV technology is extended to allow for multi-vehicle coordination a major challenge arises: how can multiple underwater robots be coordinated *efficiently* in environments where communication is severely limited? In addition, how does group size affect efficiency? How many vehicles are required for a given mission?

One possible solution lies with multi-agent systems. In these systems individual autonomous software entities called *agents* interact to solve complex problems. The autonomy and intelligence of agents, and consequently multi-agent systems make them an interesting and as of yet untested method for the coordination of autonomous underwater vehicles.

Considering these challenges the objectives of this research are as follows:

- Create a control architecture using multi-agent properties aiming to coordinate multiple autonomous underwater vehicles in environments with limited communication.

- Compare the aforementioned system to the current state of the art in multi-AUV coordination by testing both in a series of common AUV vignettes and evaluating them for efficiency.

- Investigate the factors that affect efficiency in multi-AUV operations and determine which have the greatest influence.

- Determine the benefit of multiple autonomous underwater vehicles over single vehicle systems including optimal group size.

- Use the architecture to coordinate multiple real AUVs in an actual communication unfriendly environment.

1.3 Thesis Structure

This thesis has been organised into 3 main areas; background research, system/experimental design, and results/conclusions as shown in Figure 1.1. Chapter 2 will initiate the background research chapters and present a literature review of multi-agent systems. First a definition of important concepts is given followed by a history of the field including some of the foundation systems. This is followed by a review of the current state of the art of multi-agent systems. The chapter concludes with an examination of available tools and a critique of the state of the art.

Chapter 3 presents collaborative robotics. Again, a history is given including a description of some of the early collaborative robotics systems that have influenced the field. The current state of the art is given by presenting the most common research topics of the field. The chapter finishes with an analysis of the state of the art including the pros and cons.

Figure 1.1 Diagram showing the structure of this thesis.

The background research chapters conclude with Chapter 4 where autonomous underwater vehicles are introduced. Following the structure set up in the preceding two chapters, a history of the technology is given first. This is succeeded by a presentation of the applications of AUV technology as well as the different types of vehicle currently in operation. Finally an in depth discussion on control architectures is followed by a section detailing and reviewing the current state of the art.

The DELPHÍS system is presented in Chapter 5. This chapter details the system created in this research by going through the different software modules of the architecture as well as the background research that it was based upon. The functionality is also illustrated to help describe the abilities of the system.

Chapter 6 details the experimental setup for this research. It begins with a discussion on simulated vs. real platforms and the costs and benefits of each. This is followed by two sections presenting the real and simulated resources used to evaluate this work. The experiments themselves are then described in detail including the systems being tested, efficiency metrics, mission vignettes and methodology.

Following the system/experimental design sections Chapter 7 presents the results from the aforementioned experiments. Data is given on two mission vignettes (mine countermeasures and pipeline tracking) in terms of a number of efficiency metrics. These include redundancy, missed goals, target detection, mission time and an all encompassing efficiency metric calculated by the preceding values. Graphs are displayed with accompanying descriptions. The simulated results are followed by real world results where in water experiments are presented to validate the experimental data.

The data presented in Chapter 7 is then discussed in Chapter 8. First an analysis of the recorded behaviour is given including examples of witnessed behaviour. The following sections go through the bulk of the results in detail explaining their significance as well as the reasoning behind the trends. Finally, the real world validation data is described followed by a discussion on AUV group size based on the results of this study.

The final chapter presents the conclusions that can be made from this research by explaining the achievements and then making recommendations on how this research can be used. Future research directions are then given. Following the conclusion experimental data from both the simulated and real experiments is displayed in the appendices.

Chapter 2

Multi-Agent Systems

2.1 Introduction

This research has investigated the application of multi-agent systems (MAS) to the coordination of multiple AUVs. Such systems use groups of intelligent agents to solve complex problems. Before explaining the field of multi-agent systems first the building blocks must be defined; intelligent, interacting agents. In his book *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, Gerhard Weiss writes:

> *"Agents" are autonomous, computational entities that can be viewed as perceiving their environment through sensors and acting upon their environment through effectors… "Intelligent" indicates that the agents pursue their goals and execute their tasks such that they optimize some given performance measures… "Interacting" indicates that the agents may be affected by other agents or perhaps by humans in pursuing their goals and executing their tasks.* [80]

This chapter will first explain some important concepts in multi-agent systems, then briefly review the history of such systems, go over the main attributes of the current state of the art and finally present a critique of current multi-agent systems.

2.2 Semantics

When referring to multi-agent systems (as well as multi-robot systems) there are a few words that are constantly used to describe agent interaction, and whose meanings are often confused. *Communication, awareness, cooperation* and *collaboration* are extremely important to define because they have separate and very distinct meanings in these types of intelligent systems. An excellent set of definitions drawn up for multi-robot systems but that apply equally to multi-agent systems can be found in [75] and will be summarized here.

Communication is the easiest because is it the most straight forward. It is the ability of agents to pass information to each other and can happen both *explicitly* (directly, agent to agent) and *implicitly* (indirectly, via the environment).

Awareness is the knowledge of the existence of other agents operating in the system. Awareness has a direct effect on communication because it affects whether or not explicit or implicit paradigms are used.

Cooperation and *collaboration* are harder to define and are often confused with each other. Both have to do with how agents work together to solve a problem but they do it in two distinct ways. The main difference is whether or not agents share a common goal. It can be said that multi-agent systems that *cooperate* are ones where agents each have their own goals and in the process of achieving these goals, the overall goal of the system is reached. For example, if a multi-agent system is created for a spell-check program, some agents will be checking the spelling and others will be checking the grammar. Though the agents work together to output text that is legible, the agents themselves have distinct and separate goals.

Collaboration is similar to *cooperation* but in this case all agents have the same goal, though they may go about solving it in different ways. The key factor is that agent actions are performed in response to other agent actions. One of the best examples of this is blackboard systems, which will be described in more detail later in this section. In essence each agent is a specialist that works with other specialist agents to solve a complicated problem that no one agent could solve on their own.

In addition to collaboration and cooperation this study also uses the word *coordination*. For the purposes of this research coordination is used as an umbrella term to refer to both collaboration and cooperation.

2.3 History

In order to understand the abilities of multi-agent systems it is important to understand the history of these architectures. Multi-agent research has been going on for decades, originating from the field of distributed artificial intelligence (DAI). DAI was based around the concept of using multiple artificial intelligence systems to concurrently solve

problems by dividing them up. This technology evolved into what is now known as multi-agent systems.

There are a few pivotal systems that have influenced, and act as the foundation for, most of the research going on today. Actors, Blackboard Systems, and the Contract Net Protocol are essentially the parents of all multi-agent research and because the ideas presented in these projects are still very relevant today, the next section will compare and contrast these systems as an introduction to current multi-agent systems.

Before comparing the three systems in detail a brief summary will be given on the functionality of each.

2.3.1 Actors

One of the first multi-agent systems was Actors [1]. In the Actors system the architecture is made up of a network of nodes called *workers*. Each worker contains a number of agents called *actors* and consists of one or more processors, memory, and a mail system for communication with other workers. A worker is a problem solving entity and the more workers present the more distributed the process becomes. The key however to this system are the actors themselves. Each actor agent consists of a current behaviour and a mail address. Each actor has three abilities: it may send communications to any other actor, it may create new actors, and it may specify a replacement behaviour for itself. By delegating the work between the actors a worker can thereby solve a problem.

The Actors system has not stood the test of time as well as some of the other multi-agent systems mainly due to the primitive state of distributed systems when it was developed in the 1970s. In addition, one of the biggest issues is regarding its focus on task allocation and delegation which in turn depends upon good communication. This is a major challenge for distributed systems and in particular multi-robot systems and will be echoed throughout this section as more pivotal MAS are presented.

2.3.2 Blackboards

Another pivotal multi-agent system, the blackboard architecture [19] is based on the concept of a number of specialists standing around a blackboard solving a problem together. Individually, no one specialist or *knowledge source* (KS) can solve the problem but by collaborating a solution is possible. The blackboard architecture takes this idea and digitises it in that the knowledge sources become agents and the *blackboard* becomes a centralised database. As the problem is solved agents give their input whenever they can by analysing the blackboard's data. A diagram of a blackboard system can be seen in Figure 2.1.

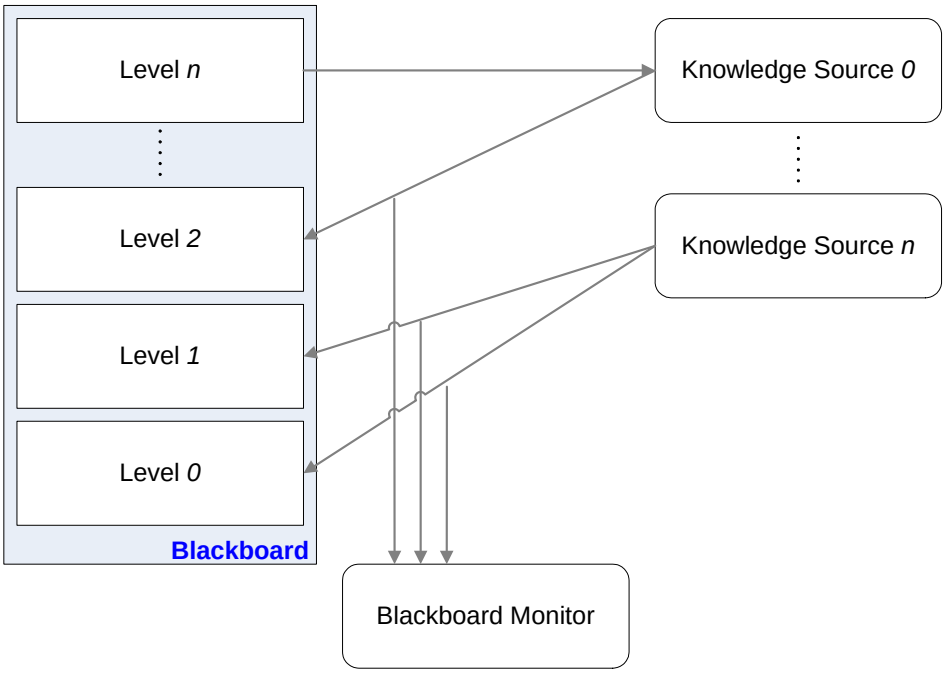


Figure 2.1 Diagram showing the blackboard architecture.

One of the first examples of blackboard architectures was the Hearsay-II system, developed by V. Lesser et al in the late 1970s and early 1980s [25]. The Hearsay-II system was a speech understanding system that allowed users to speak commands that would then be parsed and understood by the program. Individual KS program modules work with the blackboard data to determine the command spoken by the user. KS examples include functions that create word hypotheses from syllable classes and predict all possible words that might syntactically precede or follow a given phrase. KS interaction requests with the blackboard are watched by the Blackboard Monitor, which makes sure that no two KSs are writing to the blackboard at once. The Monitor writes to the Scheduling Queue which is then used by the Scheduler to control KS input.

Blackboards, though extremely powerful systems have some inherent drawbacks. The most important of which is like the Actors system, for a blackboard system to function it requires all the agents to be within communication range of the blackboard. For most applications this isn't an issue however for multi-robot applications it becomes an important challenge. One solution is to distribute the blackboard across multiple agents. In these distributed blackboard (DBB) systems each agent has a copy of the database which is synchronised with other blackboards when important changes occur [18]. This approach is used in [81] where agents using a DBB are used to control individual parts of a robot (manipulators, sensors, motion planning, etc.).

2.3.3 Contract Net Protocol

The Contract Net Protocol [62] is a multi-agent network made up of an unlimited number of nodes/agents. There are two different types of agents: *managers* and *contractors* (though roles are not specified in advance and manager agents can act as contractors too). Managers have more data processing abilities and therefore analyze information and assign tasks to the contractors who have more execution power (i.e. data collection, function computations, etc.). For a given problem, contractor agents make bids to a manger agent regarding their solution. The manger agent then chooses the best contractor solution makes a *contract* with that agent. The Contract Net Protocol is based on the pairing of managers and contractors.

Though the benefits of the CNP are many, including the ability of any agent to become a contractor, there are some potential drawbacks. One such drawback is that when a manager agent advertises a contract the highest bidding contractor agent might not be the best agent for the task. This can happen if a more suitable agent is busy with another task [80]. Another potential issue with the CNP, in particular to multi-robot systems, is that like the previous two MAS it relies heavily on inter-agent communication. If agents are distributed throughout multiple robots that have unreliable communication links there can be a problem.

2.3.4 MAS Characteristics

To get into more detail about the preceding three systems and to compare and contrast them, this section will go through a few major characteristics of multi-agent systems and see how these three pivotal architectures compare.

Table 2.1 Three pivotal multi-agent systems and their important characteristics.

	Blackboards	**Actors**	**Contract Net Protocol**
Open / Closed System	Closed	Open	Open
Communication	Implicit	Explicit	Explicit
Cooperation / Collaboration	Collaboration	Cooperation	Cooperation
Homogeneous / Heterogeneous	Heterogeneous	Heterogeneous	Heterogeneous
Conflict / Task Allocation	Supervisor agent can determine which agent is most suitable for a contested job	Delegation	"manager" can determine which agent is most suitable for a contested job

2.3.4.1 Open/Closed System

Multi-agent systems can be either *open* or *closed*. An open system is one where the agents are aware of the other agents in the environment whereas in a closed system the agents are only aware of themselves and the environment. Actors and the Contract Net Protocol are open systems because agents are aware of and have direct contact with each other whereas in contrast Blackboard systems are closed and communication is done via the centralised blackboard database. This relates directly to agents' ability to communicate, which will be discussed next.

2.3.4.2 Communication

In each of these three systems agents communicate in different ways. Blackboard systems communicate indirectly (*implicit* communication) by posting information to the blackboard; no direct agent to agent communication exists. Actors and the Contract Net Protocol allow for direct agent to agent communication (*explicit* communication) however in the Contract Net Protocol communication is done from node to node so that messages are passed from one adjacent agent to another until the recipient is reached. This issue of agent communication will be further discussed in the next section in relation to the current state of the art in multi-agent research.

2.3.4.3 Homogeneous/Heterogeneous Agents

Multi-agent systems can either be made up of *homogeneous* or *heterogeneous* agents. Homogeneous agents are agents that have the same internal structure, goals and possible actions. The only difference between these agents is their view of the world in terms of sensor data. Homogeneous MAS also have the benefit of relatively simple development because only one agent template has to be designed [35]. Heterogeneous agents are those that *do not* necessarily share these traits and therefore have different behaviours. It can be said that all three historic MAS systems mentioned here count as heterogeneous systems because by definition there are many different types of agents operating in each. The issue of homogeneous versus heterogeneous agents will be explained in more detail later in this chapter.

2.3.4.4 Cooperation/Collaboration

Another characteristic of multi-agent systems is whether they use cooperation or collaboration. That is, do all the agents have the same goal or do they work at separate *sub-goals* to reach a common one. A Blackboard system is a collaborating system where though they aren't directly aware of each other, agents work together via the blackboard towards a unified goal. The Actors and Contract Net Protocol architectures are cooperating systems; both incorporate agents that work towards specific sub-goals that result in the completion of a larger super-goal.

2.3.4.5 Conflict Resolution/Task Allocation

When dealing with both common and related goals there needs to be some sort of conflict resolution and task allocation built into a multi-agent system that allows for management in the case that two agents want to do the same thing or accidentally get in the way of each other. Blackboard systems and the Contract Net Protocol take similar approaches to this in the form of bidding. In a Blackboard system if two agents want to write to the blackboard at the same time they must first explain to a supervisor agent in the form of a bid what they would like to do. This supervisor agent then evaluates these bids and awards the ability to write to whichever agent it deems best. The Contract Net Protocol works in a similar way, having manager and contractor agents. When given a problem, the contractor agents bid to the manager with their solution. Like in Blackboard systems, the best bid leads to the contract. Actors systems work a bit differently in that any agent can delegate a job to any other agent. In doing so, a busy agent can pass of a request to another one that is idle or more capable of solving the

problem. This issue of bidding and task delegation is very dependent upon communication however and is a potential setback when this technology is used on multiple robots. This issue will be addressed later in this chapter.

2.4 Current State of the Art

The current state of the art in multi-agent systems can be divided into 4 distinct sub-sections as detailed in Peter Stone and Manuela Veloso's article in *Autonomous Robotics* entitled "Multiagent Systems: A Survey from a Machine Learning Perspective" [68]:

- Homogeneous Non-communicating Agents
- Heterogeneous Non-communicating Agents
- Homogeneous Communicating Agents
- Heterogeneous Communicating Agents

These subsections will be briefly described in the following sections including the issues brought up in [68] as well as relevant examples.

2.4.1 Homogeneous Non-communicating Agents

As mentioned earlier homogeneous agents are those that share the same internal structure and goals, differing only in their sensor inputs. One of the main areas of multi-agent research is homogeneous systems with little or no communication. In these systems agents have no explicit communication however information can still be passed via the environment. Some of the main issues in this branch of multi-agent research include reactive versus deliberative agents, local versus global perspective, modelling of other agents' states and how agents affect each other.

One of the current research areas in homogeneous non-communicating agents is how to predict agent behaviour when there is no communication. A novel solution was posed in [23,76] where the recursive modelling method (RMM) was used to predict agent behaviour. RMM works by applying local agent decision making to external agent data to predict behaviour. This is possible in homogeneous MAS because agents share

internal architectures and therefore can make educated guesses about other agents' behaviours when given sensor information.

2.4.2 Heterogeneous Non-communicating Agents

Heterogeneous non-communicating multi-agent systems are similar to homogeneous non-communicating systems except agents operating in the system do not have the same internal structure and decision making architecture. This difference creates new challenges mainly because with heterogeneous systems there is often interdependence between agents. These challenges include benevolence versus competitiveness, stable versus evolving agents, resource management, social conventions and roles. The issues raised with homogeneous non-communicating MAS still apply, though the modelling of other agents' states becomes much more difficult as each different type of agent must be modelled in every other to allow for this to happen.

This issue of getting different types of agents to predict each other has become a major research topic. One interesting approach can be seen in [34] where multi-agent coordination is achieved not through explicit communication but through the observation of agent behaviour. This creates a lack of dependency on communication (a major benefit that will be touched upon later in this chapter) as well as the ability for agents to coordinate despite not speaking the same language. Though possibly not as efficient as the techniques used for homogeneous non-communicating MAS, it shows a novel technique for this major obstacle for multi-agent coordination.

2.4.3 Homogeneous Communicating Agents

Homogeneous communicating MAS are identical those described in section 2.4.1 except they can pass information between each other. This communication can be done in an agent to agent fashion, or it can be accomplished via a centralised database like a blackboard. Aside from the need to model agents' states (which is less of an issue in communicating systems) the issues brought up with non-communicating systems still apply. In addition other research topics include distributed sensing and communication content.

A good example of a homogeneous communicating MAS can be found in [21] where a group of homogeneous robot agents communicate via both blackboard and explicit agent to agent communication to concurrently transport objects. Results showed that by utilising such a system the cost was less than that of their control system, the coordination-based cooperation protocol (CCP).

2.4.4 Heterogeneous Communicating Agents

Heterogeneous communicating multi-agent systems are the most complex MAS. These systems are extremely powerful because of the amount of specialisation that agents can attain but therefore complex because of the issues involved with getting such a wide range of agents to interact. In addition to the issues brought up in section 2.4.2, some of the issues described in [68] that occur in communicating systems include understanding each other, planning communicative acts, negotiation, commitment/de-commitment, collaborative localisation and changing shape and size.

The complexity of this type of MAS is shown in [17] where a mediator agent is used as a buffer between different types of agents in an airline booking system. In addition it utilises reinforcement learning to continually improve the ability of the mediator to translate between agents. This is an interesting solution to the problem of getting different types of agents to communicate however it creates a link in the system to which all other agents depend. This becomes a problem in multi-robot applications and will be explained in section 2.6.

2.5 Tools

Multi-agent systems are extremely powerful and consequently can be very complex to develop and implement. There are currently a number of software development environments currently available that ease this process. The Foundation for Intelligent Physical Agents (FIPA) has set up a number of general guidelines for MAS and in particular its agent communication language (FIPA-ACL) has been widely adopted. Some of the most popular agent development tools that implement the FIPA-ACL include JADE, Spyse and JACK (Agent Oriented Software Group).

2.6 Pros and Cons

Multi-agent systems are extremely powerful tools, especially when used in conjunction with multi-robot systems. However despite the benefits shown in this chapter there are a few important challenges, especially when considering the underwater environment.

The first issue has to do with complexity. Though heterogeneous MAS are the most powerful of all MAS, they add an unnecessary level of complexity. This does not mean that multi-robot systems should be homogeneous to keep things simple; on the contrary, multi-robot systems with many different types of robots are definitely the best solution. However, the agent architectures used to control these robots could remain homogeneous thereby allowing for the simple coordination of diverse assets. This problem is illustrated in [17], is briefly described in section 2.4.4. If a mediator is required to get multiple heterogeneous agents to work together, then this becomes an asset that can cripple the system if removed. Multi-robot systems, and in particular multi-AUV systems, need to be extremely robust; the simpler the MAS, the easier it is to distribute intelligence and therefore the stronger the system.

Another major issue with many MAS is their dependency on communication. Though this isn't an issue in most non-robotic domains (and even some robotic ones) it becomes a problem in underwater scenarios where communication is unreliable at best. For instance blackboard systems which require agents to communicate via centralised databases become very hard to implement in the presence of unreliable communication (distributed blackboards are a possible solution). Systems operating a non-communicating MAS on the other hand are less able to coordinate behaviour because of the obvious lack of communication. The best solution seems to be a communicating MAS with fail-safes in place to deal with the eventuality of a loss of communication. The use of agent prediction like the RMM mentioned in section 2.4.1 is a good example of this. The beneficial properties of the MAS remain in place, with the added robustness of dropped communication handling.

2.7 Conclusion

Multi-agent systems are well suited for multi-robot coordination and their benefits far outweigh the challenges mentioned in this section. By critiquing some of the pivotal MAS and the current state of the art it has been shown that these types of systems are

ideal for controlling multiple autonomous robots and that there are viable solutions to the challenge of unnecessary complexity and dependency upon communication. The next chapter will discuss the history and state of the art in the field of collaborative robotics including many examples of multi-agent control.

Chapter 3

Collaborative Robotics

3.1 Introduction

One of the most common applications of multi-agent system technology is multi-robot systems. Agents are used to control individual robots in these systems and there are many examples ranging from the CEBOT system [29] in the 1980's to more current applications like the control architecture presented in [49]. Multi-robot systems have many uses and many benefits over single robot applications both because of the force multiplication aspect as well as their ability to accomplish tasks that a single robot system could not. This chapter will explain the history of collaborative robotics[1] as well as survey the current state of the art ending with a look at the pros and cons of these systems.

3.2 History

Collaborative robotics dates back to the late 1980's when multiple robot systems began to be investigated. This was the junction of two different research areas, singular robotics and DAI, which until this point had not been studied together. The combination of theses two research areas formed the study of collaborative robotics [51]. Some examples of the early systems were CEBOT [29], ACTRESS [7] and ALLIANCE [50] and will be described here.

3.2.1 CEBOT

CEBOT, or *Cellular Robotic System*, was a system that sought to create a dynamically reconfigurable robotic system (DRRS) which is a multi-robot system made up of *cells* that can reconfigure themselves to create the most functional robot for the given task. Each cell has one mechanical function and there are three different levels: level 1 consists of joint cells, level 2 consists of branching cells and level 3 consists of work cells. In theory if all these cells can work together and reconfigure for each specific

[1] It is important to point out that the term *collaborative robotics* is used in this thesis because it is a widely used industry term for multi-robot systems. It does not necessarily follow the semantic definition of collaboration presented in section 2.2. In this research *collaborative robotics* is used interchangeably with *multi-robot systems*.

task they should be able to create the optimal robot for each goal. In order to accomplish this, there are *master cells* that create subtasks and coordinate the other cells. Communication is kept to a minimum by allowing each cell to model the behaviour of others and thereby make predictions [15].

CEBOT was initially designed as a way to customize a single robot to a specific task. This had a lot of merit and soon the technique was applied to not just subsections of a single vehicle but to multiple robots as well. In [14] cells are replaced by separate robots with the goal of coordinated movement. Using a hierarchical, reactive control architecture the robots were able to successfully navigate using obstacle and collision avoidance. There are inherent problems with solely reactive architectures however and these issues will be described in more detail in section 4.5 where robot control architectures are addressed.

3.2.2 ACTRESS

Another early system was the *Actor-based Robots and Equipments Synthetic System*, or ACTRESS. This system utilises the multi-agent system Actors [1] (described in section 2.3.1) to create a system of *robotors* which are autonomous components (both autonomous robots and other units, including assembly robots, simulators, sensors, etc.) that communicate with each other. Figure 3.1 shows a diagram of the ACTRESS system.

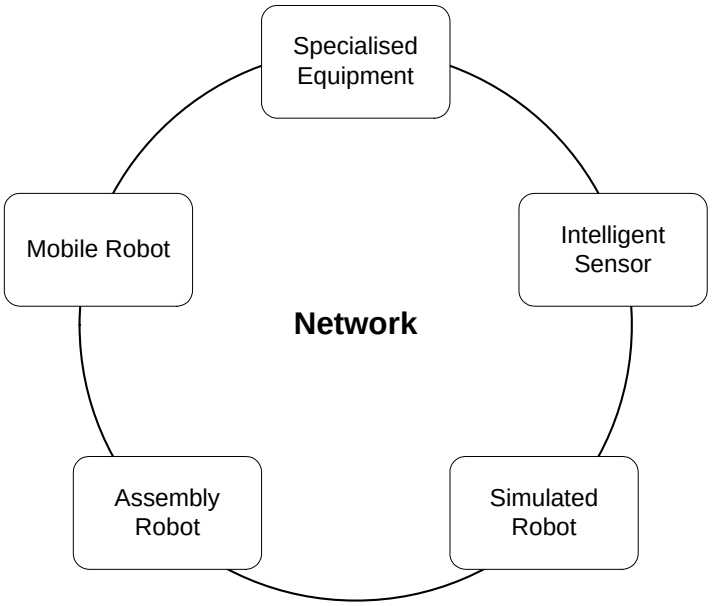


Figure 3.1 Concept of the ACTRESS system.

By passing messages to each other via wired and wireless communications the members of the system are aware of each other and can work together towards a goal. This was one of the very first attempts at creating a collaborative robotic system.

Though a successful system, ACTRESS suffers from the same problems as the Actors MAS that it was based on, namely the fact that it is dependent upon good communication between vehicles. Like in multi-agent systems, this is an extremely important issue in the field of collaborative robotics and has been a major area of research. The current state of the art will be surveyed in terms of principle attributes in the next section including this issue of multi-robot communication.

3.2.3 ALLIANCE

ALLIANCE was an architecture developed to coordinate multiple heterogeneous mobile robots using behaviour-based control and a mission based on sub-tasks with dependencies. The architecture is loaded on each robot operating in the system making it fully distributed. It uses behaviours which correspond to high-level task-achieving functions and are triggered by *motivational behaviours* [50]. These high-level functions act as goal selection tools and make decisions for each vehicle based on the requirements of the mission, environmental conditions, the activities of other robots and the local robots' current internal state [50]. ALLIANCE is a very powerful architecture and is extremely robust due to its distributed nature. Its ability to control heterogeneous multi-robot systems is another major factor.

Despite these important benefits one of the setbacks of the ALLIANCE system is that it relies on solely behaviour-based control. These control paradigms are successful at handling simple tasks but falter when attempting more complex actions. A more in depth analysis of behaviour-based control will be presented in section 4.5.1. Despite this issue the ALLIANCE system is a very important multi-robot architecture and this research will share many of its features.

3.3 Current State of the Art

In order to best explain the current state of the art in collaborative robotics, the principle attributes and research topics of multi-robot systems will be explained by summarizing

two surveys of the field; [51] by Lynne E. Parker from the Oak Ridge National Laboratory and [75] by S. Verret from Defense Research and Development Canada.

3.3.1 Biological Inspirations

Collaborative robotics has been extremely influenced by biological research both in vehicle control and coordination. Behaviour-based control, a type of control system where robots are given a list of possible behaviours to execute in a given situation, revolutionized the field when it was introduced and spurred the beginning of numerous research directions. This type of control is usually used in the form of reactive architectures where vehicles act by applying certain behaviours to environmental stimuli. These architectures will be explained in more detail in section 4.5.1.

A good example of a behaviour-based multi-robot system can be found in [37] where a neural net action selection mechanism is used to choose the best behaviour for the robot. Though this research showed that their approach did in fact improve the robots' action selection capability there are still some inherent problems with behaviour-based control that will be explained later in this chapter.

Another common biologically inspired research topic is the use of collaborative robotics to create simulated social colonies or *swarms*. Swarm robotics has resulted from the application of *swarm intelligence* [10] to collaborative robotics systems. These systems consist of large numbers of autonomous robots that can solve complex problems that would be unsolvable with single vehicle systems. These systems are usually modelled after those of ants and bees as well as the flocking behaviours of birds.

An example of swarm robotics can be found in [36]. This work created a robotic ant colony with the task of foraging for food to replenish the collective group energy. The system functions by allowing robots to only know about their colony status (mainly how much *energy* was available) and thereby make decisions on foraging. Foraging consisted of roaming the world and finding *food* which was then returned to the colony and converted into energy to be used by the entire colony. In doing so an ant colony model was effectively produced and it was shown that by utilising such a model it was possible to execute complex behaviours with relatively simple robots. Another example of swarm robotics can be found in [44] which describes the creation of the Swarm-Bot,

a mobile robot with the ability to attach itself to other Swarm-Bots to solve problems that would be otherwise physically unfeasible.

Though very powerful, swarm robotics is most suitable for very large groups of robots with good communication links. Neither of these characteristics accurately describes multi-AUV operations and consequently swarm technology hasn't been used in this research.

3.3.2 Communication

Communication has become another major research area in the field of collaborative robotics. In order for multiple vehicles to coordinate behaviour they must be able to communicate, either implicitly or explicitly. As mentioned earlier in this chapter, implicit communication is when the passing of information is done via the environment, for instance the movement of an object that other robots can sense. Explicit communication is when information is passed directly to other robots, including wireless links, flags etc. In [87] a multi-robot communication system is developed in the form of a network. Each robot is allocated an ID in the form of an IP address and communication is sent explicitly over the network. The work also shows that it can provide collision detection by passing path plans over the network to other robots.

This issue of communication is of particular interest to multi-robot systems that operate in areas where communication can't be guaranteed. In [31] a system to keep AUVs flying in formation was proposed using graph theory to help keep vehicles on track when communication between vehicles was down. The unreliability of communications in the underwater environment was directly addressed in [67] with the creation of the Compact Control Language (CCL) which is a communication protocol aimed at keeping data rates low thereby lowering the chance of lost messages.

3.3.3 Localisation, Mapping and Exploration

Mapping and exploration has been a major research area for singular robotics for quite a while and it's no surprise that it is now a major application of collaborative robotics [70]. The ability of a group of robots to cover significantly more ground than a single robot makes this an interesting and potentially valuable topic. Despite this, the

combination of multi-robot teams and mapping and exploration is a relatively new concept.

A good example of a multi-robot mapping system can be found in [60] where a collaborative robot system was created to map an unknown environment. Using a central controller, the robots in the system make bids on frontier cells (unexplored areas that border explored areas) and using a bid evaluation function the central controller assigns tasks to each of the robots. It was shown that in this way multiple robots could map an unknown environment in a coordinated, efficient way. What is particularly interesting about this research is that it was conducted both in simulation and with real robots in an office environment. Though successful however this system suffers from the same dependency on communication that many of the multi-agent systems mentioned in the previous chapter do. A decentralised approach would be far more robust. Other multi-robot mapping systems can be found in [64,69].

Localisation is another major, but relatively new research topic in collaborative robotics that until recently has been limited to single vehicle systems. Simultaneous Localisation And Mapping, or SLAM, is a technique that uses features in the world to geo-reference vehicle position and aid in the mapping of unknown environments. A good example of a single vehicle SLAM system can be found in [26] where SLAM was used to navigate a vehicle around the wreck of the RMS Titanic. This technology can be extended and strengthened through the use of multiple vehicles. In [74] a multi-vehicle SLAM system is proposed for use with AUVs that functions by passing feature information via low bandwidth acoustic communications between vehicles. This allows individual vehicles to localise using a more complete representation of the world.

3.3.4 Motion Coordination

Motion coordination is one of the most common applications of collaborative robot systems. This research area can be divided up into two main sub-categories: coordinated object manipulation and formation keeping. In coordinated object manipulation multi-robot systems the goal is to get multiple vehicles to lift/move objects that could not otherwise be done with a single vehicle. A good example of such a system can be found in [54] where multiple robots coordinate to trap or *cage* an object. To date most of these systems have simplified the problem by only operating on

flat, terrestrial surfaces [51]. This is a limitation despite the obvious benefits of such research.

Formation keeping is another major collaborative robotics issue. This topic has been applied to just about all areas of multi-robotics; ground [55], air [43,85], sea [24,31] and space [39,63,71]. With the exception of ground based formation systems, much of the work to date has been done in simulation due to the expense and logistics involved with getting multiple flying/swimming vehicles operating in the same space. Another issue with formation keeping is how vehicles keep in formation if they are out of contact with one another. This is particularly challenging in the underwater environment as mentioned in section 3.3.2 where the work in [31] posed a possible solution.

3.3.5 Learning

Collaborative robotics has begun to work with learning components to create more and more cooperative control systems. Despite this however, compared to multi-agent systems there has been significantly less research conducted [51]. An example can be found in [64] where a genetic algorithm is used to improve the efficiency of the mapping system posed in [60]. By evolving behaviour selection it was shown that when compared to the original mapping algorithm the learning component could map an area in less time and could expand to incorporate extra robots more easily. Another example of machine learning techniques applied to collaborative robotics can be found in [13]. Here a genetic algorithm was used to evolve the control systems of a swarm of robots so as to allow vehicles to work together to overcome locomotion obstacles.

3.3.6 Homogeneous/Heterogeneous Robots

Like multi-agent systems multi-robot systems can either be homogeneous or heterogeneous. Systems can either be made up of identical or different robots. Unlike MAS, the differences between these two types of systems are less severe. So long as the systems controlling the vehicles themselves are compatible, the robot differences themselves are less important. Arguably the more varieties of robot available the more tasks the multi-robot system can successfully accomplish.

3.3.7 Centralised/Decentralised Systems

In multi-robot systems there are two different types of control, centralised and decentralised. In centralised systems there is a central decision maker that allocates tasks to robots in the collective. Examples of this type of system can be seen in [60,64]. In both systems robots communicate with a central controller or *supervisor* and make bids on tasks. The supervisor then allocates tasks based on the highest bids. Though results in both systems showed efficient coordination, the loss of the supervisor would mean the loss of all coordination. This illustrates a significant weakness in centralised systems.

A solution to the previously mentioned issue is decentralised systems. In these multi-robot systems control isn't limited to one central controller but distributed throughout all vehicles. Robots don't depend on a supervisor to make decisions and consequently are far more independent and robust. Examples of this type of multi-robot system can be found in [4,54]. In [4] a decentralised multi-UAV system was created by replicating a centralised assignment algorithm on each vehicle. In this way each vehicle was capable of making its own task allocations. Results showed that with the same situational awareness this system could successfully coordinate multiple vehicles. This dependency on the same situational awareness is an obvious setback however.

3.4 Pros and Cons

Despite the obvious benefits of multi-robot systems there are some drawbacks about certain approaches. One of the main issues is regarding centralised versus decentralised systems. As mentioned earlier, though good at coordinating actions centralised systems are inherently dependent upon the supervisor which if removed will cripple the system. Systems like ACTRESS attempt to overcome this problem by giving multiple robots the ability to allocate tasks however this illustrates centralised systems' additional dependency upon communication. A large proportion of multi-robot systems need to be able to function in environments that don't contain reliable communications and this kind of dependency is a major setback. Due to these challenges, distributed control systems are a better choice with systems such as ALLIANCE taking full advantage of their robust nature.

Another system that showed that decentralised control is superior to centralised is [42] where multiple mobile robots maintained formation control using only passive acoustic communication. This research compared a number of control systems including classic logic, behaviour-based and neural networks. Their results illustrate another issue with collaborative robotics in that though it performed well in simulation, behaviour-based control did not function well in the real environment. This was due to the nature of behaviour-based systems applying behaviours to specific environmental stimuli. The problem arises however when the stimuli contains noise. In [42] this resulted in "crabbing, serpentine" behaviour from the robots as they were constantly changing behaviours. Though this example is specific to formation keeping, the limitations of behaviour-based control affect other systems as well, including ALLIANCE. These issues of control will be examined further in the control architecture section of the next chapter (section 4.5).

3.5 Conclusion

This chapter has presented a history of collaborative robotics and analysed the current state of the art. It has been shown that many of the characteristics of these systems are well suited to a multi-AUV coordination architecture, namely limited communication handling, heterogeneous robot platforms and decentralised systems. The next chapter will present autonomous underwater vehicles as well as their control architectures and the current state of the art.

Chapter 4

Autonomous Underwater Vehicles

4.1 Introduction

Once limited to military vehicles and expensive research platforms, the autonomous underwater vehicle (AUV) has become a recognisable and powerful tool in the realm of sub-sea technology. From the commercial to the military sector, AUVs have changed the way work is carried out in the marine environment. The following sections will give a brief history of AUV technology, survey the common control architectures and finally report the state of the art.

4.2 History

The history of autonomous underwater vehicles begins far before the first AUV was conceptualised. There has been a need for humans to venture underwater for centuries, ranging from salvage to exploration. Despite the high level of scuba and dive suit technology, humans can only dive safely (and economically when referring to commercial diving) to a relatively shallow depth.

Because of this limitation, there have been many technologies that have tried to extend the human presence underwater. Human occupied vehicles (HOV) are manned submersibles that are highly pressurized to allow for operation at extreme depths. These types of vehicles have famously been used on expeditions to the bottom of the Marianas Trench and the discovery of the RMS Titanic. However, due to the human cargo, they are still very dangerous, have a short dive time and are consequently very expensive to operate. The next logical step was to remove the human from the vehicle which resulted in remotely operated vehicles, the first step towards what we now know as AUVs.

4.2.1 Remotely Operated Vehicles

A remotely operated vehicle (ROV) is an unmanned submersible that is controlled by a human pilot from the surface via a tether called an *umbilical*. Though they've been

around since the 1950's, ROVs became common in the commercial market in the 1970's during the phase out of HOVs. They became famous in 1966 when the US Navy vehicle CURV (Cable Controlled Underwater Recovery Vehicle) was able to retrieve a lost hydrogen-bomb from an aircraft that crashed off the coast of Palomares, Spain. As ROV technology advanced, including the addition of better batteries, more efficient motors and the addition of onboard computers, they became far more efficient and far more useful. Today there are hundreds of ROVs in use all over the world ranging from the work class Hercules ROV operated by Subsea7 to the Offshore Hyball inspection vehicle owned and operated by the Ocean Systems Laboratory at Heriot-Watt University.

Despite the success of ROVs in the commercial and military markets, there was still a desire to eliminate the human from the equation entirely. ROVs require a large support team and deployment vessel to operate. This can be very expensive and is not realistic for many sub-sea operations. In addition the umbilical linking the ROV to the pilot on the surface is a major limitation in environments like offshore oil fields where there are many hazards that can pose a serious danger. A new type of vehicle was needed that could perform the duties of an ROV but without the limitation and cost of a complicated support team.

4.2.2 AUV Evolution

Like ROVs, AUVs have been around for quite a while despite relatively recent notoriety. One of the first AUVs was the *Self Propelled Underwater Research Vehicle*, or SPURV, developed in the Applied Physics Laboratory at the University of Washington in the early 1960's [47]. The SPURV vehicle could run autonomously at up to 3 km depth and navigate at angles of up to 50 degrees. Communication between the surface and the vehicle was done acoustically. After a number of years worth of scientific study and data collection, the SPURV vehicle was surpassed by the SPURV II [46] vehicle in the late 1970's, which was used to research the dispersion of submarine wakes.

The US Navy became interested in AUV technology in the 1970's after the sinking of two nuclear submarines, the USS Thresher and the USS Scorpion. There was a need for an autonomous recovery vehicle and because of this the Naval Ocean System Center

began work on AUSS, or Advanced Unmanned Search System [38]. The AUSS could dive to 6 km and could transmit video data via an acoustic communication system. It was launched in 1983 and made over 100 dives while in operation.

In the 1990's there was a surge of interest in AUVs. The Sea Grant AUV lab at the Massachusetts Institute of Technology developed a number of Odyssey vehicles in the first half of the decade that were capable of depths up to 6 km and were used in experiments both under ice and in the open ocean [77]. The Woods Hole Oceanographic Institute (WHOI) was also conducting research in AUVs at this time. The Autonomous Benthic Explorer (ABE) was developed for deep water substrate surveys. It could dive to 5 km and operate for up to 34 hours on a charge. WHOI's most famous AUVs are probably their Remote Environmental Monitoring Units vehicle (REMUS) now being produced by Hydroid LLC. The Ocean Systems Laboratory owns one of these AUVs and it will be formally presented in section 6.3.3.

4.3 Applications

Currently, there are three main sectors actively pursuing AUV technology: commercial, military and scientific. The commercial sector is dominated by the oil and gas industry and AUV technology is split into two types: vehicles for long distance data gathering and vehicles for repair and manipulation of sub-sea structures. AUVs are particularly good for this type of work because unlike ROVs they have no umbilical that can get tangled with risers and pipelines. This allows for safe and productive use which is a major economic benefit when the cost of ROVs and their support ships can easily cost hundreds of thousands of pounds. Currently the main application of AUV technology in the offshore industry is in pipeline tracking. In these missions AUVs are used to locate and track pipelines on the substrate and identify any possible leaks or faults. This is particularly important in the aftermath of large storms like the recent Hurricane Katrina when pipelines can easily have moved by large distances. In addition research has been conducted in autonomous riser inspection and the technologies are rapidly being adopted offshore.

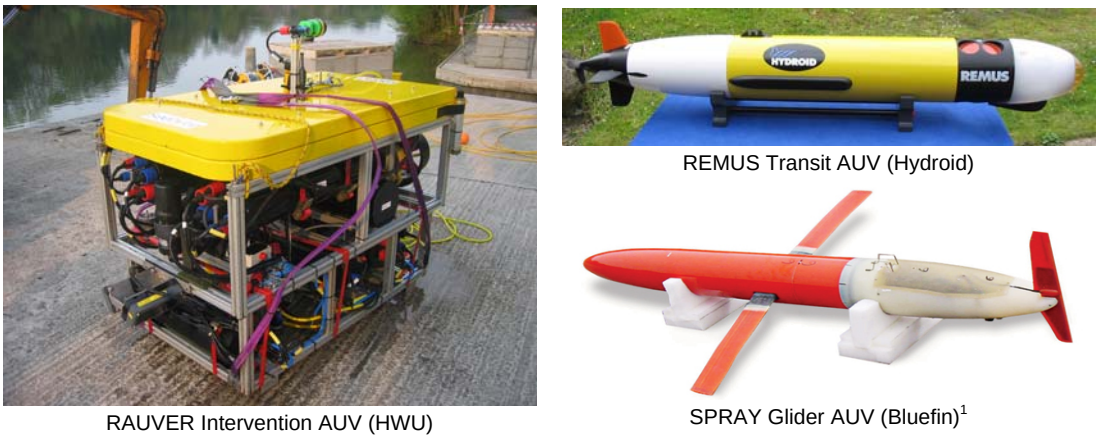
Despite the many ways that AUV technology can benefit the commercial sector, acceptance has been slow. In contrast, the scientific community has moved forward considerably with the technology. Because they tend to have significantly lower

resources than the commercial and military sectors, scientific AUV research has centred around small economic vehicles such as REMUS. Some applications of AUV technology in the scientific sector include oceanography sample collection, exploration of hydrothermal vents and seafloor image mosaicing.

Military AUV research has been moving forward steadily for quite some time. The US Navy has been working on an AUV system called the Long Term Mine Reconnaissance System (LMRS) which focuses on the mission of mine countermeasures, a major application of AUV technology. In addition, the REMUS vehicle was used during the recent conflict in Iraq for both reconnaissance and mine countermeasures (MCM). In these missions an area is searched for potential mines which are then identified, classified and destroyed. This is an extremely important capability for any navy and is arguably the most common application of AUV technology in the military today. Other military applications of AUV technology include Intelligence, Surveillance and Reconnaissance (ISR), Anti-Submarine Warfare (ASW), Inspection/Identification (ID), Communications / Navigation Network Node (CN3) and Payload delivery, among others [73].

4.4 AUV Types

Currently there are a number of different categories of autonomous underwater vehicles. The main three types are torpedo-shaped *transit* AUVs, hover-capable *intervention* AUVs and ultra low-power *gliders*. Examples of each can be seen in Figure 4.1.



RAUVER Intervention AUV (HWU)

REMUS Transit AUV (Hydroid)

SPRAY Glider AUV (Bluefin)[1]

Figure 4.1 Different types of AUV: intervention, transit & glider.

[1] Photo courtesy Bluefin Robotics

The most common type of AUVs is the transit variety. These vehicles are built like torpedoes and *fly* through the water using a propulsion system and control surfaces to adjust trajectory. Because of their speed they can cover large distances and are widely used for applications like sidescan sonar surveys. Examples of this type of vehicle include Geosub (Subsea 7), Autosub (Southampton Oceanography Centre), the Bluefin 9, 12 and 21 vehicles (Bluefin), and REMUS (Hydroid).

What transit AUVs lack is the ability to hover in an exact position and move in multiple degrees of freedom. Intervention AUVs are essentially ROVs with onboard intelligence and are made to investigate and interact with targets that require position control. These types of vehicles are not as fast as their transit counterparts however they can accomplish tasks that would be otherwise infeasible including detailed ship hull inspection and marine installation intervention. There are significantly fewer intervention vehicles in operation due to their complexity however examples of these vehicles include HAUV (Bluefin), RAUVER (Heriot-Watt University) and Nessie III (Heriot-Watt University).

In addition to the more common transit and intervention AUV types there is a third less common but no less important type called gliders. These vehicles have no active propulsion system but propel themselves through the water by varying buoyancy from fore and aft and using wings to direct ascent/descent. This lack of thrusters creates a very low power platform and gliders can operate for extremely long missions. Examples of these vehicles include SPRAY (Bluefin) and the Slocum Glider (Webb Research Corporation).

4.5 Control Architectures

The current state of the art in AUV control consists of three different types of architectures: reactive, deliberative and hybrid [30,57]. These three architectures are the standards for almost all collaborative robotics control systems. In the following sub-sections these architectures will be described and examples of each will be given followed by a discussion about the pros and cons of each.

4.5.1 Reactive Architectures

Reactive architectures, also known as behaviour-based architectures, were the catalyst that helped start the field of collaborative robotics. As mentioned in Chapter 3 these architectures are event based systems where environmental stimuli result in certain behaviours. As written in [56], "the real world acts as a model to which the robot reacts based on the active behaviours." Up until their investigation the state of the art was the sense-plan-act (SPA) deliberative approach. Reactive architectures attempted to avoid the Very Hard Problem of planning and world modelling by simply reacting to the world with pre-programmed behaviours [30].

The most well known example of reactive architecture is the subsumption architecture proposed by Brooks in 1986 [11]. In this system there are multiple levels of behaviours; lower level behaviours dealing with survival and upper level behaviours dealing with intelligence. Each layer acts as an individual unit based on a finite state machine with communication between layers via low-bandwidth links. All layers work simultaneously, performing actions depending on sensor data input. Upper levels can take priority over lower levels by suppressing their outputs. Figure 4.2 shows a diagram of this architecture.

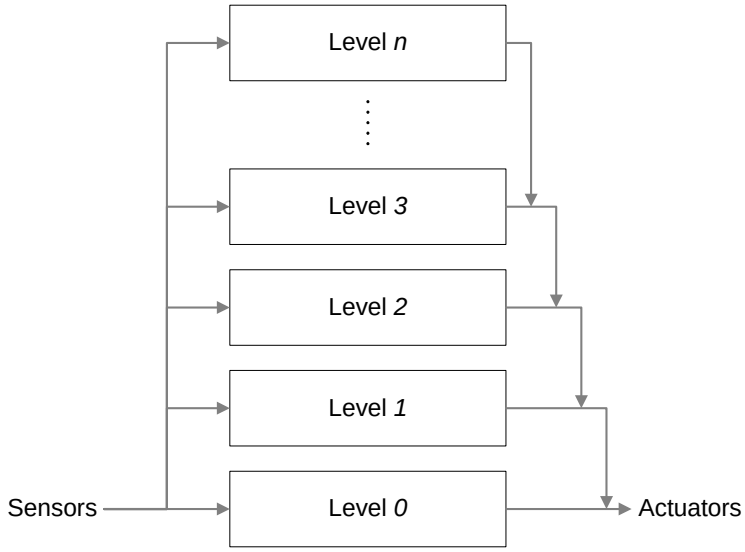


Figure 4.2 Brooks's subsumption architecture.

The subsumption architecture was a revolutionary system in its time because it was a direct reaction to the more common SPA approach. Though it was successfully demonstrated on a robot it seemed to reach a "capability ceiling." In [33] the subsumption architecture was evaluated by applying it to an airplane controller. The

results showed that it lacked the ability to handle such a complex system due to its lack of modularity. More recent examples of reactive architectures can be found in [45,50].

4.5.2 Deliberative Architectures

On the other side of the spectrum from reactive architectures are deliberative architectures which are based around the planning process and are based on the SPA approach. Within deliberative architectures there are two sub types: hierarchical and centralised architectures.

Hierarchical architectures consist of control levels that progressively handle more complex actions, i.e. diagnostics (data collection) up to mission planning. These architectures are usually broken up into three distinct layers: a planning or *deliberative* layer, execution or *executive* layer, and control or *functional* layer [30]. The planning process is done via a planner in conjunction with a world model that consists of a set of high level mission objectives. A plan is generated based on these objectives and is then passed to and executed by the executive layer. The executive layer then sends directives to the functional layer which controls the actuators and the sensor information. This information is then passed back up the architecture as necessary.

One of the first examples of this type of architecture is NASREM (NASA/NBS Standard Reference Model) [3]. Though it is made up of six layers instead of the now more common three, NASREM was one of the originators of the hierarchical deliberative approach and consists of a tree like plan that increases in complexity as it is passed down through the control levels. Three parallel processes control sensor processing, world modelling and task decomposition. As task actions are passed down through the layers sensor data is passed up, all of which is stored in the world model. This architecture was demonstrated on a number of different platforms in the late 1980's and in particular in the MAUV (Multiple Autonomous Undersea Vehicles) distributed AUV control system [2].

Another good example of hierarchical deliberative architectures and their application in AUV control is the Autonomous Underwater Vehicle Controller (AUVC) developed at Texas A&M University [9]. As shown in Figure 4.3 the AUVC architecture consists of the three distinct layers, though they replace deliberative, executive and functional with

planning, control and diagnostics respectively. The planning layer creates a plan in the form of unordered tasks. This plan is then passed to the control level which goes through the tasks and attempts to complete all of them. The diagnostic level consists of sensor and runtime data and can alert the control layer if a fault is detected.

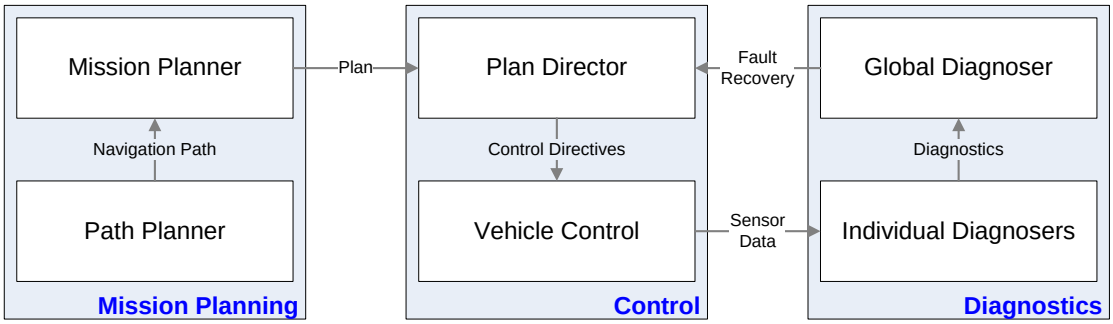


Figure 4.3 AUVC software architecture.

Centralised deliberative architectures consist of a central control unit that all other systems, or expert agents, communicate with via message passing. The central control unit acts as a supervisor and coordinates all action within the system. No expert agent can make a decision on its own, and must go through the central control module before it can execute an action. Having one central controller in the architecture is a possible limitation in the same vein as that of centralised multi-robot systems (see section 3.3.7) because of the inherent dependency on one module. The task control architecture (TCA) [61] is a good example of such an architecture.

4.5.3 Hybrid Architectures

An alternative to the more classic reactive and deliberative architecture approach is hybrid architectures which take deliberative and reactive components and merge the best parts into one control system. As said in [56], "deliberative elements are used for obtaining a system with a predictable function and relative elements are used for obtaining a quick response action to situations that the system is not able to predict." Like deliberative architectures these systems are divided into three distinct layers: a high level deliberative layer, a mid-level executive layer and a low level functional layer [57]. The difference between hybrid and deliberative architectures is that in hybrid architectures the functional layer is made up of reactive, behaviour based modules.

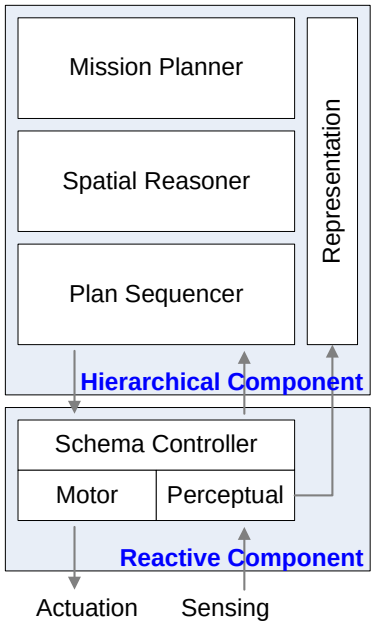


Figure 4.4 Schematic of the AuRA hybrid architecture.

A classic example of a hybrid architecture is the Autonomous Robot Architecture (AuRA) [5,6]. In this architecture a deliberative system is used to plan and coordinate behaviour which is then executed by a behaviour-based, reactive system. The schematic of this system can be seen in Figure 4.4. The top level of the AuRA is controlled by a deliberative system containing a mission planner, spatial reasoner (navigator) and plan sequencer (pilot). Below this deliberative layer is a reactive system that executes the behaviours specified by the plan sequencer.

One of the most important attributes of the AuRA, and of all hybrid architectures, is that once control has been handed over to the reactive layer it isn't passed back to the deliberative layer until either the behaviour is completed or a problem is detected. In this case control is passed bottom up through the deliberative layer so that a re-plan by the mission planner is the last option available.

Because of the fact that hybrid architectures take the best characteristics from deliberative and reactive architectures they are currently the most common choice for AUV systems. An example of such a system is the Intelligent Task-Oriented Control Architecture (ITOCA) [57] currently under development for the US Navy funded semi-AUV, SAUVIM [84]. Other examples of hybrid architectures include [72] where the AuRA was modified and applied to an AUV and [27] where a hybrid architecture is proposed with the aim of coordinating multiple UxVs.

4.5.4 Pros and Cons

Though all of these architectures have been successful there are some important setbacks in some of the approaches. Reactive architectures, despite being founded on the basis of biological systems have a number of issues. Most importantly, though they simplify the SPA approach, they lack the ability to handle complex problems. In addition because behaviours can pursue goals with little concern for other behaviours they can interfere with each other resulting in the robot becoming unpredictable, as shown in [42]. This lack of high level planning becomes a major setback when attempting to use reactive architectures to control a mobile robot that will be operating in unknown, potentially dangerous/hostile environments.

One of the biggest downsides of deliberative architectures (both centralised and hierarchical) is that missions have to be planned by the high level planner modules. This becomes a problem when unexpected events occur. Say for instance a plan calls for a vehicle to navigate to a certain point where it will rendezvous with another vehicle, but on the way to this point it is discovered that it is unreachable. This would result in the mission having to be re-planned or modified in the deliberative level. This is both computationally intensive and dramatically slows decision time, which is an issue in most coordinated robot operations. Though they have the ability to plan at a high level, they don't have the reactive behaviours to handle simple unforeseen events.

Hybrid architectures take the high level planning abilities of deliberative architectures and couple them with the behaviour-based execution of reactive architectures. Rather than requiring the deliberative level to handle all control, hybrid architectures use reactive modules to solve problems such as obstacle avoidance and vehicle control. This is arguably the best of both worlds because it limits the need for constant re-planning while allowing for robust execution of simple tasks. Because of these benefits hybrid architectures are currently the most common choice for AUV control.

4.6 Current State of the Art

Compared to other mobile robot technologies autonomous underwater vehicles are still relatively new. Due to the complexity of operating underwater and in hazardous environments AUVs and AUV systems have been kept as simple as possible to avoid unnecessary complications. Though some work has been done in simulation, currently

most in water applications consist of only one vehicle. There have been some recent examples of multiple vehicles in the water together however coordination has been minimal at best.

4.6.1 Communication

One of the major limitations in AUV systems is the lack of dependable underwater communication between vehicles. At present, the only realistic form of communication is via acoustic modem and due to the relatively low speed of sound in water (1500m/s) communication is extremely delayed and not very reliable [82] (optical and RF data transmission is possible, but only if vehicles are in very close proximity to each other). In order for AUVs to truly coordinate their actions there need to be architectures that allow for high intelligence and low communication. One option posed by [83] is to link the AUVs in the system with a cable. This would allow for easy, high-bandwidth communication however the limitation of locomotion is a considerable problem. A better solution lies in the ability of AUVs to utilise what little communication channels there are and supplement this with intelligent predictions about other vehicles. This approach will be described in more detail later in this thesis.

4.6.2 Multiple Vehicles

Recently there has been motivation to develop AUV systems that incorporate more than just one vehicle. Multi-AUV systems allow for missions that would be otherwise infeasible using only one vehicle and can benefit particularly when heterogeneous vehicles are utilised. These systems are still relatively new but there is research being conducted in both the scientific and military communities.

Representing the scientific community, the National Oceanic and Atmospheric Administration (NOAA) is currently conducting multi-AUV research in a number of its departments including the National Marine Fisheries Service, the Office of Coast Survey and the National Undersea Research Program [40]. The goal is to use multi-AUV technology to obtain oceanographic data from a wide area in one mission; i.e. the more AUVs working together, the more data that can be collected. Another current multi-AUV project is being conducted in the field of adaptive ocean sampling in Monterey Bay, California [28]. By using a group of AUV gliders spread across the bay,

the system hopes to use their data to help analyze and predict ocean processes and create a dynamic model based on this data. A single AUV would not be suited for this task because it requires data to be recorded simultaneously in variable locations, which also eliminates the possibility of using buoys for data collection. Other research in multi-AUV coordination can be found in [59,86] where multiple robotic fish are coordinated in a box pushing task and water polo respectively. Though they both illustrate the benefit of multiple vehicles, these systems rely on surface wireless communication and are therefore limited to solely laboratory experiments.

The military is also looking into multi-AUV technology. One of the main applications of these systems is in mine countermeasures (MCM). In the US Navy UUV Master Plan [73] this type of mission is referred to as "the most problematic of the missions facing the UUV community and the Navy at large" and is therefore an extremely important issue. A major setback to missions of this type is the communication issue, an issue "fundamental to the problem of cooperation" [66]. In order for an MCM mission to function the AUVs need to be able to work together and because communication underwater is difficult there are major hurdles that need to be addressed. Because of this the MCM mission vignette has been chosen as one of the test scenarios for this research.

4.6.3 Control

The current state of the art in AUV control is if-then-else script based mission executives (See Figure 4.5). In these controllers the vehicle is given a chronological list of goals which is executed sequentially. They have the ability to react to certain events by executing secondary scripts and then returning to the main one at their conclusion. Hybrid architectures are most often utilised however the incorporation of the deliberative planning level is still rare. This results in missions having to be planned by the user and then loaded onto the vehicles, thereby restricting the ability to re-plan a mission during execution. Consequently these systems are extremely simple and lack the ability to handle any unforeseen events, dynamically choose the most suitable goal at any given time and optimise mission efficiency. They work well in predictable situations but in the marine environment this is rarely the case.

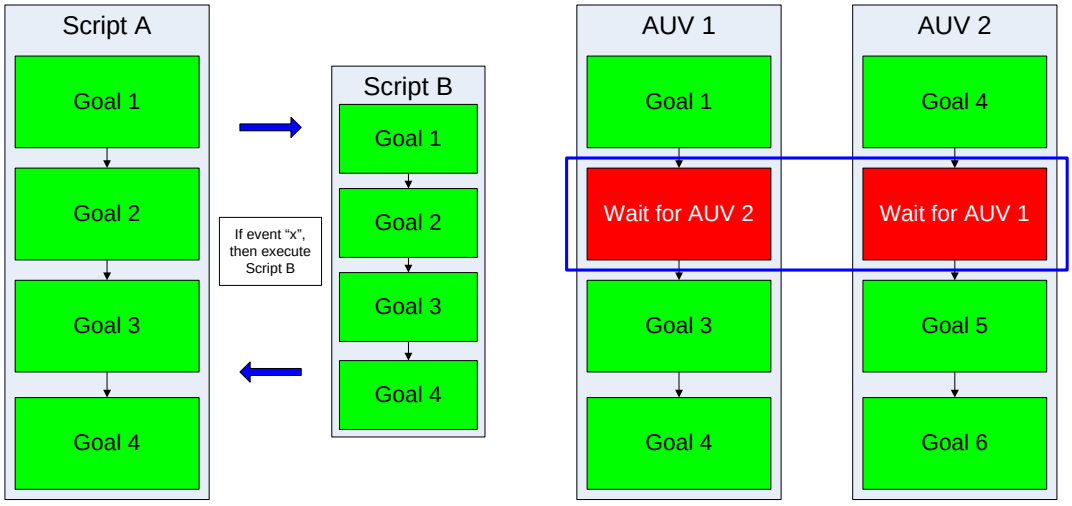


Figure 4.5 Current state of the art in AUV control.

In operations where multiple AUVs are required the current state of the art in control is what is known as *stoplight* systems (See Figure 4.5). In stoplight systems each vehicle is controlled by a script in the same way that was just described however these scripts include synchronisation points or *stoplights* where vehicles pause to exchange information before continuing on the mission. In terms of multi-agent systems this type of system is *open* since vehicles (agents) are aware of each other and utilises *explicit* communication since messages are sent directly from vehicle to vehicle as opposed to via the environment. This type of synchronised communication is called a *rendezvous* [12] and suffers from the same setbacks as the single vehicle scripted approach which are compounded by the issue of getting the vehicles to synchronise, which given the communication limitations mentioned earlier is a challenge in itself.

As AUV technology matures the limitations of these control systems become more significant. An architecture is needed that can both coordinate multiple vehicle actions without a major dependency on communication as well as execute missions dynamically (without a priori user defined goal order) and robustly in unknown environments. Chapter 5 will present a possible solution to this problem.

4.7 Conclusion

Autonomous underwater vehicle technology is a rapidly growing research area and has a large number of useful applications. Of the common control systems for these vehicles hybrid architectures that merge the best characteristics of deliberative and reactive architectures show the most promise and consequently are the most popular

choice for AUV control. Despite this however, the current state of the art in AUV, and in particular multi-AUV, technology is still somewhat primitive and illustrates a requirement for a more robust and dynamic control architecture.

Chapter 5

The DELPHÍS System

5.1 Introduction

Incorporating multi-agent and collaborative robotics technologies this research has created the DELPHÍS[1] system, an architecture that allows for efficient multi-AUV coordination in environments where communication cannot be guaranteed. Because it doesn't rely on centralised task allocation the system remains completely distributed and in addition avoids unnecessary complexity by requiring only one plan for all vehicles in the collective. A custom hierarchical, blackboard based mission representation has been created and missions are executed in a dynamic manner by choosing the most suitable goals while considering the intent of other vehicles. To handle the intermittent communication rates of the marine environment the DELPHÍS system employs real time agent prediction in conjunction with a broadcast (as opposed to unicast) communication protocol.

This chapter will present the DELPHÍS system both in relation to the background research presented in the previous three chapters and in terms of the modules and functionality that make up the system.

5.2 System Design

Based on the review of multi-agent systems presented in Chapter 2 a homogeneous communicating MAS approach is taken where all vehicles in the collective are controlled by the same architecture and coordination is accomplished via inter-agent, asynchronous communication. To handle the unreliability of communications in the marine environment this system also functions like a homogeneous non-communicating multi-agent system by including prediction functionality based on the recursive modelling method (RMM) described in section 2.4.1. Table 5.1 reproduces Table 2.1 with the addition of the DELPHÍS system to show how it compares to the three pivotal MAS presented in Chapter 2.

[1] delphís (**δελφίς**); Ancient Greek for "dolphin", chosen in homage to the coordinated strand feeding bottlenose dolphins of Kiawah, South Carolina that helped inspire this research.

Table 5.1 The DELPHÍS system and how it compares to three pivotal Multi Agent Systems.

	Blackboards	Actors	Contract Net Protocol	DELPHÍS
Open / Closed System	Closed	Open	Open	Open
Communication	Implicit	Explicit	Explicit	Implicit
Cooperation / Collaboration	Collaboration	Cooperation	Cooperation	Collaboration
Homogeneous / Heterogeneous	Heterogeneous	Heterogeneous	Heterogeneous	Homogeneous
Conflict / Task Allocation	Supervisor agent can determine which agent is most suitable for a contested job	Delegation	"manager" can determine which agent is most suitable for a contested job	Agents individually decide which agent is most suitable for a contested job

Despite the benefits of the MAS tools described in section 2.5, this research has decided against their utilisation for a number of reasons. First of all, one of the biggest benefits of tools such as JADE is the implementation of the FIPA-ACL communication protocol. Though extremely powerful, this protocol is designed for MAS in networks and less useful in systems where agents are robots acting in the physical world. In addition this research aims to keep agents simple and therefore it was decided that the functionality of systems such as JADE was not required. In the DELPHÍS system agents (in the form of robots) exist in the simplest form as intelligent entities able to sense and react to both the environment and other agents.

To avoid the limitations of centralised multi-vehicle control presented in Chapter 3 the DELPHÍS system is fully distributed. Each vehicle has an identical copy of the architecture and consequently the same abilities to make decisions. This prevents the situation where the decision making agent or "supervisor" is disabled and the system crippled. It also prevents the dependency on communication that such centralised systems suffer from.

Due to the limitations of both reactive and deliberative control architectures presented in Chapter 4 this study has utilised a hybrid architecture and focuses on an intelligent mission executive that aims to maximise coordination between vehicles while also minimising the need for a re-plan by the mission planner. An architecture [27] currently in development in the Ocean Systems Laboratory has been used as a framework and can be seen in Figure 5.1 with the DELPHÍS system acting as the mission executive in the executive layer (indicated in red). Mission planning is accomplished in the deliberative

layer after which plans are passed to the mission executive (the DELPHÍS system) in the executive layer. A reactive functional layer then handles vehicle control and sensor data acquisition. Data from all layers is stored in a unified world model (this aspect of the architecture is still under development and a local database will be used in this study). Despite the utilisation of this architecture, the DELPHÍS system has been designed to be easily incorporated as the mission executive in any existing hybrid architecture.

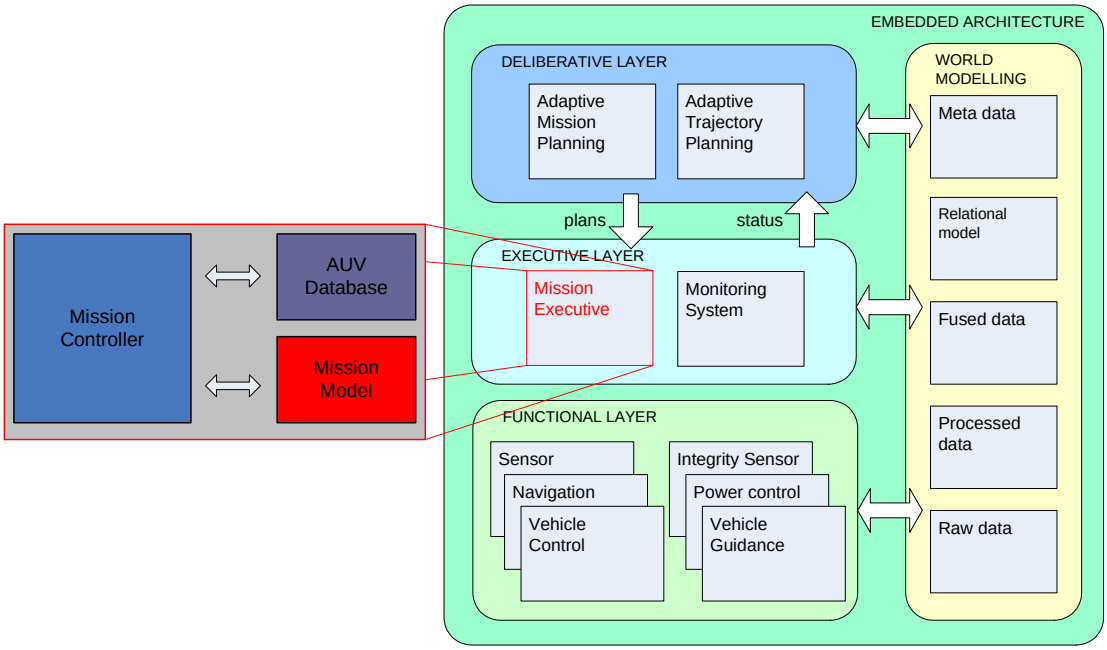


Figure 5.1 Hybrid control architecture highlighting the context of this research in red.

This research has focused on the executive layer in an attempt to keep problems *local* and not *global*. In mission planning *global* problems are those that require the mission planner to either re-plan or repair the mission. This is a very expensive process both in terms of computation and time. In addition it can result in asynchronisation of mission plan data between vehicles. In contrast, *local* problems are those that can be solved by the mission executive. This process is far less expensive and is able to maintain synchronisation of plans between vehicles.

The DELPHÍS system looks to improve upon current systems by significantly enhancing the mission executive to allow for intelligent execution to keep mission problems *local*. To simplify this research the deliberative and world model aspects of Figure 5.1 will be replaced with user planned missions and a local database

respectively. Within the mission executive there are three major modules: the mission model, AUV database and mission controller, as shown in Figure 5.2.

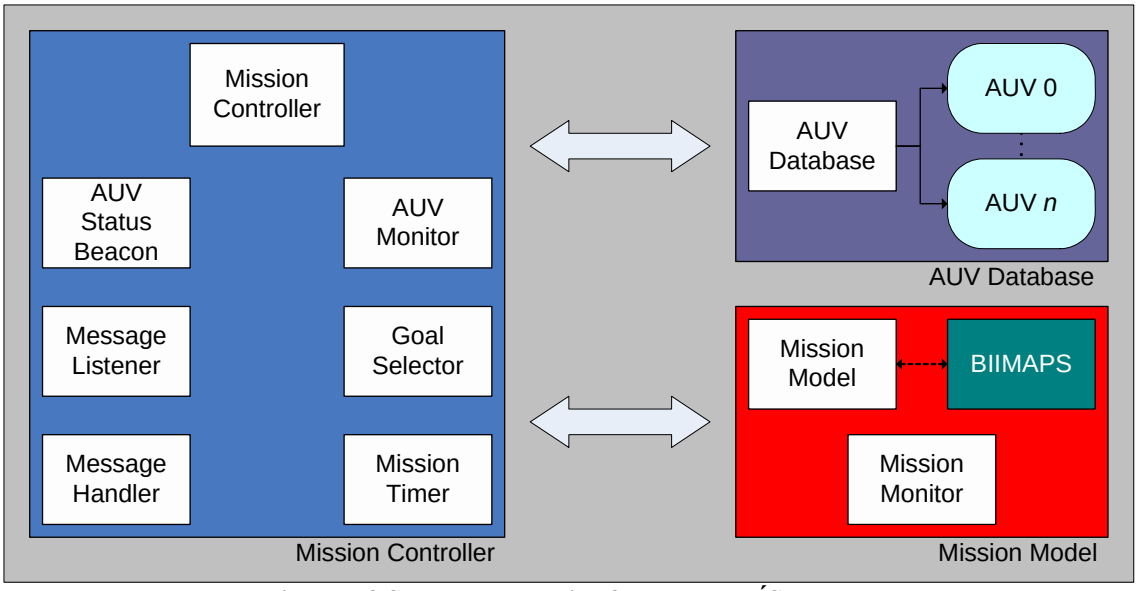


Figure 5.2 System schematic of the DELPHÍS system.

The next sections will explain in detail the sub-modules of this system in addition to it's functionality in terms of multi-AUV operations.

5.3 Mission Model (BIIMAPS)

This research incorporates a custom mission representation system in order to build upon some of the limitations of the current state of the art in mission description. Co-developed with Nick Johnson in the Ocean Systems Laboratory, the Blackboard Integrated Implicit Multi-Agent Planning Strategy (BIIMAPS) [65] system aims to significantly improve the current mission representations as well as allow for specific multi-vehicle applications.

As mentioned in section 4.6 mission plans are currently represented in a number of different ways. The most common options are if-then-else scripts and hierarchical task networks. If-then-else scripts are the current state of the art however they tend to be rather inflexible and unable to cope with unforeseen events. Hierarchical task networks show more promise for flexibility due to the fact that the goals aren't constrained by any order. Examples of such systems can be seen in [50] and [32]. The BIIMAPS system is based on these plan representations and adds the functionality of blackboard systems to

allow for more intelligent, robust behaviour. The following sections will explain the system by breaking it down into its component parts.

5.3.1 Plan Generation

BIIMAPS plans can be generated in one of two ways. Because of their human readable nature (section 5.3.8) these plans can be easily written by a person and loaded into the system. An alternate plan generation technique is to use an automated mission planner. In this situation an adaptive mission planner located in the deliberative layer (shown in Figure 5.1) would automatically generate a BIIMIAPS plan based on high level user entered criteria. An example of such an adaptive mission planner can be found in [53].

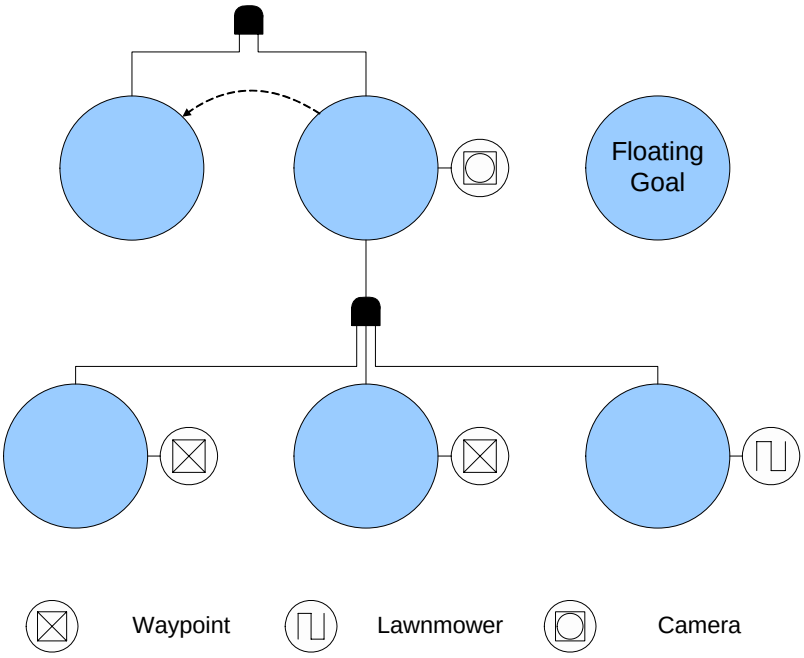


Figure 5.3 BIIMAPS mission representation diagram.

5.3.2 Goals and Sub-Goals

In the BIIMAPS system (See Figure 5.3), mission plans are represented as a tree of tasks, or goals, which are the major building blocks of the system. Goals (represented by large circles in Figure 5.3) can either be atomic (*leaf* goal), or divided into sub-goals. Each goal can be in one of three states: *complete*, *ready* (an agent may attempt to complete the goal) or *locked* (no agent may attempt to complete the goal). The state a goal is in is based upon its conditions, dependencies, and constraints, all of which are described later in this section.

A further sub-state a goal may occupy is *current* which means that that goal is currently being executed, either by the local agent or another working in the system. The current goal for any agent should always be a leaf goal, but all parents of this goal are also considered to be current (thus the root goal is always current whenever the plan is being executed).

5.3.3 Operations

In BIIMAPS, all leaf goals have operations associated with them (represented by white circles in the diagram). An operation specifies the behaviour required of the agent when executing a given goal. For instance if a goal is to navigate to a waypoint, the operation consists of the coordinates. Operations can also be given to non-leaf goals which again specify behaviour. In this case it is a behaviour which should be combined with those of the *current* sub-goal(s). For instance, if a goal is to navigate to a number of waypoints (notated as sub-goals), the super-goal could have an operation that calls for a video recording to be taken throughout the sub-goal operations.

5.3.4 Conditions, Dependencies and Constraints

Each goal in the BIIMAPS system contains a condition that is used to determine when it has finished. For a leaf goal, the condition could be as simple as the completion of its operation. A condition may also specify the receipt of a message from another module. For example, this could be a message from a computer-aided classification/detection (CADCAC) program indicating that a particular object has been detected.

In the case of non-leaf goals, the condition should be the logical combination of its sub-goals. This can be an *and* relationship to specify a compound task, an *or* relationship to specify options for the completion of the goal, or some other logical combination. Whereas *and* relationships are used almost exclusively for disseminating larger tasks to more simple ones, the *or* relationship is used to reduce the need for re-planning by encoding the potential actions capable of completing a task into the plan itself. This helps minimise the need to alter the plan during execution.

The dependencies and constraints of a goal determine its availability based on the states of the other goals in the plan and the state of the world respectively. A goal is

considered to be *ready* when its dependencies and constraints are met, and *locked* when they are not. Dependencies are essentially links to the status of other goals in the plan; this may be a link to the status of single goal, or a logical set of the statuses of many. Constraints function under a similar mechanism, although instead of the states of other goals in the plan, they refer to the state of the world (this information is received in the form of messages from other modules in the system). A similar technique can be found in [32].

Dependencies and constraints can be demonstrated in a scenario where multiple robots are required to cross a mine field. Here *navigate across field* would depend upon *clear field of mines*. Additionally, a goal might have a constraint requiring that the goal only be attempted if there is a certain amount of battery power remaining.

5.3.5 Execution and Completion Locks

The BIIMAPS system can further constrain mission execution through the use of execution and completion locks. An execution lock can be applied to a super-goal indicating that if one of its sub-goals is being executed no other agent acting in the system can attempt another sub-goal. Completion locks function in a similar manner and require that if one sub-goal is executed, the remaining sub-goals must be completed before any other available goals. A good example of the application of this functionality would be two dependent goals such as *goto mine* and *destroy mine*. In this case the vehicle that goes to the mine should also be the one that destroys it.

Although these locks are extremely useful in constraining mission execution they could in theory lead to *deadlock* where two vehicles concurrently execute goals that lock each other. This situation is handled in two separate ways. First, plans are generated in such as way so as to avoid the possibility of deadlock. Mission planners are designed to prevent this and aim to output deadlock-free plans. Second, if a plan capable of deadlock did occur the DELPHÍS system has been designed with mission optimisation techniques that would recognise this situation and only allow the most suitable vehicle to proceed. This functionality will be explained in more detail in section 5.5.6.

5.3.6 Priorities

A priority is a way of indicating the importance of goals in relation to each other. This helps the goal selection phase during the execution of a BIIMAPS plan because it allows more important goals to be given more weight. In this way goals with high priorities are executed first. This is particularly useful in the situation where two sub-goals are related by an *or* logical. Priorities can also be used to suggest the order in which goals should be attempted; if a high priority goal is not available the system will move to one with the next highest priority. This utilisation of priorities prevents the goal sequence from being fixed as it would be if a dependency was used to impose the ordering.

5.3.7 Blackboard Functionality

The BIIMAPS system has been designed so that it contains much of the functionality of a blackboard. Here the plan takes the role of the blackboard with the agents working on the plan taking the roll of the knowledge sources. The system is constantly refreshing itself and as agents post goal completions new goals are made available based on their dependencies. If at any point a goal *x*, which was previously believed to be completed, is found to in fact be incomplete, the system will refresh and all goals which depend upon goal *x* will be rolled back. This functionality is extremely important for multi-AUV mission execution because it allows for goals to be accomplished concurrently and more importantly, for recovery should any conflicts arise between vehicle plans after a period of unreliable communication.

5.3.8 Representation

BIIMAPS plans are represented in the form of XML files which describe the object structure. XML was selected due to its suitability for describing hierarchical structures and it's easily machine parsable and human readable nature. An excerpt of a BIIMAPS plan represented in XML can be seen in Figure 5.4.

```
<goal name="Waypoint 1">

   <dependency ref="Clear Area of Mines"/>

   <condition>
      <completed ref="this"/>
   </condition>

   <operation>
      <waypoint enable="111000" absolute="true" local="true" mode="10">
         <request>
            <coordinate x="0" y="20" z="0"/>
            <heading ref="stdSpeed"/>
         </request>
         <tolerence>
            <coordinate ref="stdPTol"/>
            <heading ref="stdHTol"/>
         </tolerence>
      </waypoint>
   </operation>

</goal>
```

Figure 5.4 Excerpt from a BIIMAPS XML file.

5.4 AUV Database

The AUV database is a module that contains information about all of the AUVs in the mission, a system similar to that being used in [8,41,58]. This information includes static data such as vehicle type and capabilities as well as variable data like battery life, average speed, sensors, and sensor status. Variable data is updated periodically when acoustic broadcasts are received to ensure that each AUV has the most recent status of its peers.

The goal of the AUV database is to allow for intelligent goal selection as well as prediction of intent in the case of a loss of communications. When communications are lost, each vehicle still has to make decisions about the mission, despite not being able to contact the others. By consulting the information contained in the AUV database, predictions can be made using the most recent status of each vehicle. This process will be described in more detail in 5.5.5.

5.5 Mission Controller

The mission controller is the main decision making unit in the DELPHÍS system. Using data from the mission model and the AUV database it coordinates all mission execution decision making within the vehicle. The following sections will explain the mission controller by breaking it into its responsibilities and component parts.

5.5.1 Mission Control Loop

Initially, all AUVs start out with an identical copy of the mission plan, created by either the user or mission planner a priori. Before beginning execution of the mission, the acoustic status broadcast is started so that vehicles acting in the system can register with each other, thereby populating each other's AUV databases. This allows for external AUV information (including position, intention, etc.) to be factored in to local AUV goal selection. In addition the AUV monitor (used for prediction (section 5.5.5)) and mission monitor (used for mission optimisation (section 5.5.6)) module threads are started.

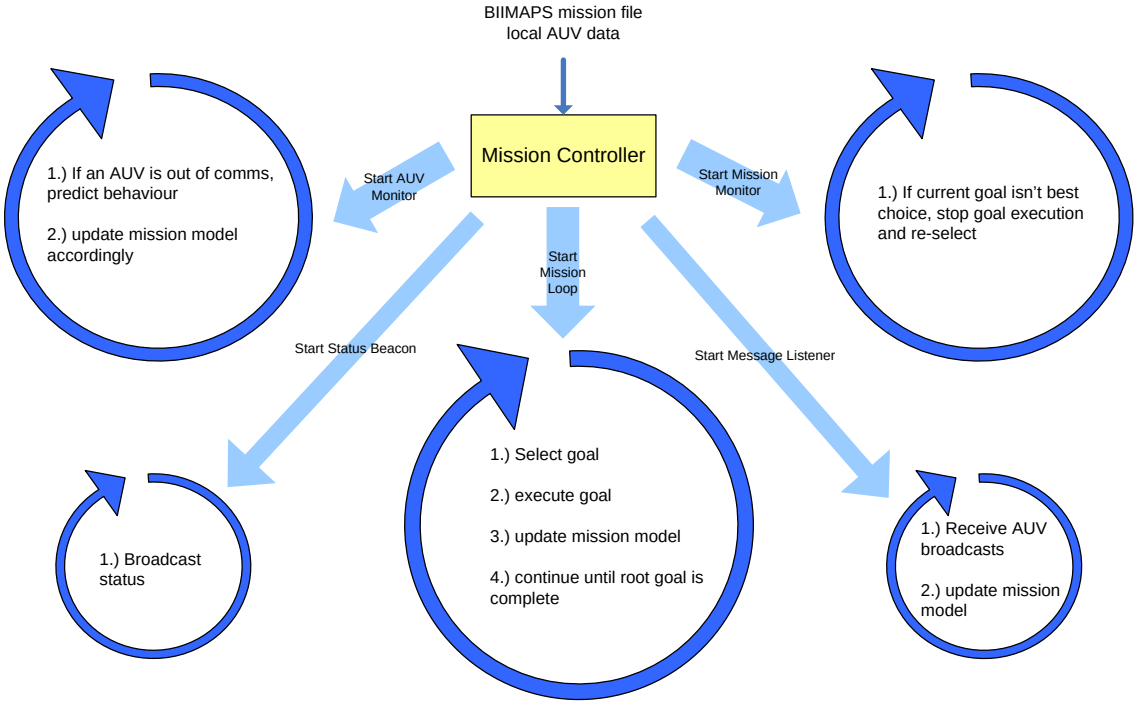


Figure 5.5 Diagram illustrating mission control loop.

Mission control functions as a simple finite state machine during operation. Using three basic states, *IN_PROGRESS*, *FINISHED* and *BREAK*, the system can distinguish between when a goal is being executed, when it is finished and when execution should be halted in response to an event (See section 5.5.6). The mechanisms for goal selection and execution can be found in section 5.5.3.

5.5.2 Communication

Communication in the mission controller consists of both internal and external message passing. Internal communication utilises the OceanSHELL broadcast message system developed in the Ocean Systems Laboratory at Heriot-Watt University (see section

6.3.4). In this way control commands can be easily passed to the vehicle autopilot in addition to allowing for simple interfacing with any OceanSHELL capable vehicle.

Due to the constraints of the underwater domain, external communication between vehicles is generally limited to acoustic transmissions (optical and RF communication are possible underwater but have limited range capabilities). This architecture utilises an *asynchronous* communication system [12] whereby vehicles periodically broadcast status information to all other vehicles in range without the need for acknowledgment. This information includes AUV data with a unique ID number, AUV type (transit, intervention, etc.), current position as well as mission information, consisting of current goal, a list of previously achieved goals and newly discovered targets. Broadcast, asynchronous communication has been chosen over vehicle to vehicle (unicast) transmissions to allow for greater ability to handle intermittent communications, which is common in systems operating in the underwater environment. The XML definition of the acoustic broadcast message can be seen in Figure 5.6 (A slightly modified message was used in the real world trials as will be explained in section 8.4). This process will be explained in more detail in the next section.

```
<data>
   <message name="AcousticCommsSimMsg">

      <field type="Char8" name="AUV_ID"/>
      <field type="Char8" name="AUV_Type"/>
      <field type="Char8" name="Current_Goal"/>
      <field type="Float32" name="X_Coor"/>
      <field type="Float32" name="Y_Coor"/>
      <field type="Float32" name="Z_Coor"/>
      <field type="Char8Array" name="Completed_Goals"/>
      <field type="Float32Array" name="Mines"/>

   </message>
</data>
```

Figure 5.6 XML definition of the acoustic broadcast message.

5.5.3 Goal Selection and Execution

Goal selection works in conjunction with the mission model to select the best possible goal for the AUV to achieve. Utilising the BIIMAPS system, the mission model is able to return a list of the goals in the mission that are available for execution based on their conditions, dependencies and status. This list is then passed to the goal selection algorithm where it is pruned down to a single choice representing the best available task

for the given AUV. If no goal is available the AUV will wait in a holding pattern until one is free or the mission is completed. The execution of goals works by retrieving the goal's operation and issuing the necessary corresponding commands. For a waypoint goal for instance the operation is a coordinate which is then passed in the form of a coordinate request to the vehicle autopilot. Once the goal is executed, its status is updated in the mission model and the loop repeats until the mission is complete. The algorithm is represented in pseudocode in Figure 5.7.

```
(1)    while(there are available goals){
(2)       get list of available goals
(3)       select goal{
(4)          remove goals currently in progress
(5)          remove goals that require payloads AUV lacks
(6)          remove all but the highest priority goals
(7)          choose the closest goal
(8)       }
(9)       execute goal
(10)      update goal status in mission model
(11)   }
```

Figure 5.7 Goal selection algorithm pseudocode.

5.5.4 Multi-Vehicle Coordination

As mentioned in section 5.5.2 multi-vehicle coordination is accomplished via acoustic broadcasts. During execution when broadcasts are received, the information pertaining to the sending AUV is used to update the receiving AUV database. In addition, the current goal and list of completed goals from the sending AUV are passed into the receiving mission model thereby synchronising it. In this way when communication is present all mission models contain the same data across all vehicles.

Vehicles all have the same mission plan, and thus goals can be referred to by ID number only, keeping the size of the message relatively small. New targets are given unique IDs created from the ID of the AUV that discovered them. Targets are then transmitted in the broadcast with both the ID and the coordinates. In the event of a mission re-plan by the mission planner, the current plan can be effectively invalidated (but not removed) and a new one added on to the root goal. In this way goal IDs would always be associated with the correct goal. This situation is not shown in this research however as it is the functionality of the mission executive, not the planner, that is being tested.

Information in the broadcast relating to the mission status is continually built upon throughout runtime. As the vehicles execute the mission their list of completed goals (and discovered targets) continually grows. Due to this, every message received contains the entire mission history of the sending vehicle. This is extremely important for two reasons: first it allows for simple reconciliation of mission data in the likely event of dropped communications, and second, it allows for new AUVs to easily enter the mission at any point during mission execution. With the reception of only one complete message from each vehicle a new AUV has all the information needed about the mission so far.

5.5.5 Prediction

The ability to handle and synchronise data in intermittent communications is not enough for a multi-AUV coordination system, since in many cases vehicles may be out of contact for extended periods of time. In these scenarios it is important to be able to predict the actions of other AUVs to enable more intelligent goal selection.

Currently one of the main research topics in agent prediction is plan recognition systems. In these systems the plan of an agent, be it a software agent, robot or even a person, is inferred by its actions (an example system can be found in [52]). Often using machine learning techniques plan recognition learns to detect certain action states and then predict what behaviour will follow. One of the benefits of these systems is that no prior knowledge of the agent decision making process is necessary. The flipside of this benefit is that these systems require training a priori and are often computationally expensive.

In this system all agents are homogeneous and therefore are controlled by the same architecture. This simplifies the prediction problem greatly and subsequently the Recursive Modelling Method [23] (section 2.4.1) can be utilised. Using this method the local agent's decision making algorithm is used to predict a plausible next step of other agents by passing in their information. This prediction system avoids the infinite prediction loop by only predicting what a certain vehicle is going to do next and not factoring in its own prediction algorithm. This allows for simple yet accurate predictions while keeping computation overhead low.

This architecture handles prediction with an *AUV Monitor* module that keeps track of the AUV database and makes predictions for vehicles when the time since the last communication rises above a certain threshold. Utilising the most recent data from the AUV database (last known position, average speed, current goal, etc.) the AUV Monitor can estimate the current position of the vehicle. In addition it can update the mission model should the predicted position indicate that the vehicle has achieved its goal (in this case the updated status is recorded in the mission model with a *predicted* flag). Additionally, the AUV Monitor can then calculate the next goal by passing the AUV information into its own goal selection algorithm. This is possible due to the homogeneous multi-agent nature of this system where all vehicles are controlled by the same architecture and therefore have the same goal selection algorithm. When communication returns, the mission model is re-synchronised and the predicted values replaced with actual data.

5.5.6 Dynamic Goal Re-Selection

As multiple vehicles are collaborating and constantly updating a common plan, in certain cases the current goal for a certain AUV may become less suitable than another. Rather than force complete execution on any goal attempted, the DELPHÍS system has a *Mission Monitor* module that keeps track of the mission model and constantly checks to see if the current goal is the best possible choice. Should it be determined that this is not the case, the current goal execution is stopped, reset to *ready* and goal selection is run again. In this way the system ensures that each vehicle is always performing the best possible action.

In the cases of intermittent communications, it sometimes occurs that two AUVs are found to be executing the same goal. The Mission Monitor is able to detect this and calculate which AUV is most suited for execution. Because this module is running on both vehicles and each one has information about the other in their AUV database, they will in theory come to the same decision (the exception to the rule is when vehicles are exactly equidistant from the goal, however due to the amount of precision in the position this is unlikely and has so far yet to be observed).

5.6 Conclusion

The DELPHÍS system is a control architecture that has been designed to facilitate efficient multi-AUV coordination in poor communication environments. Taking the form of a distributed homogeneous communicating multi-agent system it employs a hybrid architecture to control the vehicle. Through the use of a custom hierarchical blackboard based mission representation, dynamic mission execution and agent intent prediction the DELPHÍS system aims to maintain vehicle coordination efficiency in face of intermittent communication.

The next chapter will describe how this system was tested in order to evaluate its performance in efficiently coordinating multiple AUVs. Metrics for quantifying efficiency will be explained as well as the experimental setup used.

Chapter 6

Experimental Setup

6.1 Introduction

In order to evaluate the effect of agent prediction, dynamic goal selection, and mission optimisation techniques on multi-AUV coordination the DELPHÍS system was tested in varying conditions against the current state of the art in multi-AUV control. Experiments were designed to illustrate the way both systems coordinate multiple AUV missions by rating each for efficiency in conditions representative of the marine environment. In addition tests were run to evaluate some of the specific tools of the DELPHÍS system including prediction, and the ability to reconcile mission conflicts. The primary goal of these experiments is to show that the functionality of the DELPHÍS system will increase the efficiency of multiple AUV systems as compared to the state of the art.

This chapter will first explain the resources used both actual and simulated, followed by a description of the multi-AUV systems being tested. The metrics used to evaluate the experimental runs will then be explained in detail before finally describing the mission vignettes chosen as representative multi-AUV scenarios.

6.2 Simulated Vs. Real Platforms

This research employs a number of different resources, both actual and simulated. This section will describe these resources as well as their justification. First however, it is important to understand the necessity for simulated platforms in the first place. One of the most important reasons for the use of simulation is that it allows for testing and debugging systems before they are loaded onto actual vehicles. This is paramount both in terms of safety and cost where minor code mistakes can end in disaster. By testing in simulation first, these common errors are cleared out before real missions are attempted.

Another reason why simulation is a necessity in AUV research, in particular multi-AUV research, is that it allows for multitudes of tests to be carried out over long periods of time in controlled environments. In order to get viable experimental data literally

hundreds of trials need to be completed. The sheer scale of this fact makes using real platforms simply not an option, particularly in research such as this where multiple vehicles are required for each trial. Running these experiments in simulation cuts down on the high cost and danger of multi-AUV operations while strengthening the data by controlling the conditions in which the study is completed.

Despite the benefits of simulation however it is important to validate this kind of research on actual platforms as well. Though simulation provides a means for controlled environments for testing it is the *un*controlled nature of the real world that AUVs have to perform in. With this in mind this research aims to demonstrate the functionality proved in simulation through in water demonstrations.

The next section will describe the real platforms used in this research followed by a section detailing the simulations.

6.3 Real Platforms

As mentioned in the previous section, much of the work in this research has been completed in simulation. However, because the DELPHÍS system was designed to be run on real vehicles the aim is to demonstrate its functionality using the AUVs of the Ocean Systems Laboratory. These vehicles and some of the support software will be described here.

Nessie III REMUS

Figure 6.1 AUVs used in the real world validation of the DELPHÍS system.

6.3.1 RAUVER

The Remote Autonomous Underwater Vehicle for Experimentation and Research (RAUVER, Figure 4.1) AUV is a vehicle designed and built in the Ocean Systems Laboratory. It is an *intervention* AUV which means unlike many AUVs it can move in 4 degrees of freedom and function like a traditional ROV, sans operator. RAUVER consists of two pressure hulls, modular tool skid and can accommodate a large variety of sensors depending upon the application, including forward look sonar, Doppler Velocity Log (DVL) and low light video cameras. The RAUVER AUV was not used in this study but is mentioned here as it served as a first step towards the next generation of intervention AUVs in the Ocean Systems Laboratory, namely the Nessie III vehicle.

6.3.2 Nessie III

Nessie III [16] is the third generation of an intervention AUV created to compete in the Student Autonomous Underwater Challenge – Europe (SAUC-E) competition [22], of which it was the 2008 champion. Like RAUVER, Nessie III can move in 4 degrees of freedom and has the ability to maintain position with a high amount of accuracy. Sensors on board include binocular forward and down facing cameras, DVL and acoustic modem. Again, like RAUVER additional sensors can be easily accommodated. Its reduction in size and weight in comparison to the RAUVER vehicle have not come with a proportional reduction in capability resulting in an extremely powerful platform that is extremely easy to work with.

The RAUVER and Nessie III AUVs are relatively slow vehicles compared to other AUVs but due to their design are extremely manoeuvrable and can get in close to objects for investigation, an ability that most torpedo or *transit* AUVs lack. When teamed with a *transit* AUV this type of vehicle is an excellent choice for most multi-AUV missions.

6.3.3 REMUS

The Remote Environmental Monitoring UnitS (REMUS, Figure 6.1) AUV is the industry standard AUV. Unlike RAUVER this is a *transit* AUV which means it *flies* through the water much like a plane through the air. Though it lacks the hovering capabilities of an intervention AUV, it is significantly faster and can cover a lot of sea

in a relatively small amount of time. Sensors include sidescan sonar, DVL (both downward and upward facing) and acoustic modem in addition to a host of environmental sensors (water temperature, salinity, etc.).

In addition to the REMUS vehicle itself the Ocean Systems Laboratory also has a highly realistic simulator able to model the behaviour of the vehicle and output all status information. This allows for very reliable testing to be carried out pre-mission, an extremely useful tool, especially when combined with the ARF system (section 6.4.2).

The REMUS AUV is the choice of most navies throughout the world including the US, UK and New Zealand services among others. This widespread usage and its proven service as an area search vehicle [78] make it an excellent choice for this research, particularly when combined with an intervention AUV such as the Nessie III vehicle.

6.3.4 OceanSHELL

One of the main software tools used in this research is the OceanSHELL system [48]. This is a communication protocol based on the principles of UDP packet broadcast transmission that allows software modules within autonomous vehicles to easily transmit information regardless of their platform or even the language in which they were programmed. The main benefit of this system is that modules can be easily "plugged" and "unplugged" as long as they implement the same messages. As shown in Figure 6.2 this allows simulated and real modules to communicate seamlessly without any modification which is a major benefit when debugging and testing.

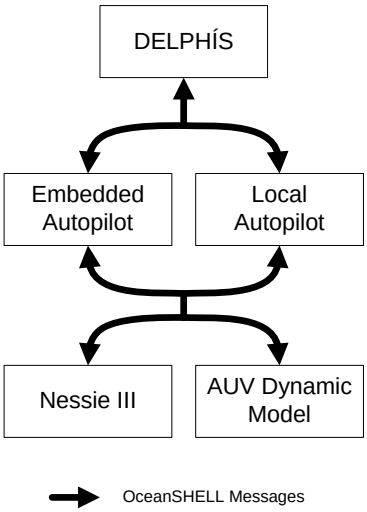


Figure 6.2 Diagram of the OceanSHELL protocol used to link real and simulated modules.

OceanSHELL also provides a suite of control methods (*start*, *stop*, *standby*, *reset*, etc.) that can be incorporated into software module code. Messages types are defined in XML allowing for the simple and rapid modification and creation of new messages. Though OceanSHELL is a C++ program suite, the Ocean Systems Laboratory has also developed JavaSHELL which is the equivalent protocol translated into Java. The result is that both Java and C++ program modules can communicate without the need of native method calls while also gaining the benefit of the low overhead and efficiency of the OceanSHELL protocol.

6.4 Simulated Platforms

In addition to the real platforms previously mentioned this research also employs the use of a number of simulated assets. In Figure 6.2 OceanSHELL and its ability to link real and simulated AUV control modules is presented. In this research the only *simulated* control module used is the AUV dynamic model. This is used in conjunction with all real systems. The AUV dynamic model be described here followed by some of the other simulation tools used.

6.4.1 AUV Dynamic Model

To take the place of actual AUVs in this research a highly accurate dynamic model of the RAUVER vehicle was utilised. This model was created in the Ocean Systems Laboratory and reproduces with a high level of precision the way that the RAUVER AUV handles in the water. Using this in conjunction with the real auto-positioning system from the vehicle allows for extremely accurate control of the vehicle in simulation. Because it responds to the same messages that real OceanSHELL enabled vehicle do, when simulation is complete, the dynamic model can be simply unplugged and replaced with the AUV with very minor if any code modification (Figure 6.2). This results in not only very accurate simulation but also excellent development flexibility. In this study this model was used to represent all vehicles in the efficiency experiments because it exists in software and multiple copies can be easily run (unlike the REMUS simulator which is a hardware tool).

6.4.2 Augmented Reality Framework (ARF)

To visualise the experiments, both during and after the mission, an augmented reality system was used. For this purpose the Augmented Reality Framework (ARF) [20] developed by Ben Davis in the Ocean Systems Laboratory was chosen. This system is an extremely powerful tool that allows users to easily create and simulate autonomous vehicle missions including (but not limited to) vehicle simulation, sensor simulation, world object creation and object interaction. In addition ARF was designed to allow for hardware-in-the-loop testing where real sensor/vehicle data is used in conjunction with simulated modules. Modules like the aforementioned AUV dynamic model as well as actual vehicle code like the autopilot can be easily incorporated into the framework allowing them to work seamlessly together via OceanSHELL (Figure 6.2). In addition by using the modelling capabilities of ARF in conjunction with OceanSHELL simulated vehicles can coordinate with real ones acting in the physical world. This is enormously useful because it allows for real world testing of multi-vehicle systems without the danger of having many untested vehicles in the same space. An image of the ARF displaying one of the experimental runs can be seen in Figure 6.3.

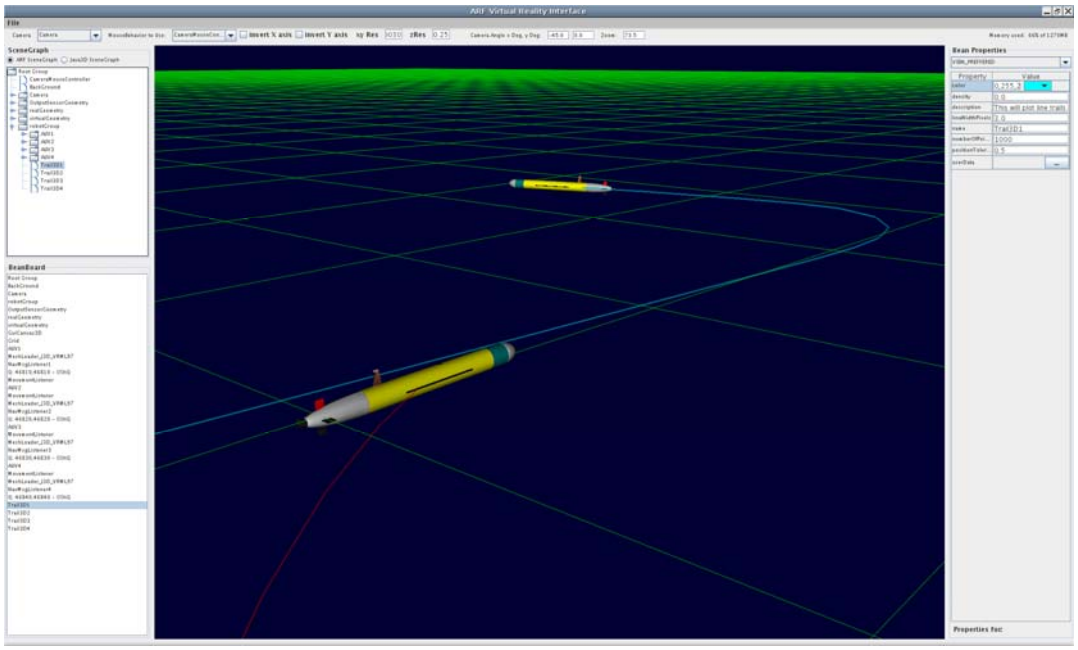


Figure 6.3 A multi-AUV mission as displayed by ARF.

6.4.3 Acoustic Communication

As mentioned earlier one of the main issues in getting multiple AUVs to cooperate is the lack of reliable communication underwater. This unreliability takes the form of both dropped (lost) and corrupted messages. All of theses issues result in the same outcome,

namely data loss between vehicles. In order to properly evaluate a multi-AUV system this situation must be simulated in a controlled manner to see how the system reacts.

Because underwater communication issues come in many forms, simulating a real acoustic communication system can become a very complicated undertaking. Issues like distances between vehicles, message deterioration rates and other factors all have to be calculated to make the simulation realistic. This becomes more complex considering more than two vehicles can be communicating well while others are out of range. For the purposes of this study however it is the lack of information being passed between vehicles that is important, rather than the acoustic communication itself. A special message was created using the OceanSHELL system (Figure 5.6) that contained the information that was to be sent acoustically. This message was then sent on a separate port reserved for and representing a simulated acoustic channel.

To simulate acoustic message loss a module was created that simulated the worst-case scenario where all vehicles are unable to communicate. Messages were prevented from being received by blocking the acoustic message port. This was done in a controlled manner so that the user could enter the percentage of messages that should be let through and the maximum length of time that communication could be down. A random duration in seconds was selected between zero and the entered maximum. Communication would be allowed for the percentage entered by the user (in these experiments the acoustic broadcast interval was set to 3 seconds) and then prevented for the remaining seconds. This is illustrated in Figure 6.4 where a 70% success rate is shown.

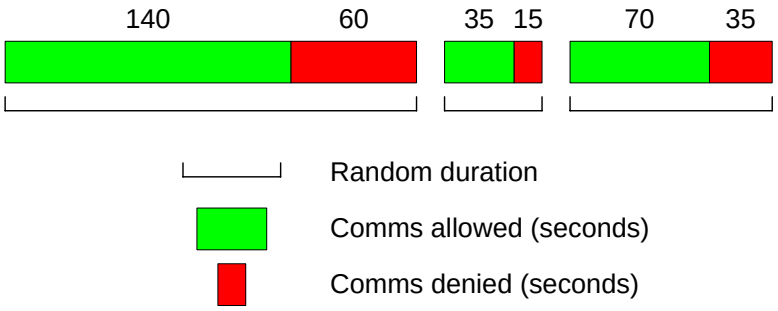


Figure 6.4 Diagram showing acoustic communication simulation.

6.4.4 Target Acquisition

Another aspect of this study that was simulated was the detection and classification of targets, both mines and pipeline irregularities. In practice AUVs use Computer Aided Detection and Computer Aided Classification (CADCAC) systems to identify targets. These are a complex set of algorithms that monitor sensor data to detect and classify unknown objects on the seabed. Despite the proven ability of CADCAC systems, in this research it is simply the presence of objects that is required, not the detection. In this vein a simple detection module was created to run on each vehicle that alerted the AUVs to the detection of simulated targets.

The module operates by first pre-loading the coordinates of the targets including the threshold under which an object should be detected. The simulated detection module monitors the vehicle's position and when it falls within the threshold of any of the targets' position, the vehicle is informed of the detection via an OceanSHELL message. The DELPHÍS system can be easily upgraded to incorporate a more traditional CADCAC system by simply replacing the simulated target acquisition message with the real one.

6.5 Experiments

To evaluate the ability of the DELPHÍS system it was tested to see how its efficiency compared to the current state of the art in multi-AUV control architectures. The tests aimed to show that in addition to being faster and more efficient than the leading system it is also far more robust and able to handle the loss of communication common in multi-AUV missions.

This work was evaluated in simulation by testing it against both single AUV systems as well as the leading multi-AUV coordination system (Script-based stoplight control). In addition two alternate versions of the DELPHÍS system were also tested, each with specific optimisation modules turned off to show their effect. These systems were tested in a range of communication success rates for two different mission vignettes, representing the most common applications of multi-AUV systems today: mine countermeasures & pipeline tracking. Experiments were evaluated for efficiency (defined in section 6.5.2) and the systems' performance was compared.

6.5.1 Systems

As mentioned earlier this work was tested by comparing the performance of the DELPHÍS system with that of both single AUV systems and the state of the art in multi-AUV control. This section will describe these systems in more detail.

6.5.1.1 Single AUV

To date most research with AUVs has been done with single vehicles. This is due both to the infancy of the technology as well as the expense of owning/running multiple vehicles. In order to truly evaluate a multi-AUV control system its performance must be compared to that of a single vehicle. In this study the single AUV control system consists of a vehicle capable of accomplishing an entire mission. For the purposes of this study, this consists of both wide area search and object inspection. The REMUS AUV has these capabilities and is therefore the model for this system.

6.5.1.2 Multiple AUV – Stoplight

Chapter 4 described the current state of the art in multi-AUV operations, namely script-based, stoplight systems. In this study these systems were represented by a heavily restricted BIIMAPS mission plan that functioned like a script. Each mission was essentially broken up in to sections, the number of which dependent upon the number of expected AUVs operating in the system.

For instance in a typical mine countermeasures mission there are two main objectives: search an area and identify mines (and eventually neutralized, though this part of the mission was left out of this study for simplicity sake). To represent a script-based, stoplight system these objectives were each restricted. In a two vehicle operation the mission was broken up so that one vehicle performed the search while the other investigated the targets. As more vehicles were added to the operation, the search was broken up as evenly as possible by the user before the mission, with one vehicle assigned to target identification.

In addition to the limitation of the BIIMAPS plan, all the mission optimisation functionality of the DELPHÍS system was turned off. In this way the current state of the art in multi-AUV operations could be easily tested.

6.5.1.3 Multiple AUV – DELPHÍS

The third system being tested is the architecture created in this study, the DELPHÍS system. Unlike the previous two control systems this system executes the mission dynamically and aims to remain efficient in the presence of less than optimal communication environments. For more details on the specifications of this system please see Chapter 5.

6.5.1.4 Multiple AUV – DELPHÍS (Un-optimised)

To help demonstrate the usefulness of the mission optimisation functionality the DELPHÍS system was also tested with these modules *disabled*. These modules include prediction and dynamic goal optimisation described in sections 5.5.5 and 5.5.6 respectively. This system compared to the optimised DELPHÍS system would hopefully show the benefit of such modules.

6.5.1.5 Multiple AUV – DELPHÍS (Prediction Failure)

In addition to testing the DELPHÍS system with its mission optimisation tools disabled it was also tested with one of its mission optimisation tools functioning incorrectly. In this case the prediction algorithm was set to predict events twice as fast as they were actually happening. The hope was to show that even with incorrect prediction the DELPHÍS system would be able to successfully control multiple vehicles better than the leading coordination system.

6.5.2 Efficiency Metrics

In order to accurately compare these different approaches to multi-AUV coordination, there needs to be a value or metric that can be tested in a controlled experiment. In this work, efficiency was determined to be the most suitable value. Before this can be used as a comparison value however it had to be defined. To do this a number of characteristics were selected that have the most effect on *efficiency* in multi-AUV operations: *mission speed*, *mission accuracy* and *target acquisition*.

In addition to being used to define and calculate efficiency these metrics will also be used later to determine the key performance indicator (KPI) of the system. This will help demonstrate which aspects of multi-AUV missions have the greatest effect on

efficiency and will be discussed in more detail in Chapter 8. These metrics will be explained in the following subsections.

6.5.2.1 Mission Speed

Mission speed is defined as the time required to complete the mission. In this study it is given a value by determining how closely it relates to the expected mission time as recorded by the DELPHÍS system time when run with 100% communication. Thus the mission speed metric *(t)* is:

$$\frac{time_{expected}}{time} = t \tag{1}$$

If the mission time returned is longer than the expected time *t* can range from 0-1, the higher the number the faster the time. If mission time is less than the expected time *t* will be greater than 1. This rewards systems for speed.

6.5.2.2 Mission Accuracy

In this study mission accuracy is defined by the number of goals that have been missed as well as those that have been accomplished more than once. The missed goals metric *(m)* is calculated by subtracting the number of missed goals from the total number of goals and then dividing by the total. The formula for missed goals is:

$$\frac{goals_{total} - goals_{missed}}{goals_{total}} = m \tag{2}$$

Mission redundancy *(r)* is calculated by a similar method except the total number of goals is weighted by 2. This effectively gives goal redundancy half the weight of that of missed goals. This was done because it was deemed that missed goals should affect efficiency more than redundant ones. In addition this is also because a goal can effectively be redundant more than once. The formula for mission redundancy is:

$$\frac{2(goals_{total}) - goals_{redundant}}{2(goals_{total})} = r \tag{3}$$

For both missed and redundant goals the values range from 0-1, the higher the number of missed and redundant goals the lower the value of *m* and *r* respectively. This penalises systems that return missed and redundant goals (though redundant goals are weighted less as just explained).

6.5.2.3 Target Acquisition

Target acquisition *(x)* is defined as the percent of targets that were detected and subsequently investigated. This is calculated by dividing the number of investigated targets by the total number of targets expected to be in the world. The formula used is:

$$\frac{targets_{detected}}{targets_{total}} = x \tag{4}$$

Like the mission accuracy values the target acquisition value ranges from 0-1, the higher the value the more targets that were discovered. Again, this penalises systems for missing targets

6.5.2.4 Evaluation Formula

Mission efficiency is calculated by taking the previous 4 metrics and combining them into the following formula:

$$100(t * x * m * r) = efficiency \tag{5}$$

Using this formula *efficiency* values normally range from 0-100, with 100 being perfect efficiency. An important note is that this value in theory can rise above 100 in the case where the mission time is faster than the expected time. This will be described in more detail in the Discussion in Chapter 8.

6.5.3 Mission Vignettes

This study uses two of the most common applications for multi-AUV systems to compare the different approaches: mine countermeasures and pipeline tracking. As described in Chapter 4, these two mission vignettes are currently the most important

applications of multi-AUV technology to the military and offshore industries respectively and both are being currently undertaken on a regular basis.

As described in section 5.3.1 BIIMAPS plans can be generated either by a high level mission planner or by a human user. Because in this study it is the mission executive that is being tested, not the planner, missions were written by the user to avoid unnecessary complexity. The two scenarios used in these experiments will be described here.

6.5.3.1 Mine Countermeasures

The mine countermeasures mission used in this study consists, like all MCM missions, of an area search with targets to be identified. Also called "Search-Classify-Map, Re-Acquire-Identify and Neutralisation" this type of mission is standard operating procedure for United States Navy Explosive Ordinance Disposal (USN EOD) operations. The area search is done via a lawnmower search pattern that can be broken up into individual legs so vehicles can break up the task. There are 5 simulated targets in the world for the vehicles to discover. A diagram of the mission can be seen in Figure 6.5.

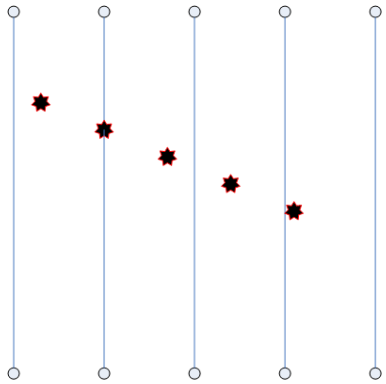


Figure 6.5 Diagram of the MCM mission.

As mentioned in section 6.4.4, in practice targets are discovered by an onboard CADCAC system, however to simplify this experiment a simulated target acquisition system was developed and utilised. The lawnmower legs are 40m long and spaced 10m apart. Compared to most MCM missions this is a rather compact search however due to the number of trials run in this experiment a smaller mission is proportionally faster while remaining long enough to prove the concept.

6.5.3.2 Pipeline Tracking

The pipeline tracking mission used for this study contains many of the same traits of the MCM mission previously described. It consists of three tracks: one low altitude track over the pipe for camera inspection and two higher altitude offset tracks for sidescan sonar.

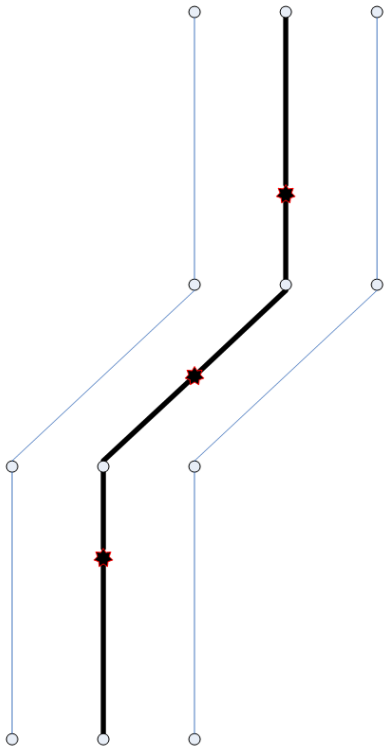


Figure 6.6 Diagram of the pipeline tracking mission.

Each leg has been broken up into sections so that like in the MCM mission vehicles can break up the task. Each sub-leg is 30m long and mission legs are spaced 10m apart. Again, like the MCM mission, this mission is smaller than most pipeline tracking missions to allow for many experiments.

6.5.4 Methodology

As mentioned in section 6.2 it is often necessary to run experiments in simulation before they are demonstrated on real platforms. In this study this was not only a good idea but also very necessary for a number of reasons. First of all because of the large number of variables that were to be tested large numbers of experiments were required to obtain sound data. To run over 1000 experiments with real AUVs isn't a viable option especially since the differing environmental conditions could add an unwanted skew to

the data. This brings up another need for simulation, that of a consistent environment between trials to maintain data comparability. These two benefits in addition to the safety and cost concerns brought up in section 6.2 made the decision to run a majority of the experiments in simulation a clear one.

For each experimental vignette each of the four control systems were tested in simulation using 1, 2, 3 and 4 vehicle operations. For each number of vehicles, a range of communication loss was tested (information on communication simulation can be found in section 6.4.3). For each communication rate (100-10% in 10% intervals) each system was run 10 times. Data was then analyzed and the efficiency value calculated using the formulas described in section 6.5.2.

Despite this requirement to run experiments in simulation when working with systems designed for real vehicles it is important to validate the simulated results on real platforms. The inconsistent environment of the real world that could impede testing is in itself one of the main challenges that can trip up simulation-only systems. By testing this work on real vehicles the aim is to both show that it can handle the hostile environment of, as well as prove that the tests conducted in simulation were valid representations of, the real world. By showing that real world trials return comparable results to the simulated trials the simulations themselves are validated. Therefore in addition to the aforementioned simulated experiments tests were carried out using real vehicles to validate the simulated results. Using REMUS and Nessie III a number of MCM missions were carried out to prove the ability of the system in the uncertain environment of the real world. The REMUS vehicle was chosen for its speed, sidescan sonar and proven ability in the field. Nessie III was selected as the second vehicle due to its relatively small size and its inclusion of an acoustic modem.

6.6 Conclusion

To test the functionality of the DELPHÍS system experiments were designed to evaluate how efficiently multi-AUV missions were coordinated as compared to the state of the art in realistic mission vignettes. In addition certain aspects of the system including agent prediction and conflict resolution were to be tested to demonstrate the ability to handle unreliable communication and other mission run time errors. An evaluation formula was created that factored in 4 metrics to determine efficiency: mission speed,

missed goals, redundant goals and target acquisition. Having justified the need for both simulated an real tests, experiments were to be conducted in simulation and then validated in the water with real vehicles. The data from these experiments will be presented in the following chapter, and then discussed in Chapter 8.

Chapter 7

Results

7.1 Introduction

This research has compared the functionality contained in the DELPHÍS multi-AUV coordination architecture to the current state of the art in multi-AUV control. To compare the systems' performance they were rated using a number of metrics that make up a more encompassing "efficiency" metric as presented in section 6.5.2. These include mission speed, missed goals, redundant goals and target acquisition. The goal of these tests is to compare the functionality of each system in different areas in order to illustrate any enhancement of performance provided by this work. Two of the most common multi-AUV mission vignettes were tested (MCM and pipeline tracking) with different AUV group sizes and varying communication rates.

Graphs will be displayed that summarize the results in terms of efficiency by focusing on the factors used to calculate it, namely goal redundancy, missed goals, mine/target detection and time. Four systems' performance is shown: scripted stoplight, the DELPHÍS system, the DELPHÍS system un-optimized and the DELPHÍS system with prediction failure. The graphs in this section will illustrate how the systems coped as the communication rate was lowered. The data will detail the results for 2, 3 and 4 vehicle systems. Single AUV systems were tested however by definition their coordination efficiency was always 100%. In addition because communication rate isn't a factor for a single vehicle mission, the graphs have been left out. However the difference between single and multiple AUV missions comes into play when mission time is considered. This will be described later on in this chapter.

Data presented is the average of 10 trials. The individual trial data can be seen in Appendix A and Appendix B. In the following graphs the optimised DELPHÍS system is represented in blue, the un-optimised DELPHÍS system is in red, the stoplight system is in yellow and the prediction failure DELPHÍS system is in turquoise.

The goal of these experiments is to show how the four systems rate in terms of the four efficiency metrics individually as well as the combined efficiency function. This will both show how the systems compare for each metric as well how each metric affects efficiency. The results are divided into the two mission vignettes and subdivided into the efficiency metric data. Following the simulated results the real world trials will be presented showing both the AUV/Simulated AUV and multi-AUV tests. This will demonstrate the DELPHÍS system's ability to operate on real vehicles as well as validate the simulated findings.

Results will show that through the use of the DELPHÍS system efficiency and its component metrics are increased compared to the state of the art. In addition the data recovered from the prediction failure DELPHÍS system experiments will show that even with incorrect prediction the system is able to reconcile mission conflicts successfully and remain efficient. In water trials were successful and will show the robust nature of the architecture to work in realistic environments. The following sections will present the results followed by an in depth analysis in Chapter 8.

7.2 Mine Countermeasures

This section will present the data collected from the mine countermeasures mission vignette. As mentioned above graphs will show how the systems dealt with varying communication rates in terms of goal redundancy, missed goals, mine detection, time and finally general efficiency.

7.2.1 Redundancy

The data illustrating how each system handled degrading communication in terms of redundant goals will be shown here. Data is presented as the average number of goals that were achieved more than once and can be seen in Figure 7.1, Figure 7.2 and Figure 7.3.

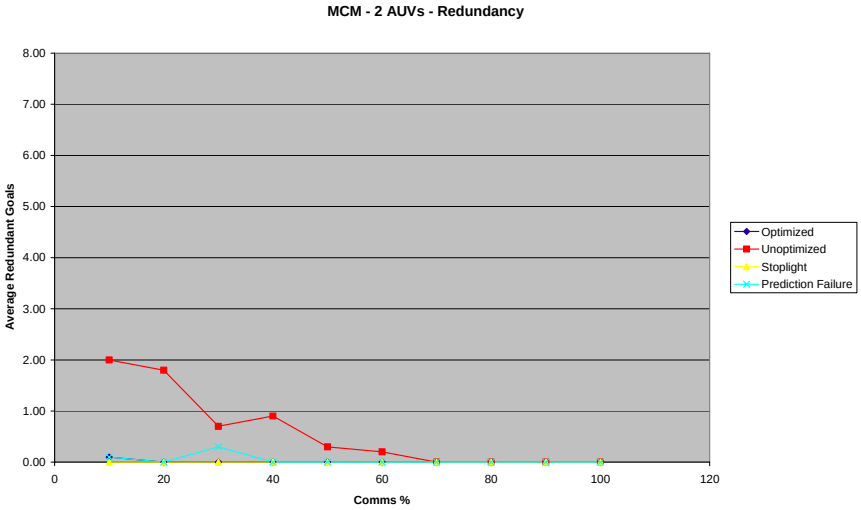


Figure 7.1 MCM redundancy system comparison for 2 AUVs.

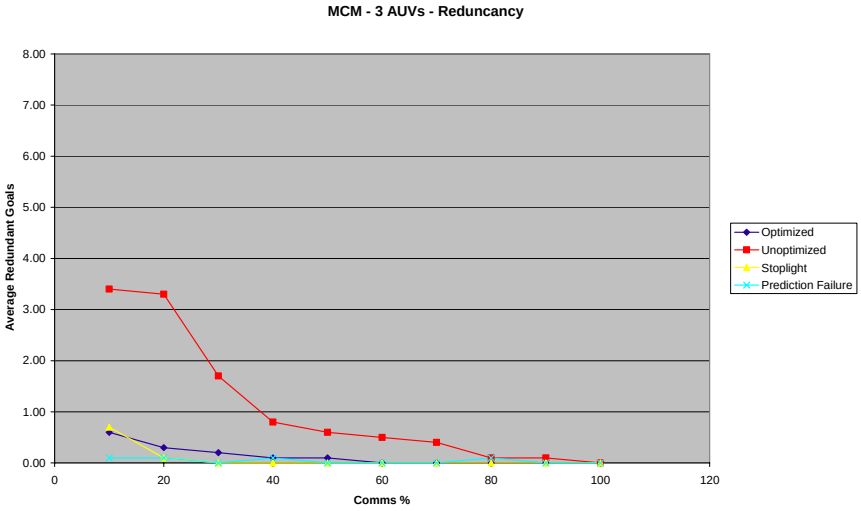


Figure 7.2 MCM redundancy system comparison for 3 AUVs.

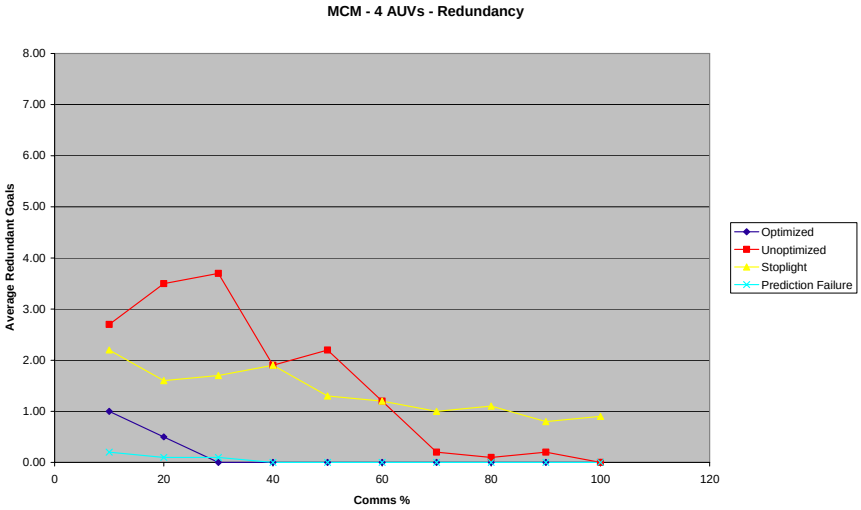


Figure 7.3 MCM redundancy system comparison for 4 AUVs.

The 2 vehicle data shown in Figure 7.1 shows the average number of redundant goals stayed extremely low for the optimised and prediction failure DELPHÍS systems as well as the stoplight system. The un-optimised DELPHÍS system gradually increased in redundancy as communications decreased. The same trends were seen in the 3 vehicle data in Figure 7.2. Similar data was recorded for the 4 vehicle scenario (Figure 7.3) except that the stoplight system had initially high redundancy. This will be explained in section 8.3.1.

7.2.2 Missed Goals

Similar to redundant goals, missed goals are also used to calculate mission coordination efficiency. Here the data showing the average number of missed goals is presented in Figure 7.4, Figure 7.5 and Figure 7.6.

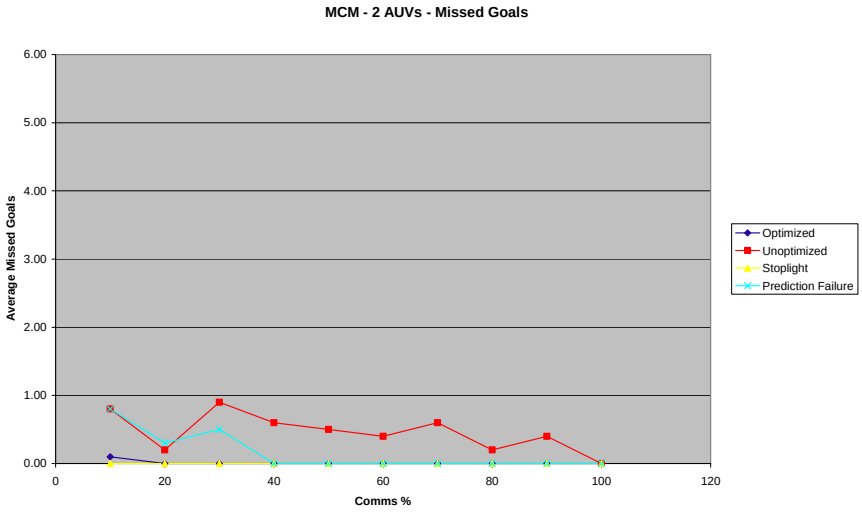


Figure 7.4 MCM missed goals system comparison for 2 AUVs.

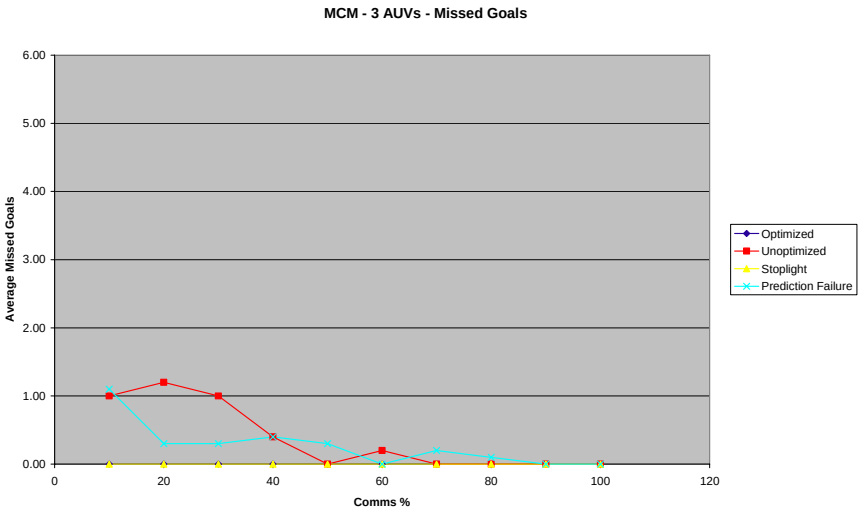


Figure 7.5 MCM missed goals system comparison for 3 AUVs.

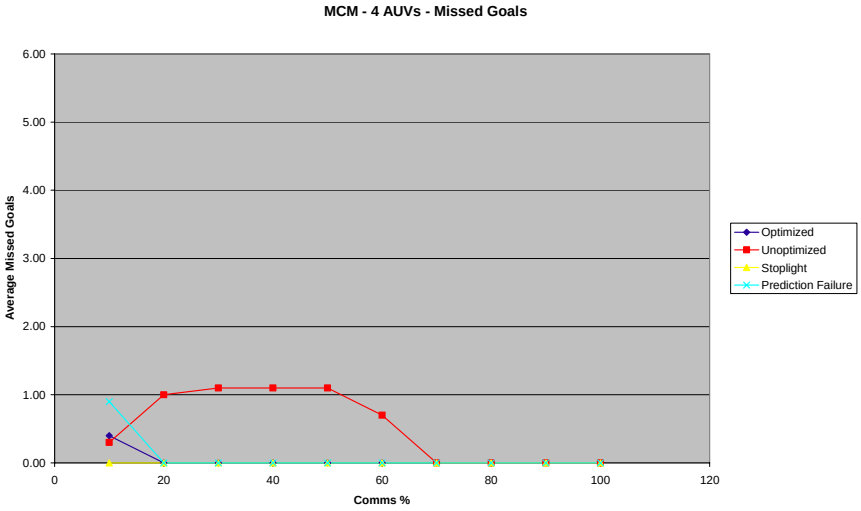


Figure 7.6 MCM missed goals system comparison for 4 AUVs.

In the 2 vehicle scenario shown in Figure 7.4 the average number of redundant goals stayed below 1 for all four systems. The un-optimised DELPHÍS system had more redundant goals than the other systems though there was a minor increase as communications fell below 40%. Figure 7.5 shows the 3 vehicle data where again the un-optimised system recorded more missed goals than the other systems (although the prediction failure system did increase as communications dropped). The 4 vehicle data in Figure 7.6 again showed the un-optimised DELPHÍS system recording more average missed goals than the other systems which recorded virtually none.

7.2.3 Mine Detection

In the vignette being tested there are 5 mines that should have been detected. The data showing the average number of mine detections is shown in Figure 7.7, Figure 7.8 and Figure 7.9.

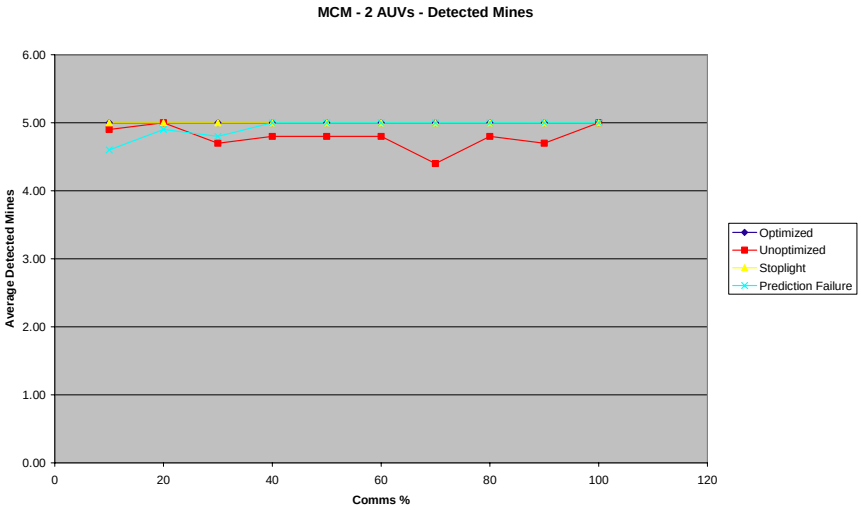


Figure 7.7 MCM mine detection system comparison for 2 AUVs.

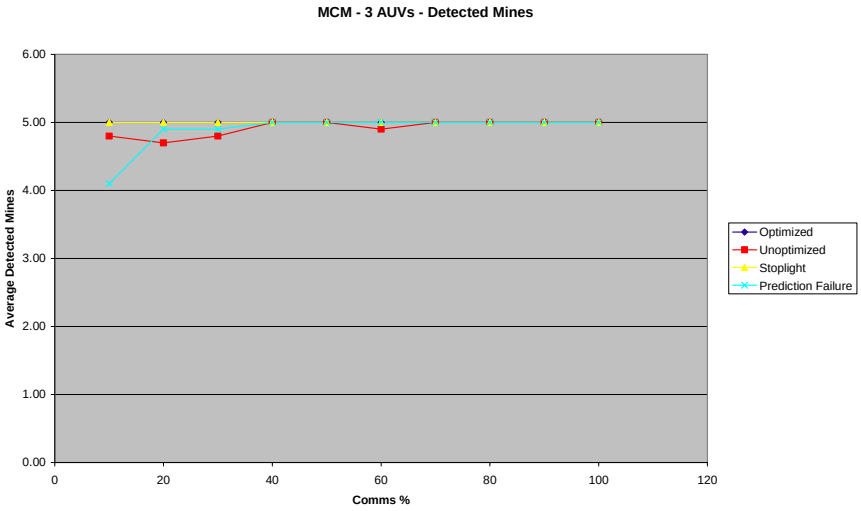


Figure 7.8 MCM mine detection system comparison for 3 AUVs.

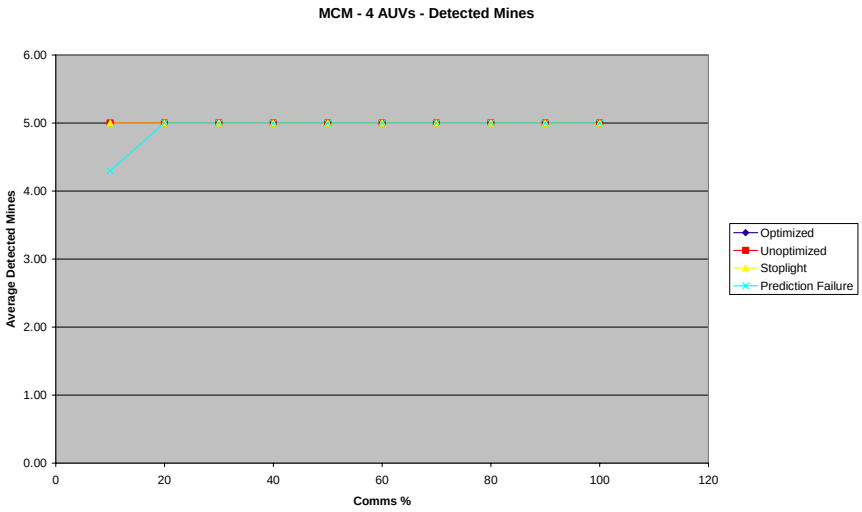


Figure 7.9 MCM mine detection system comparison for 4 AUVs.

In Figure 7.7 the optimised DELPHÍS and stoplight systems were successful in discovering 100% of the mines. The prediction failure system recorded similar data with only a minor drop towards the lower communication rates. The un-optimised system recorded slightly worse data than the other three systems. This trend is seen again in the 3 vehicle data (Figure 7.8) though in the 4 vehicle data (Figure 7.8) 100% of mines were discovered by all systems except the prediction failure system.

7.2.4 Time

Here the time data will be presented. As mentioned earlier single AUV systems are not affected by communication loss. Therefore the average mission time is the same in any communication environment. Table 7.1 shows the average expected mission time for different numbers of AUVs in 100% communications.

Table 7.1 Average MCM mission time in 100% comms.

# of AUVs	Mission Time
1	17:30
2	10:41
3	8:23
4	8:23

The data for multiple AUV missions in declining communication rates is shown in Figure 7.10, Figure 7.11 and Figure 7.12. Communication rate determines what percent of the messages were received.

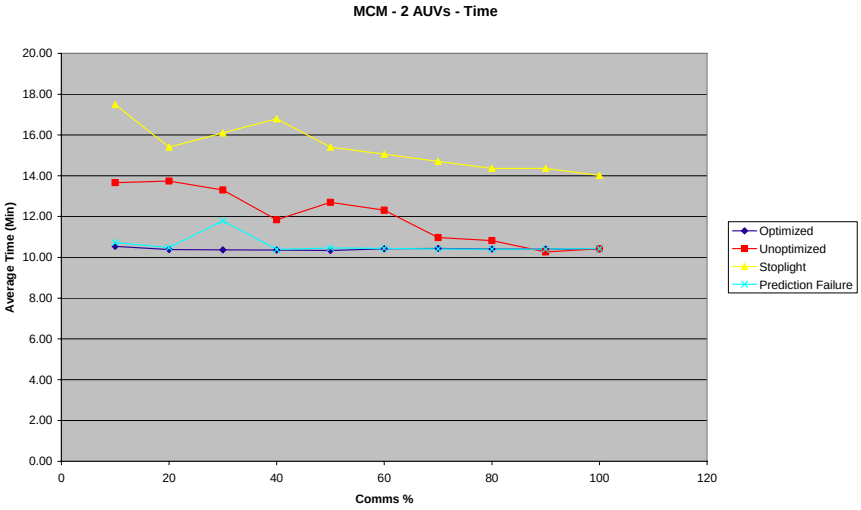


Figure 7.10 MCM time system comparison for 2 AUVs.

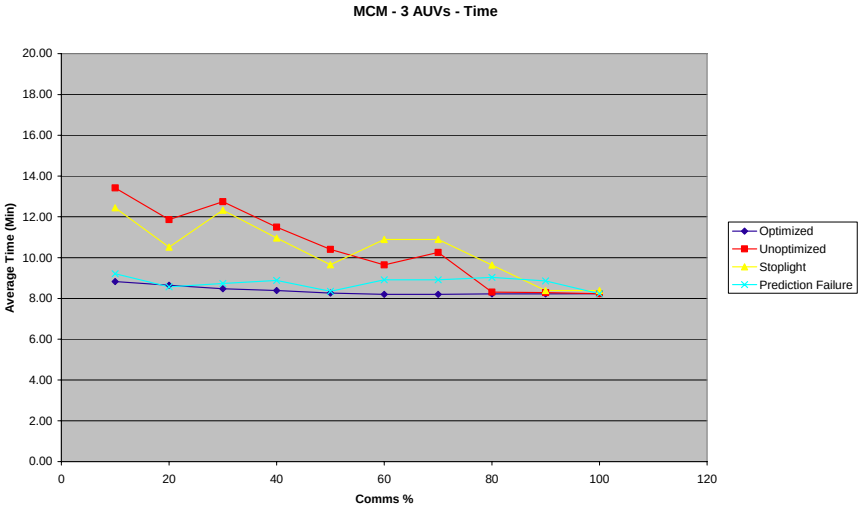


Figure 7.11 MCM time system comparison for 3 AUVs.

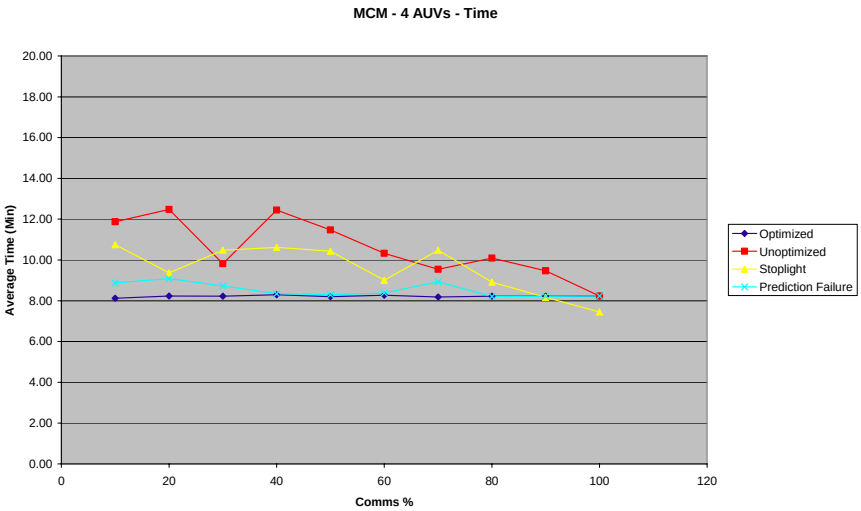


Figure 7.12 MCM time system comparison for 4 AUVs.

Figure 7.10 shows the 2 vehicle data where the optimised and prediction failure DELPHÍS systems returned relatively steady times. The un-optimised system's time increased as communications dropped, as did the stoplight system which returned significantly higher mission times than the other three systems (this will be explained in section 8.3.4). In Figure 7.11 the 3 vehicle data shows similar trends with the exception of the initial stoplight system time being equivalent to the other systems'. This is again repeated in the 4 vehicle data (Figure 7.12).

7.2.5 Efficiency

In order to calculate the total efficiency of a multi-AUV mission the previously presented data was combined using the formula described in section 6.5.2. The results for the multi-AUV system tests are shown in Figure 7.13, Figure 7.14 and Figure 7.15.

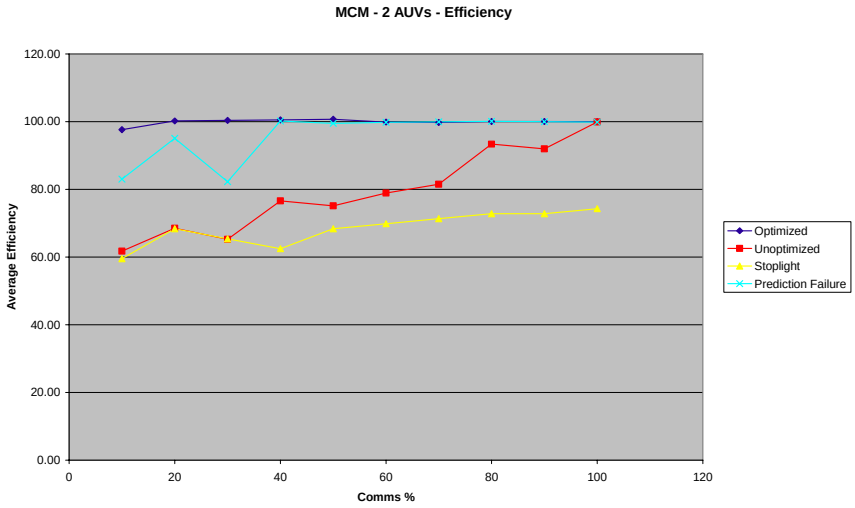


Figure 7.13 MCM efficiency system comparison for 2 AUVs.

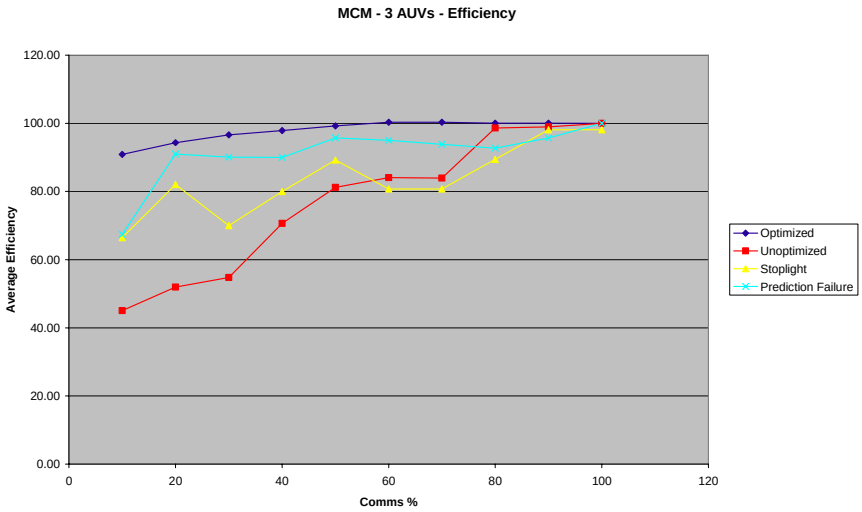


Figure 7.14 MCM efficiency system comparison for 3 AUVs.

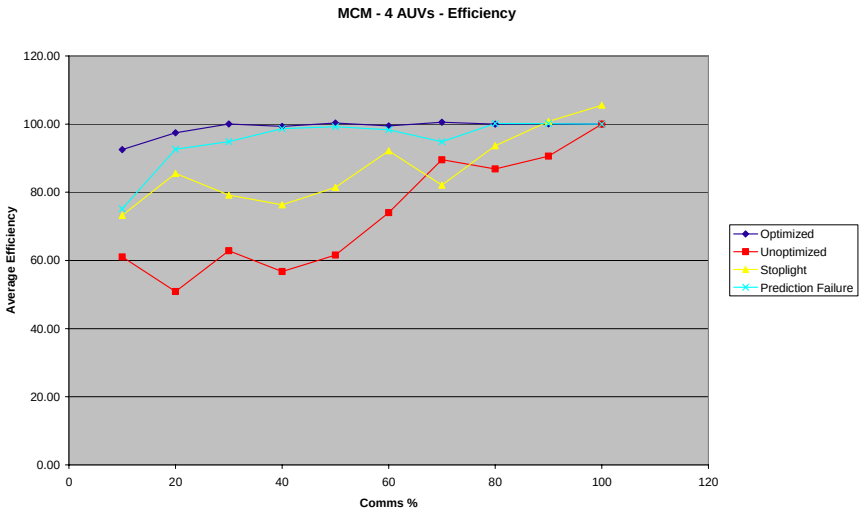


Figure 7.15 MCM efficiency system comparison for 4 AUVs.

Mission efficiency in the 2 vehicle scenario (Figure 7.13) stayed above 90% for the optimised DELPHÍS system. The prediction failure system returned similar results with only minor fluctuation. The un-optimised system's efficiency decreased steadily as communication dropped. The stoplight system had a much lower initial efficiency and again decreased with communications (this will be explained in section 8.3.5). This situation is echoed in the 3 vehicle data (Figure 7.14) except that the stoplight system's initial efficiency improved before decreasing as before. The 4 vehicle data presented in Figure 7.15 again showed similar results.

7.3 Pipeline Tracking

The data for the pipeline tracking mission vignette is presented in the same way as the MCM mission vignette starting with redundancy followed with missed goals, target detection, mission time and finally total mission efficiency.

7.3.1 Redundancy

Like in the MCM vignette section, graphs will be shown presenting the average number of redundant goals. Data for 2, 3 and 4 AUV systems is shown in Figure 7.16, Figure 7.17 and Figure 7.18 respectively.

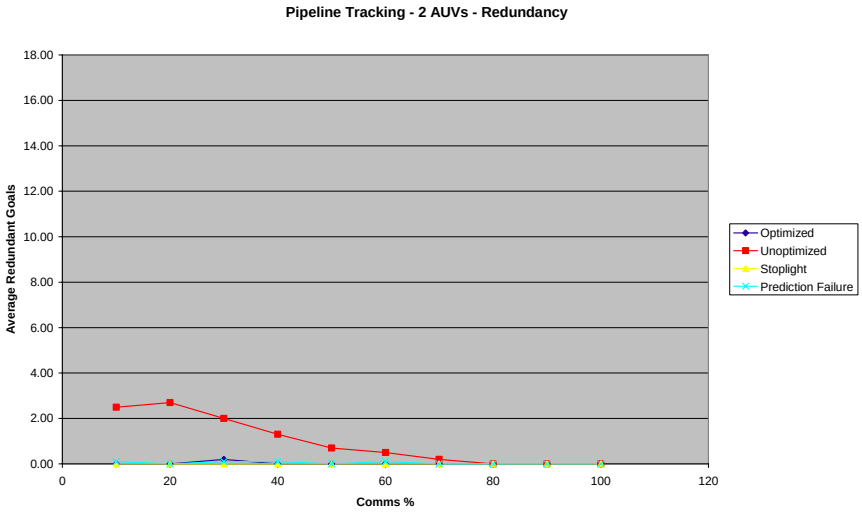


Figure 7.16 Pipeline tracking redundancy system comparison for 2 AUVs.

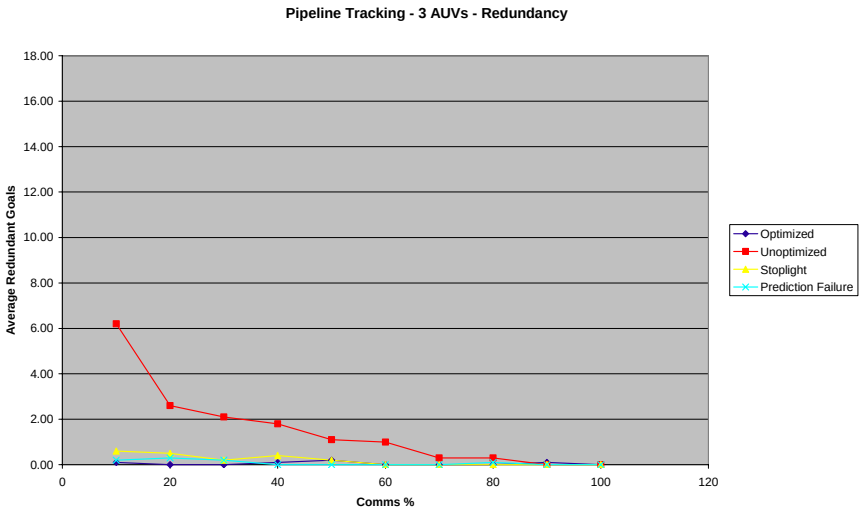


Figure 7.17 Pipeline tracking redundancy system comparison for 3 AUVs.

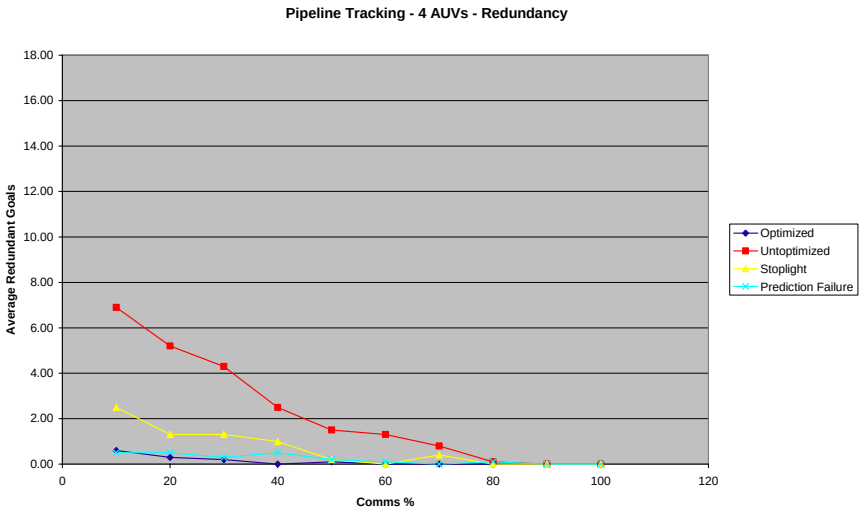


Figure 7.18 Pipeline tracking redundancy system comparison for 4 AUVs.

In the 2 vehicle data (Figure 7.16) average goal redundancy was 0% for the optimised DELPHÍS system as well as the prediction failure and stoplight systems. The un-optimised system's average redundancy gradually increased to just over 2 goals as communications deteriorated. The 3 vehicle data (Figure 7.17) was similar with the un-optimised system recording more and more redundant goals as communications dropped while the other three systems maintained much lower values. Figure 7.18 shows that the 4 vehicle data was similar with relatively minor increases by the optimised, prediction failure and stoplight systems.

7.3.2 Missed Goals

Here the data illustrating the average number of missed goals will be presented (Figure 7.19, Figure 7.20 and Figure 7.21).

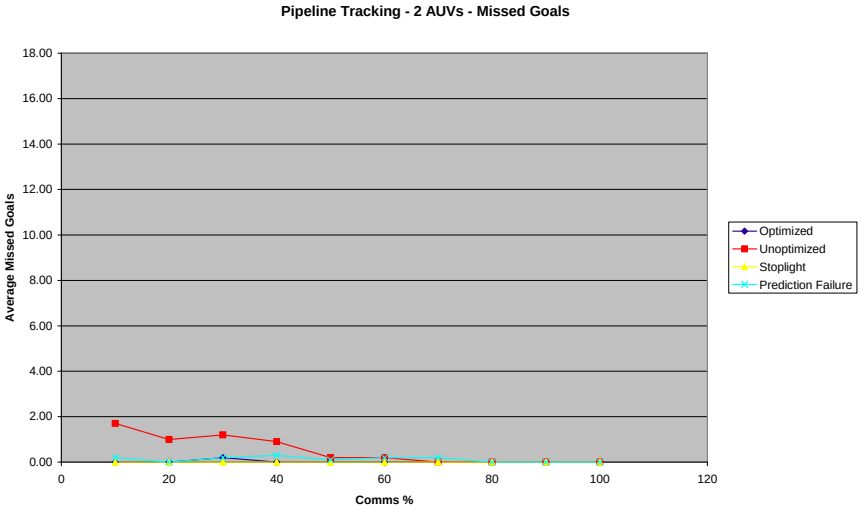


Figure 7.19 Pipeline tracking missed goals system comparison for 2 AUVs.

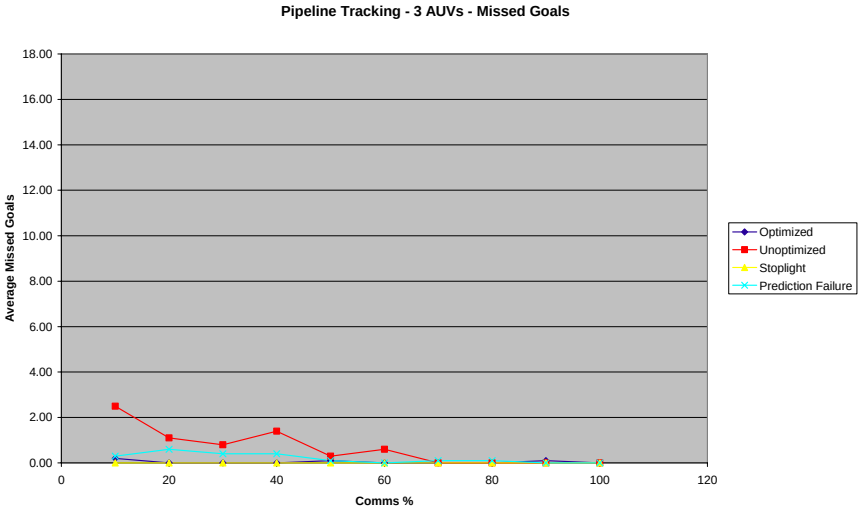


Figure 7.20 Pipeline tracking missed goals system comparison for 3 AUVs.

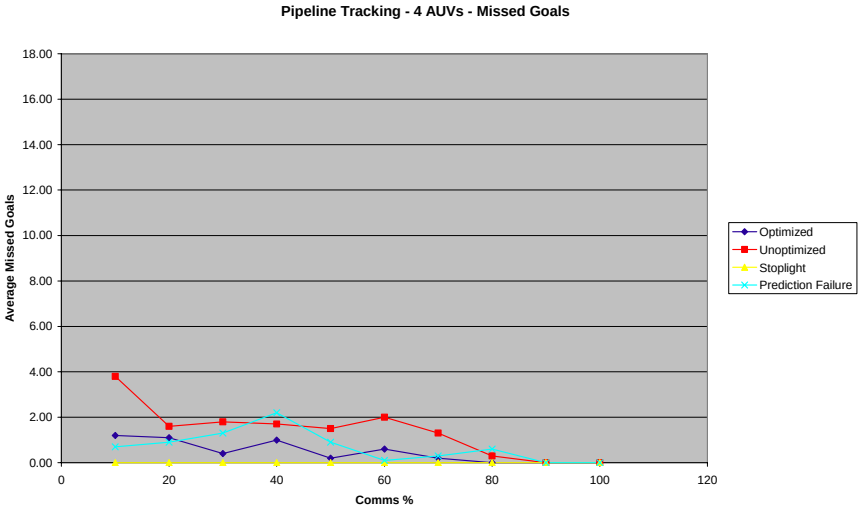


Figure 7.21 Pipeline tracking missed goals system comparison for 4 AUVs.

Figure 7.19 shows the 2 vehicle data where with the exception of the un-optimised DELPHÍS system missed goals were kept to a minimum. In the 3 vehicle scenario shown in Figure 7.20 similar data is returned with the un-optimised system gradually increasing the average number of missed goals as communication dropped. The prediction failure DELPHÍS system had a marginal increase as compared to the 2 vehicle data but still remained relatively low. The 4 vehicle data (Figure 7.21) again shows the upward trend of the un-optimised system with another increase by the prediction failure system. The optimised DELPHÍS system also recorded more missed goals than in the other two group sizes but still remained just over 1 on average in the worst communication rate.

7.3.3 Target Detection

Unlike the MCM vignette the number of detected targets didn't vary much throughout the experiments. The only scenario where target detection wasn't 100% was the DELPHÍS with prediction failure and in this case missed targets were very minimal (1 target was missed in 6 of 300 trials). The reasoning for this will be explained in Chapter 8.

7.3.4 Time

Here the time data for the pipeline tracking mission will be shown. As mentioned earlier a single AUV system works the same in any communication environment and therefore the data in this section focuses on multiple vehicle data. Table 7.2 shows the average mission time in 100% communication for different numbers of AUVs.

Table 7.2 Average pipeline tracking mission time in 100% comms.

# of AUVs	Mission Time
1	21:14
2	12:59
3	8:17
4	7:36

The multi AUV time data is illustrated in Figure 7.22, Figure 7.23 and Figure 7.24.

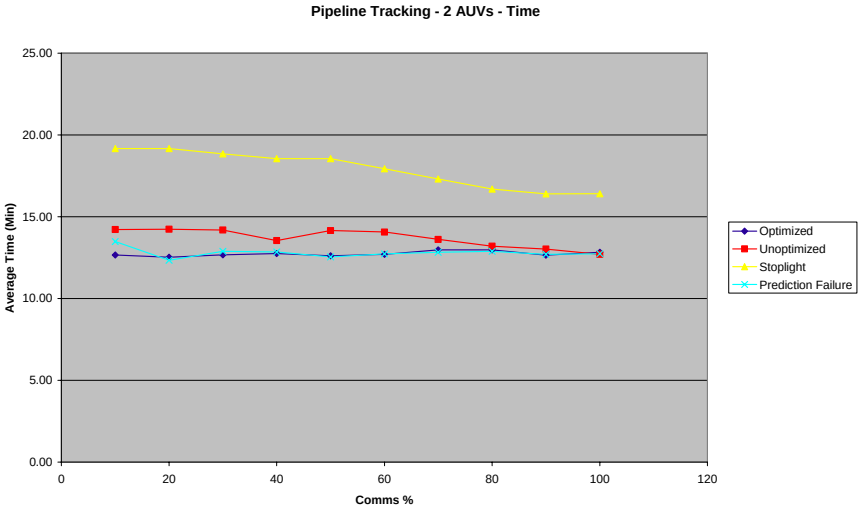


Figure 7.22 Pipeline tracking time system comparison for 2 AUVs.

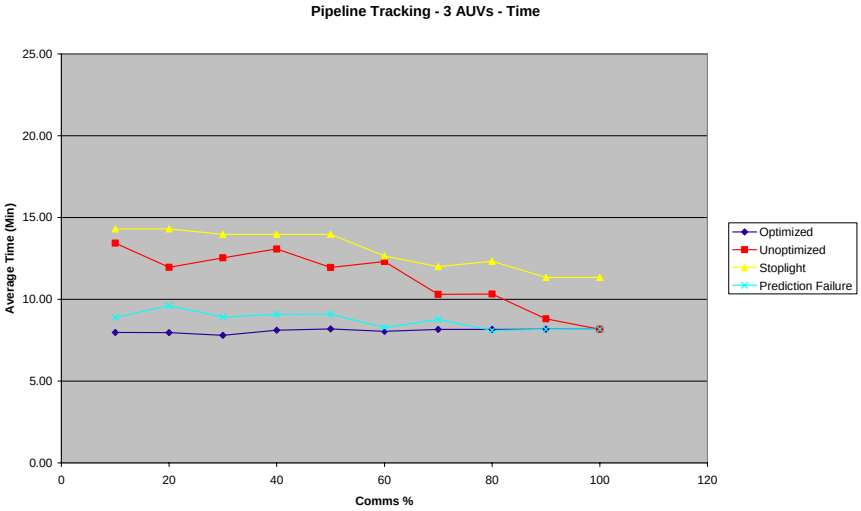


Figure 7.23 Pipeline tracking time system comparison for 3 AUVs.

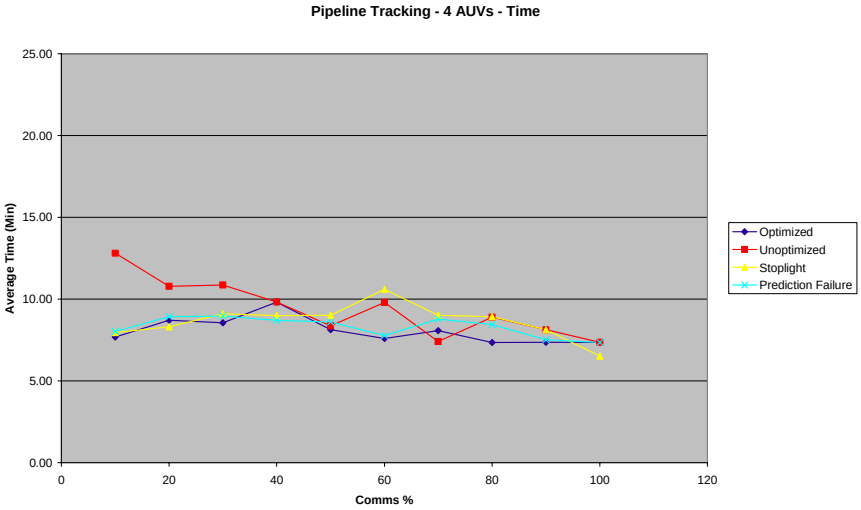


Figure 7.24 Pipeline tracking time system comparison for 4 AUVs.

In the 2 vehicle data shown in Figure 7.22 the optimised and prediction failure DELPHÍS systems returned relatively steady mission times. The un-optimised system increased a little bit as communications dropped with the stoplight system showing a similar trend (in addition to an initially high mission time which will be explained in section 8.3.4). The 3 vehicle data (Figure 7.23) showed similar trends but with shorter mission times. In the 4 vehicle scenario the data was more erratic. The un-optimised DELPHÍS system repeated its increasing trend while the optimised and prediction failure systems recorded relatively "bumpy" data. The stoplight system data was equally erratic and didn't show the same trends seen in the previous 2 group sizes. This will be explained in section 8.3.4.

7.3.5 Efficiency

Like in the MCM vignette the efficiency of a single AUV completing a pipeline tracking mission is always 100%. Hence, the data presented in Figure 7.25, Figure 7.26 and Figure 7.27 will focus on 2, 3 and 4 vehicle results.

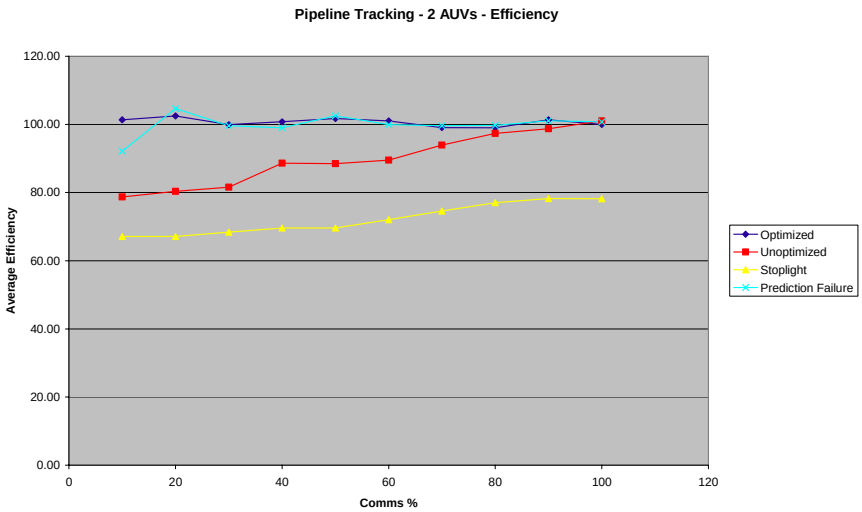


Figure 7.25 Pipeline tracking efficiency system comparison for 2 AUVs.

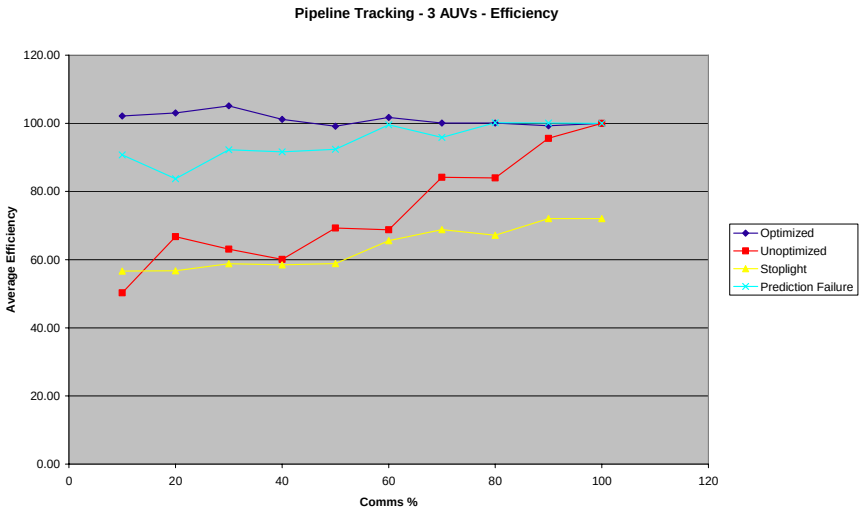


Figure 7.26 Pipeline tracking efficiency system comparison for 3 AUVs.

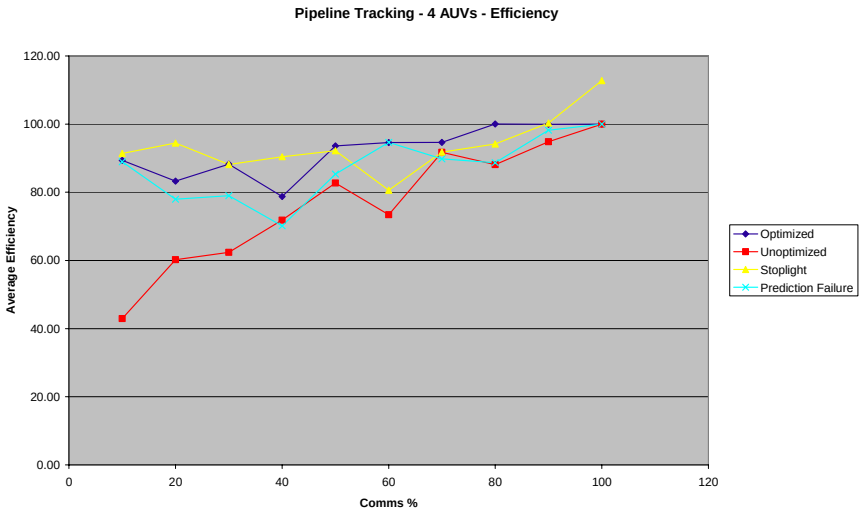


Figure 7.27 Pipeline tracking efficiency system comparison for 4 AUVs.

In Figure 7.25 the 2 vehicle efficiency data is presented showing relatively steadily high efficiency rates for both the optimised and prediction failure DELPHÍS systems. The un-optimised system shows a steady decline in efficiency as communication rates dropped. This is also true for the stoplight system which again had initially low values which will be explained in section 8.3.5. The 3 vehicle data (Figure 7.26) returned high efficiencies from the optimised system and only slightly lower values from the prediction failure system. The un-optimised and stoplight systems showed the same trends as they did in the 2 vehicle data with slightly lower values. In the 4 vehicle efficiency data (Figure 7.27) the data was more erratic. As shown in the time graphs, the un-optimised system showed a distinct decline in efficiency as communication rates worsened. The other three systems returned less clear data. This will be explained in detail in the next chapter.

7.4 Real World Validation

As mentioned in Chapter 6, in addition to the simulated trials the DELPHÍS system was also demonstrated in the real world by coordinating a mine countermeasures mission with the AUVs REMUS and Nessie III. Trials were conducted to both validate the simulated results as well as demonstrate the ability of the DELPHÍS system in the presence of real world environmental factors, mainly communication unreliability. Initial trials to demonstrate coordination between a real and simulated AUV were run at Threipmuir Reservoir on the 18th of September, 2008. These were followed by multiple AUV trials held at Loch Earn from September 30th to October 2nd, 2008.

Figure 7.28 REMUS and Nessie III at Loch Earn (October 2, 2008)

7.4.1 Preliminary Work

Before true multi-AUV trials could be run the DELPHÍS system had to be validated on both the REMUS and Nessie III AUV. For REMUS this required the development of an interface layer that would essentially translate the Ocean Systems Laboratory OceanSHELL messages to commands that the commercial REMUS vehicle could understand. Called the Application Layer Interface (ALI) this software module developed in the Ocean Systems Laboratory acts as a buffer between OSL software and proprietary robotics architectures allowing for general architectures (such as the DELPHÍS system) to control many different types of vehicle.

Using the ALI the DELPHÍS system was tested on the REMUS AUV at Loch Earn on April 23rd, 2008. In these tests REMUS was given a simple mission that was overridden

by the DELPHÍS system at a certain point so that it could take control of the vehicle and execute a lawnmower search of an area. These tests were successful and having proved the ability of the DELPHÍS system to control REMUS served as the first step towards the goal of multi-AUV coordination.

Unlike the work done for REMUS, interfacing the DELPHÍS system to the Nessie III AUV was far simpler. Because the vehicle was designed and built in the Ocean Systems Laboratory there was no need for the ALI and therefore the DELPHÍS system was able to control Nessie using OceanSHELL messages. Trials were conducted at Threipmuir Reservoir on September 9th, 2008 and successfully proved the DELPHÍS system's ability to conduct a lawnmower search with simulated mines using the Nessie III vehicle

In addition to the AUV validation work the acoustic message sending ability for both vehicles had to be created. For REMUS this functionality was built into the ALI and utilised the acoustic modem already built into the vehicle. For Nessie however this required the installation of a WHOI MicroModem and the development of a software driver to control it.

The addition of the modems also required a minor change to the acoustic broadcasts mentioned in section 5.5.2. As mentioned earlier, due to the nature of acoustic communication messages have to be kept small. In the case of the WHOI MicroModem the default size of the messages is limited to 32 bytes (This size can in theory be increased however not all resources owned by the Ocean Systems Laboratory have this functionality and therefore the default size was used). For the simulated experiments this limitation was not enforced for simplicity however it was mandatory in the real world trials. In order to compact the necessary information into such a small size a number of tactics were used. First, the current position of the vehicle was referred to not in global frame (latitude, longitude) but in local frame (north, east). Vehicles could then give their location in relation to a common origin in meters removing the need to store position in floats. This local frame modification was also applied to the discovered target information. Another way the messages were compacted was by limiting the number of completed goals and discovered targets that were transmitted. Upon completion of this software and the testing of the modems the coordination trials

could begin. The message definition displayed in Figure 5.6 was modified and the 32 byte version can be found in Figure 7.29.

```
<data>
   <message name="AcousticCommsMsg">

      <field type="Char8" name="AUV_ID"/>
      <field type="Char8" name="AUV_Type"/>
      <field type="Char8" name="Current_Goal"/>
      <field type="Int16" name="X_Coor"/>
      <field type="Int16" name="Y_Coor"/>
      <field type="Char8" name="Z_Coor"/>
      <field type="Char8Array" name="Completed_Goals" length="10"/>
      <field type="Int16Array" name="Mines" length="6"/>

   </message>
</data>
```

Figure 7.29 XML definition of the 32 byte acoustic broadcast message.

7.4.2 AUV / Simulated AUV Trials

Once the DELPHÍS system had been validated on the vehicles and the acoustic communication functionality tested the next step was to begin the coordination trials. Before putting two vehicles in the water together however, tests were conducted between a real AUV and a simulated one. The goal was to have a real AUV in the water coordinating acoustically with a simulated vehicle operating on a computer on shore. This would prove the system able to handle the difficulties of in water acoustic communication without the complexity of two mobile robots. Trials were carried out at Threipmuir Reservoir on September 18th, 2008 and aimed to show coordination between Nessie III and a simulated vehicle. The mission demonstrated was a small MCM consisting of 4 30 metre legs and 2 simulated mines. A diagram of Threipmuir Reservoir and the mission can be seen in Figure 7.30.

Figure 7.30 AUV/Simulated AUV mission in Threipmuir Reservoir[1].

Three separate experiments were run to test the coordination ability of the DELPHÍS system. In the first experiment the lawnmower leg super-goals (made up of two completion and execution locked waypoints) were encompassed in a "Search" super-goal and restricted by both execution and completion locks. This resulted in only one vehicle being allowed to execute the search. In this test Nessie started the mission first and completed the lawnmower while a simulated vehicle investigated discovered targets as they were added to the plan.

The second experiment removed the "Search" super-goal so that both Nessie and the simulated AUV could attempt the lawnmower concurrently. This resulted in the search being split up between the two vehicles dynamically. To simplify this test the mine detection ability was deactivated on both vehicles. The third experiment however reactivated the mine detection ability resulting in vehicles splitting up the search task *and* the investigate tasks.

Data from these experiments showing the behaviour of both Nessie and the simulated vehicle can be found in Figure 7.31, Figure 7.32 and Figure 7.33. Nessie's path is represented in blue with the simulated vehicle path shown in dotted white.

[1] Image courtesy of Google Earth.

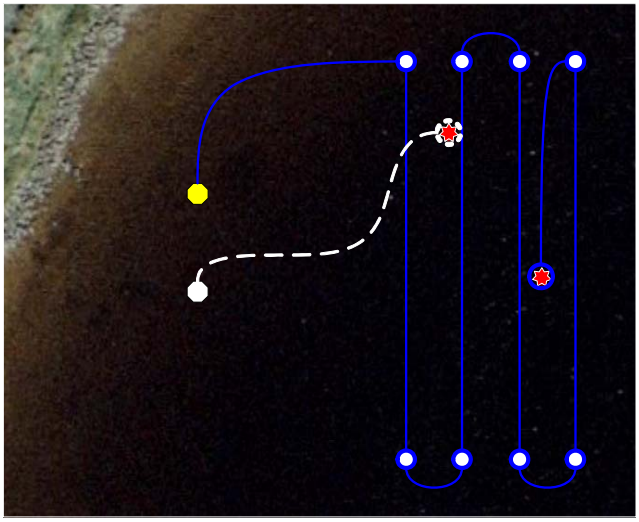


Figure 7.31 Trial 1 of AUV/Simulated AUV tests in Threipmuir Reservoir.

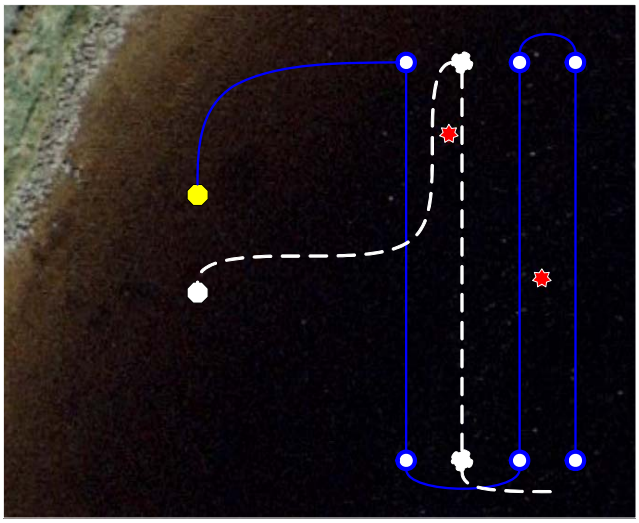


Figure 7.32 Trial 2 of AUV/Simulated AUV tests in Threipmuir Reservoir.

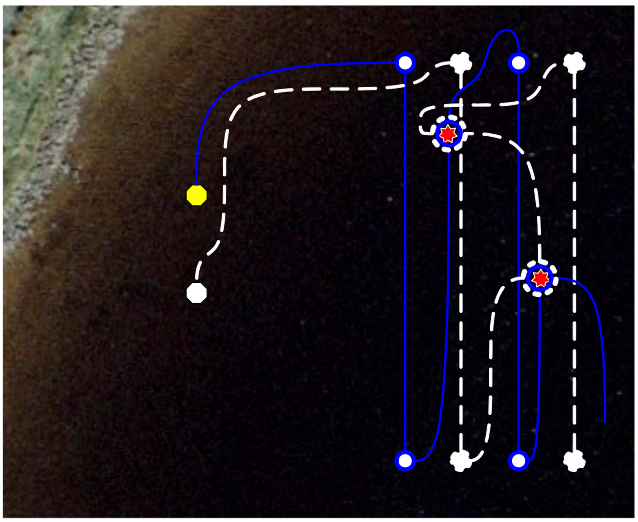


Figure 7.33 Trial 3 of AUV/Simulated AUV tests in Threipmuir Reservoir.

In trial 1 Nessie successfully completed the lawnmower search while the simulated vehicle investigated the first discovered target. After the search was complete, Nessie

continued by investigating the second target. Trial 2 showed the lawnmower search dynamically broken into legs with no missed or redundant goals. There were a few coordination errors in trial 3 which resulted in some mission redundancy. These errors were due to some bugs in code which will be explained in Chapter 8 and which were fixed before the multi-vehicle trials in Loch Earn. Full mission logs can be found in Appendix C which will also be analysed in more detail in Chapter 8.

7.4.3 Multi-AUV Trials

Having demonstrated the ability of the DELPHÍS system to coordinate a real AUV with a simulated one the final test was to replace the simulated vehicle with a real platform. Held from September 30th to October 2nd trials were conducted at Loch Earn to demonstrate coordination between the REMUS and Nessie III AUVs. Like the Threipmuir tests the mission was a lawnmower consisting of 4 legs (200 metre) and 2 simulated mines. A diagram of the mission and Loch Earn can be found in Figure 7.34.

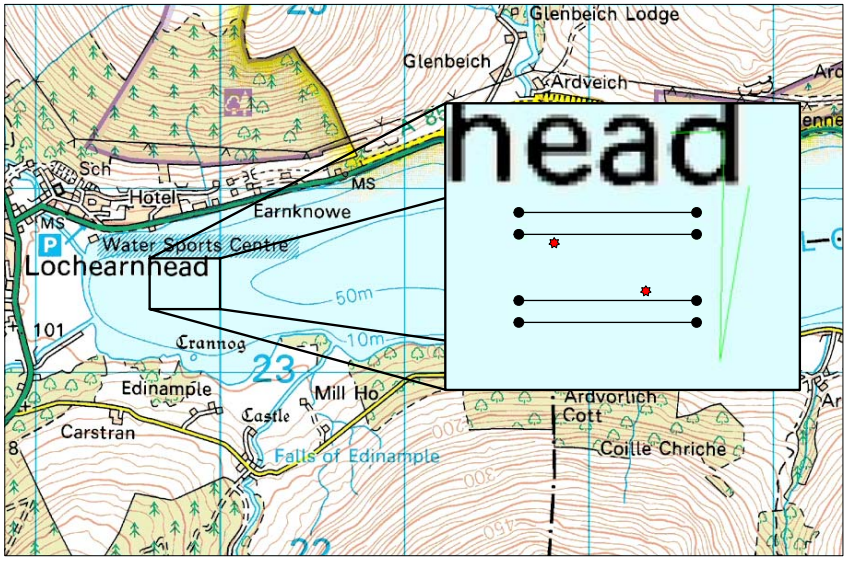


Figure 7.34 Multi-AUV mission in Loch Earn.

Due to the speed difference between REMUS and Nessie (2.0 and 0.4 metres per second respectively) the lawnmower legs were again encompassed in an execution and completion locked "Search" super-goal so that only one vehicle (in these trials REMUS) could attempt it. A number of trials were run with REMUS conducting the search and Nessie investigating discovered targets. The data from these experiments can be found in Figure 7.35 and Figure 7.36.

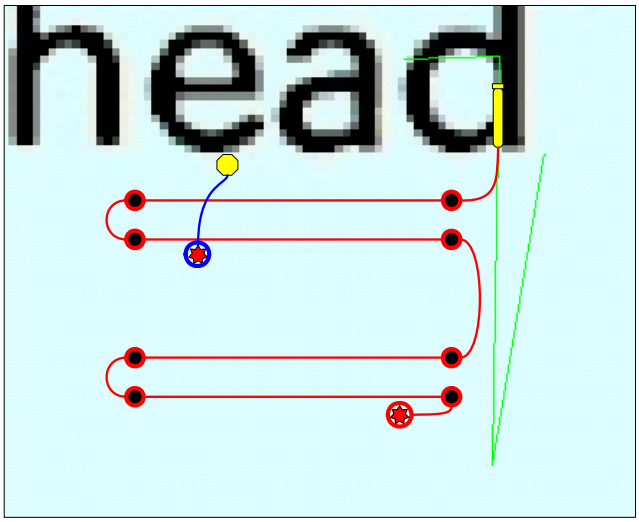


Figure 7.35 Trial 1 of multi-AUV tests in Loch Earn.

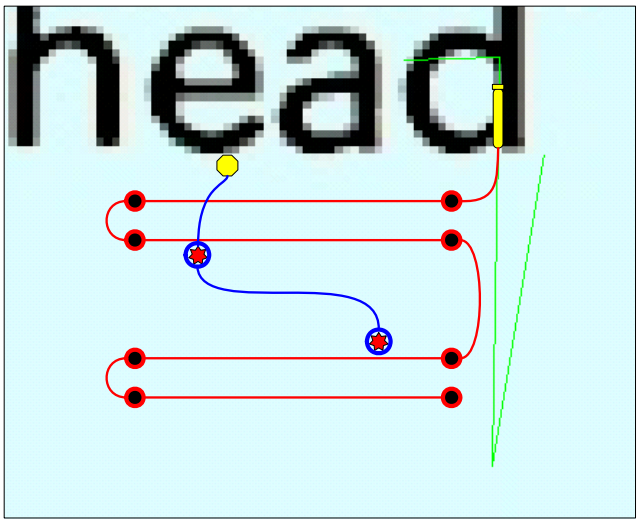


Figure 7.36 Trials 2-3 of multi-AUV tests in Loch Earn.

In the first trial the mines were spread out in such a way that Nessie was too slow to accomplish both resulting in REMUS executing one of the target investigations. For the second two trials the second mine was moved north so that Nessie would have a better chance of selecting it before REMUS. The result was the same in both, with Nessie investigating both targets while REMUS waited in holding pattern following the search. Logs for these missions can be found in Appendix D and will be analysed further in Chapter 8.

Having proved the DELPHÍS system able to coordinate two real AUVs the same mission was performed in simulation using the same communication rates observed during the in water trials. Results showed that the mission was executed in exactly the same manner and thereby validated the results obtained in simulation. This will be discussed in more detail in the next chapter.

7.5 Conclusion

Experimental data has been presented showing that the tools encompassed by the DELPHÍS system increase efficiency as compared to the state of the art. Graphs for each of the individual efficiency metrics were displayed for each of the four experimental systems for both MCM and pipeline tracking and proved the ability of this research to maximise each. In addition the system was demonstrated in coordination trials with the REMUS and Nessie III AUVs. In water missions were re-executed in simulation and data was the same, thereby validating the simulation results. The next chapter will discuss these results in more detail and explain the significance of the data.

Chapter 8

Discussion

8.1 Introduction

This chapter will explain the data presented in Chapter 7. First the behaviour of the vehicles will be shown to help visualise how the missions were executed in addition to showing how the DELPHÍS system was able to optimise mission execution. Then the data will be examined in detail to explain the trends and make initial conclusions about the multi-AUV coordination systems. Following this analysis the real world validation data will be discussed concluding with a description of the results in terms of AUV group size.

8.2 Behaviour

Before explaining the results in terms of the data presented in Chapter 7 the results will first be explained in regard to the behaviours witnessed during experiments. First the general behaviour of the vehicles during mission execution will be presented followed by some of the mission optimisation techniques used when communications were intermittent.

8.2.1 Mission Execution

In the previous chapter data was presented showing how the multi-AUV coordination systems coped with degrading communication in terms of a number of efficiency metrics. In order to understand these results it is important to visualize what an *expected* mission looks like for each system. In this study an expected run is the average mission completed in 100% communications. Diagrams showing the expected data for the mine countermeasures and pipeline tracking mission vignettes can be seen in Figure 8.1 and Figure 8.2 respectively.

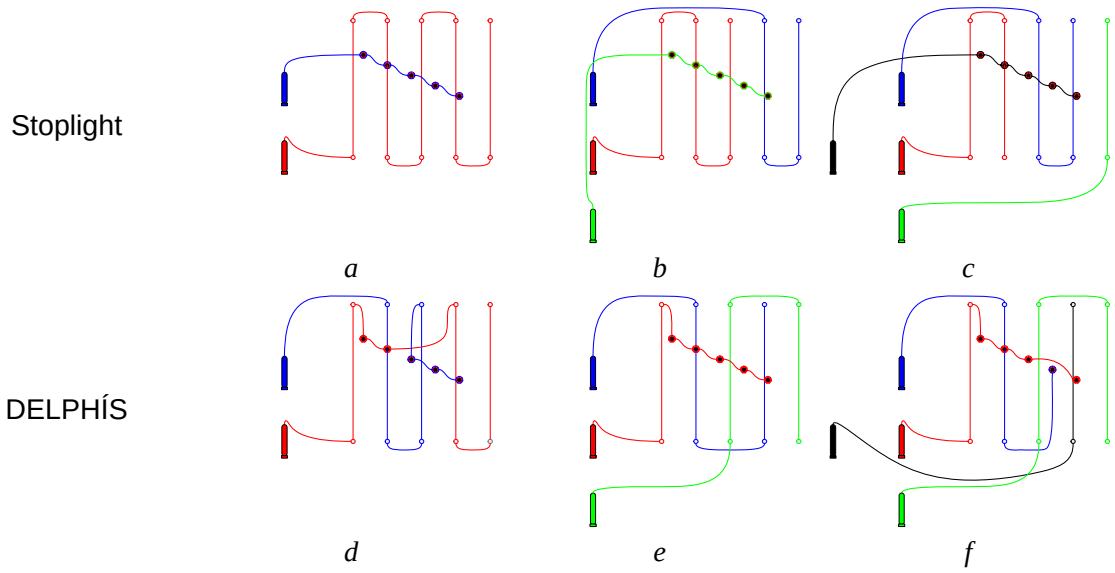


Figure 8.1 Expected mission behaviour for the MCM vignette.

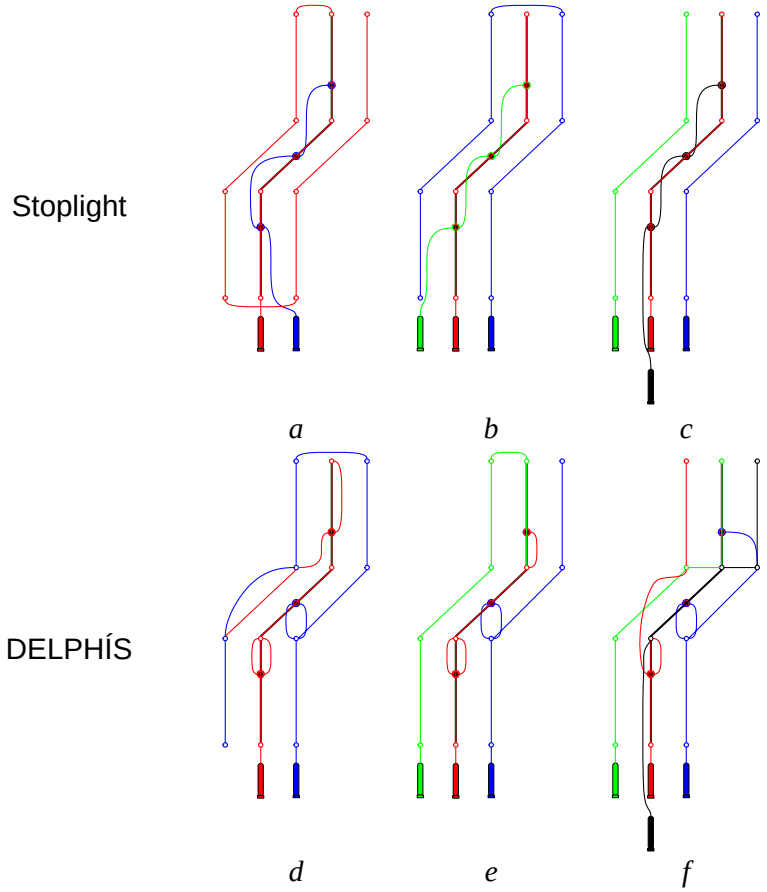


Figure 8.2 Expected mission behaviour for the pipeline tracking vignette.

The DELPHÍS data encompasses all three versions tested in this study (optimised, un-optimised and prediction failure) because at 100% communications they all function the same. This will be explained in further detail in section 8.3.

As described in section 4.6.2 stoplight multi-AUV coordination systems operate by giving vehicles scripts to run a priori, included in which are points where vehicles must synchronise with each other. In the mine countermeasures mission shown in Figure 8.1 the stoplight system was designed so that there would be vehicles performing the search and vehicles investigating the discovered mines. In the 2 vehicle scenario this designation was straight forward. As more vehicles were added, the search was broken up accordingly to attempt to maximise efficiency a priori. The DELPHÍS system on the other hand was given the mission and allowed to break up the task on its own during run time. This resulted in different yet equally successful approaches to the same mission.

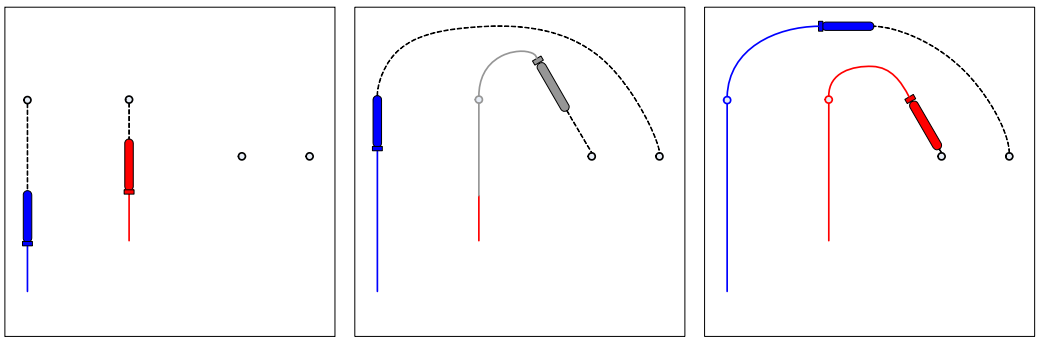
In the pipeline tracking mission vignette the stoplight system was programmed in the same manner as the MCM vignette where there were vehicles assigned to tracking the pipe and vehicles investigating targets. As more vehicles were introduced, the tracking legs were divided. The DELPHÍS system however took advantage of its ability to choose tasks on the fly and because of the high priority of the targets was able to investigate them as they were found, resulting in somewhat less elegant mission paths but more efficient mission execution. This will be proven in section 8.3.

To reiterate, Figure 8.1 and Figure 8.2 show how missions were performed in *100% communications*; this does not take into account what happens when communication rates drop and unforeseen events are introduced. The next section will explain some of the optimisation behaviours that the DELPHÍS system used to handle these unforeseen events and keep mission execution as efficient as possible.

8.2.2 Mission Optimisation

As described in sections 5.5.5 and 5.5.6 the DELPHÍS system was built with mission optimisation modules to help maintain efficient mission execution when communication falters. During the experiments these modules allowed vehicles to both make intelligent decisions about other vehicles as well as reconcile conflicts when they occurred.

One of these mission optimisation techniques is the ability to predict what other vehicles are doing and will do next. This is an important issue as communication underwater is intermittent at best. Figure 8.3 shows an example of how this occurred in the experiments.

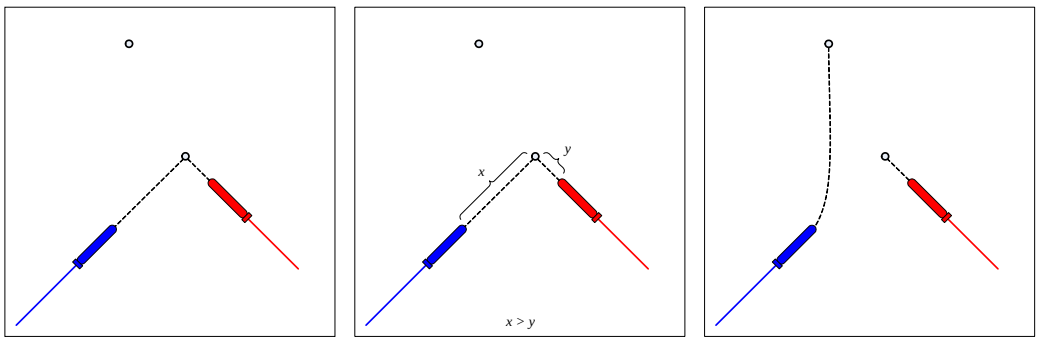


a *b* *c*

Figure 8.3 AUV prediction behaviour.

In *a* there are two vehicles working on two different goals. Between *a* and *b* however communication fails. At this point the blue vehicle has accomplished its goal but doesn't know whether or not the red vehicle has. Using the methods described in section 5.5.5 it is able to predict that the red vehicle has finished its goal and selected the next closest. Using this information the blue vehicle then chooses the next closest goal to it, which it wouldn't have done had it not taken into account the actions of the red vehicle (because the red vehicle goal is technically closer). In *c* communication has returned and the actual mission status shows the prediction made to be accurate. In the case that the prediction was *not* accurate, there are other mission optimisation techniques used by the DELPHÍS system to get things back on track.

Often the result of an incorrect prediction is that two vehicles end up concurrently attempting the same goal. This happens when two vehicles predict the opposite of each other. Say for instance vehicle *x* predicts that vehicle *y* was working on goal *a* so it would attempt goal *b*. If at the same time vehicle *y* made the exact same prediction, but inverting the vehicles a conflict would arise. In this scenario the DELPHÍS system has functionality to decide which vehicle should continue and which should break off and select another task. Figure 8.4 shows a situation where after communication returning two vehicles are found to be working on the same goal.

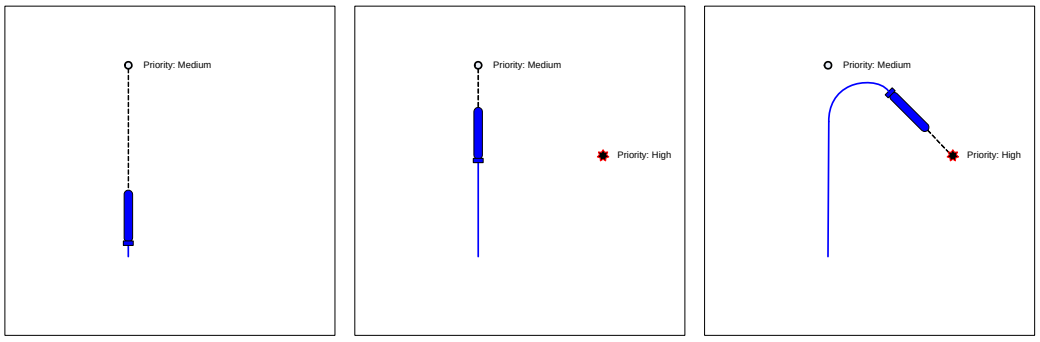


a *b* *c*

Figure 8.4 Two AUVs attempting the same goal resolution behaviour.

In this situation both vehicles are aware of the situation because their respective mission models have been alerted to the conflict. They also both know the position of the other since at this point communication has returned, even if this is only in the form of one complete message. By calculating both vehicles' distance to the target, the vehicles can decide which is in a better position and whether or not to continue. In this case the blue vehicle is farther away so it breaks off and selects another goal.

Another example of vehicles stopping execution of a goal for another is when a higher priority goal becomes available, determined by the priority flag contained in goal nodes. Figure 8.5 shows a situation where a mine is detected while the vehicle is accomplishing another goal.



a *b* *c*

Figure 8.5 Higher priority goal appearance behaviour.

Here, the initial goal has a medium priority, whereas the mine has a high priority. The DELPHÍS system becomes aware that there is a more suitable goal available and is able to break execution in favour of it. This behaviour was witnessed in these experiments when a target was discovered while they were traversing between mission legs (vehicles couldn't break out from mission legs because they were restricted by execution and completion locks to maintain smooth runs). This allowed the vehicles to handle the

most important tasks as they became available. The next section will explain the experimental data presented in the previous chapter.

8.3 Coordination Data

This section will go through the data in terms of the different efficiency variables described in section 6.5.2. First the mission accuracy data will be described in terms of redundancy and then missed goals. This will be followed by the target acquisition data, mission time data and finally the calculated efficiency data.

The data described in this section will be separated in terms of mission vignette (MCM, pipeline tracking) and system type (DELPHÍS, stoplight, etc.). Before starting the data analysis it is important to point out that the three versions of the DELPHÍS system (optimised, un-optimised, prediction failure) differ in the way they optimise mission execution in the face of intermittent communications. In 100% communication environments however there are no communication related conflicts and therefore there is no application of the mission optimisation techniques. This results in all three systems performing identically in the initial full communication state. This can be seen in the data presented in Chapter 7. Because this research is evaluating the efficiency of the optimised DELPHÍS system the expected data (defined in section 8.2.1) of this system was used as a comparison baseline.

8.3.1 Redundancy

The first measure of efficiency that will be presented here is the redundant goals data. It is logical that as communications between vehicles declined the average number of redundant goals would most likely increase in communication dependent coordination systems. The data from the mine countermeasures mission vignette confirms this. The most communication dependent of the four systems is the un-optimised DELPHÍS system and the results clearly show that as communication rates dropped the average number of redundant goals grew. This is due to the fact that vehicles could choose a goal while communications were down unknowing that another vehicle had just done the same. Because this specific version of the system has the full optimisation capabilities turned off it is unable to handle the conflict resulting in goals being accomplished more than once.

The optimised DELPHÍS system was able to keep the number of redundant goals to a minimum in all three group sizes. This number does increase with group size which is to be expected as the more AUVs added to the system the more concurrent decisions are being made and consequently the more complex the coordination task. In addition as the time between communications grows so does the chance that a goal could be selected and completed in the blackout period, resulting in redundancy without the knowledge of the system. The DELPHÍS system with prediction failure managed to not only achieve similar results but actually surpassed the regular optimised DELPHÍS system and kept its average goal redundancy lower. This could be due to a few factors. One possibility is that there might not have been enough trials to accurately measure the systems performance (see section 8.3.6). Another possibility is that because the prediction module was predicting events to be happening twice as fast as they actually were it might be that this *looking ahead* was beneficial in terms of goal redundancy.

The stoplight system redundant goal data was affected by group size as much if not more than by communication rate. Because of the nature of these systems where mission segments are pre-designated to vehicles a priori, the only chance of goal redundancy is in the mines that are discovered at run time. In the 2 vehicle scenario there wasn't a single redundant goal recorded. This is because with only two vehicles the one designated to the search has so much more to do than the one designated to target inspection that there is almost no chance that both will be vying for targets at the same time. In the 3 vehicle scenario redundant goals didn't begin to show up until the communication rate was very low and it was possible that it could be out for long enough for more than one vehicle to attempt the same mine.

In the 4 vehicle scenario however goal redundancy was on average nearly one goal per run even in 100% communications. Though this data may seem strange, it's possible because of a random timing issue. When 4 AUVs were executing a stoplight version of this particular MCM mission, two vehicles can happen to finish their respective mission legs within 1-2 seconds of each other. Because the acoustic broadcast beacon sends information every *3* seconds it was possible for both to be in 100% communications but decide upon the same mine. Because of this, the data for redundancy was artificially high at the start however as communication rates dropped the mission execution order changed removing this synchronisation issue. An important note here is that this kind of situation is a simulation specific one. Though a synchronisation problem like this

could happen in real environments, it is far more unlikely for it to happen repeatedly trial after trial they way it can in simulation where environmental variables are static.

The data from the pipeline tracking mission vignette follows the exact same trends. In this case however there isn't the aforementioned random synchronisation issue so the data for 4 vehicles is more logical. In addition to the affect of communication loss AUV group size had an affect on goal redundancy also. Though the trends stayed mostly the same the magnitude of redundancy increased as more AUVs were coordinated. This is a logical result as more vehicles mean more concurrent decisions and larger chances that conflicting decisions will be made while out of communication.

8.3.2 Missed goals

The missed goal data was similar to the redundant goal data (though in this study somewhat more important since it was weighted more in the efficiency equation). Like goal redundancy, the number of missed goals was expected to rise in the presence of low communications in systems that were communication dependent since in these situations there was a better chance of incorrect assumptions being made about other vehicles.

Again this was seen to be true in the MCM mission vignette. In the un-optimised DELPHÍS system the effect of dropped communication on the number of missed goals was apparent even when the drop rate was only 10% in the 2 AUV mission. As more vehicles were added to the system the missed goals were still apparent but more and more limited to the missions with lower communications. This seems to suggest that in this mission vignette the more vehicles in the system the less of a chance there are going to be missed goals. This is logical because as was shown in section 8.3.1, the more vehicles there are in the system the more redundant goals and it would make sense that this would essentially raise the probability that any given goal will be accomplished.

The same trend was seen in the DELPHÍS system with prediction failure however because of its mission optimisation modules (sans correct prediction) it was able to keep the number of missed goals a bit lower. In the 4 vehicle mission missed goals weren't reported until the communication rate dropped to 10%, again suggesting that with larger group sizes missed goals are less likely.

In converse to the un-optimised and prediction failing DELPHÍS systems the optimised DELPHÍS system and stoplight system had virtually no missed goals at all in the MCM vignette. In the stoplight system this was because of the nature of the system where the goal order is predefined and the only way a goal could be missed is if it wasn't pre-programmed by the user. The DELPHÍS system successfully kept missed goals to a minimum by taking advantage of the mission optimisation techniques that it was designed with, and that were lacking in the previous two DELPHÍS system variants.

The pipeline tracking mission vignette provided similar data. Again the un-optimised and prediction failing DELPHÍS systems recorded more missed goals than the other systems however unlike the MCM mission vignette data the missed goal rate didn't improve as vehicle group size increased. It seems that the pipeline tracking mission wasn't as affected by larger group sizes, or more specifically not by group sizes up to 4. This could be because unlike the MCM mission vignette which has 5 legs, the pipeline tracking vignette essentially has 9 (3 groups of 3). Perhaps in order to see the number of missed goals decline like they did in the MCM mission a vehicle group size larger than 4 vehicles would have to be present. Unlike these two systems, the stoplight system and the optimised DELPHÍS system still successfully kept missed goals to bare minimum.

8.3.3 Target Acquisition

Unlike the redundancy and missed goal measurements previously presented, target acquisition wasn't greatly affected by the degeneration of communications between vehicles, regardless of the group size.

In the pipeline tracking vignette 100% of the targets were discovered in every single run with the exception of the DELPHÍS system with prediction failure as described in 7.3.3. This was mainly due to the nature of the pipeline tracking mission (Figure 6.6). In this mission there are three legs (broken up into sub-legs) that are 10 metres apart; one directly over the pipe and one 10 metres offset on either side. Because the AUVs were set to be able to detect targets within a 15 metre range, targets could be discovered from any of the three legs. This meant that any target had essentially three chances of discovery, resulting in an almost perfect target acquisition rate, regardless of how well or badly coordinated the mission was.

In the case of the DELPHÍS system with prediction failure, targets were missed due to one of two reasons. In the 2 AUV test there was one trial where one target was missed. This was because of an unusually inefficient run in 10% communications that led to the vehicles running out of batteries before the last target could be investigated. In all other situations the missed target was due to incorrect predictions where all vehicles predicted that another was accomplishing the task. Because these examples occurred in high communication loss trials, the missions were predicted completed before communications could return and synchronise the data (which would have alerted the AUVs to the incomplete target).

The mine countermeasures vignette had similar data. Though not 100%, the average number of detected mines stayed high throughout the trials regardless of communication loss. Like the pipeline tracking mission, legs were spaced 10 metres apart though unlike the pipeline tracking mission vehicles could only discover mines within a 5 metre radius. This resulted in more precise positioning requirements to assure mine detection. Despite this however, detection rates were still high.

The DELPHÍS and stoplight systems both had 100% mine detection in all communication rates and all group sizes. The un-optimised DELPHÍS system struggled a bit with 2 AUVs but improved as the group size grew. This was likely because of the fact that the fewer vehicles present the more likely that legs would be missed, and therefore targets left undiscovered. The DELPHÍS system with prediction failure fared better than the un-optimised one and like in the pipeline tracking vignette only decreased in target detection in high communication loss environments were there was a greater chance of incorrect predictions. In all cases the numbers improved as more vehicles were added to the system indicating that the more AUVs operating in the mission the better the chance a mine will be found.

8.3.4 Time

Before analysing the systems compared in terms of time it is important to see how mission time is affected by group size. Looking at Table 7.1 and Table 7.2 it is clear that in 100% communications as group size increases mission time decreases. However there is a limit to this decrease as shown by the MCM data. In Table 7.1 time decreases as the number of vehicles goes up however the 4 vehicle scenario reported the same

time as the 3 vehicle one. This is because in the 4 vehicle MCM scenario the last leg (leg farthest to the right in Figure 6.5) took a long time to be accomplished because it was the farthest away. In the 3 vehicle scenario this extra time wasn't noticeable because all other vehicles were busy while it was being executed. In the 4 vehicle scenario however most vehicles were finished and in holding patterns. Most of the mission was complete far earlier but the last leg increased the final time. It is expected that for this mission a similar situation would occur with 5 or more vehicles.

As communication rates drop, these times increase even more and like with redundancy and missed goals the time it took to accomplish the mission was noticeably affected by this drop in communications. In both the mine countermeasures and pipeline tracking mission vignettes mission time increased for most of the systems tested as the communication rate dropped.

In the MCM vignette all three versions of the DELPHÍS system started out with about the same average time since while the communication rate is 100% they are all essentially the same system. In the 2, 3 and 4 vehicle scenarios however as the communication rate dropped the un-optimised system began to take longer to accomplish the mission. This increase in time is due to the inefficiency of the run and as coordination was hampered by lack of information passing, more back-tracking was required resulting in longer and longer mission times. In addition, as shown in section 8.3.1 redundancy increased as communications dropped thereby increasing the time required to complete the mission. This became more pronounced as the group size increased. The optimised DELPHÍS system however did not show this increase in time. In all three group sizes the mission time remained mostly stable indicating that with the benefits of the system enabled it was able to cope with the communication loss. The data for the DELPHÍS system with prediction failure was similar with only a minor increase in mission time showing that even with incorrect prediction the system was able to reconcile mission errors efficiently.

The stoplight data for the MCM vignette was similar to the un-optimised DELPHÍS system in that mission time increased as communication integrity deteriorated. This was due to two things. First, vehicles often had to wait at certain points until other vehicles had checked in before continuing their mission, which caused delays. Secondly the increase in time in the stoplight system was often due to the percentage of

mission runs where the system got stuck and ran out of battery. As can be seen in section A.3 there were essentially two different times that the stoplight system took to complete the mission; an *expected* range of shorter times, and a longer time where the vehicles got stuck and ran out of power. As communication rates decline there is more waiting around and the chance of running out of power is higher. Therefore the probability of running out of power is inversely proportional to the communication rate.

In the 2 AUV stoplight scenario the initial mission time was significantly higher than that of the DELPHÍS systems. This was simply due to the fact that with 2 vehicles the stoplight approach (See Figure 8.1a) takes longer to accomplish the mission than the dynamically split up approach taken by the DELPHÍS system (note the battery life for this mission had to be extended beyond that of the other systems due to the longer *expected* mission time). This changes however as more vehicles are added to the system. In the 3 AUV scenario the stoplight system is initially on par with the other three systems and in the 4 AUV scenario it is actually initially faster. In both cases however the time increases significantly as communication drops whereas the optimised DELPHÍS system remains mostly constant.

The time data for the pipeline tracking mission vignette was similar to that of the MCM vignette in the 2 and 3 vehicle scenarios. In both cases the DELPHÍS systems started out at the same time and as communication degraded the un-optimised system took progressively longer, with the optimised system staying the most constant the prediction failure system only just below. In the 2 and 3 vehicle scenarios the stoplight system initially took longer than the DELPHÍS systems and steadily increased in time on average taking the longest. Again this data showed a range of *expected* mission times as well as a growing percentage of missions where vehicles got stuck and ran out of power.

In the 4 AUV mission however the data changed. As shown in Figure 7.24 the trend shown in the 2 and 3 vehicle missions is replaced with a slightly different one. Here all the versions of the DELPHÍS system show a somewhat similar trend as previously observed; the un-optimised system showing an increase in mission time as communication decreases, and the optimised and prediction failure systems staying about the same (there are some anomalous data but this will be explained in section 8.3.6). The stoplight system however abandons its previous increasing time trend in

favour of a relatively stable one that somewhat closely resembles that of the optimised DELPHÍS system (again, see section 8.3.6). This change in trend is due mainly to the fact that in the pipeline tracking mission 4 vehicles is the optimal setup with 3 vehicles handling the mission legs and one vehicle examining targets. Because this is exactly what was pre-programmed into the scripted mission plan it was inadvertently fast by default. This was to have a great affect on efficiency as will be explained in the next section.

8.3.5 Efficiency

Using the data collected about redundancy, missed goals, target acquisition and mission time, the efficiency of each mission run was calculated for each system type. In the MCM mission vignette the data showed many of the same trends that have been described in the previous sections (which is understandable since the efficiency data is derived from the data presented in the previous sections). For all three vehicle group sizes the un-optimised DELPHÍS system showed a clear decreasing trend in efficiency as communications worsened. This is an expected outcome as it has been shown that many of the metrics used in the calculation of this efficiency follow a similar pattern. The efficiency of the DELPHÍS system with prediction failure also echoed much of the previous data in that it performs well in most situations only showing a decreased efficiency in the very low communication rates. The optimised DELPHÍS system recorded even higher efficiencies, remaining close to 100% in all three group sizes. It too was affected by only very low communication rates and in these scenarios only very slightly.

The stoplight system efficiency varied depending on the group size. In the 2 vehicle scenario the system was initially notably less efficient than the DELPHÍS systems. This, as explained in the previous section, was because of the extra long time it took to accomplish the mission. In fact in the 2 vehicle scenario the mission time was the only negatively contributing factor to efficiency as there were no redundant or missed goals and all the targets were acquired. In looking at Figure 7.10 and Figure 7.13 one can see that the efficiency trend is the inverse of the mission time.

As more vehicles were added to the system however the data changed. Where it recorded the least efficiency before, now in the 3 vehicle scenario the stoplight system

improves and surpasses that of the un-optimised DELPHÍS system. Again this is mainly due to mission time though the increase in redundant goals had a slightly negative affect as well towards the lower range of the communication rate. In the 4 vehicle scenario the efficiency is virtually the same as with 3 AUVs. This is because though the mission time improved, the goal redundancy increased as well, consequently negating any time benefit in the efficiency equation.

An interesting note is that initially the stoplight system starts off with an efficiency value *above* 100%. This is because in full communications the stoplight system was able to accomplish the mission faster than the optimised system, which at 100% communication is the benchmark for comparison.

The pipeline tracking mission efficiency data showed many of the same trends as the MCM data. In the 2 and 3 vehicle scenarios the un-optimised DELPHÍS system again showed a clear decrease in efficiency as communications lessened. The prediction failure and optimised DELPHÍS systems also repeated the trends seen in the MCM mission vignette with the optimised system staying close to 100% efficiency and the prediction failure system either reflecting the same result, as in the 2 vehicle scenario, or falling just below it. Like with the stoplight system in the 4 vehicle MCM scenario the optimised DELPHÍS system's efficiency rises above 100% in both the 2 and 3 vehicle scenario (so does the prediction failure system). This is for the same reason mentioned before where in these cases the system was accomplishing the mission faster than it did in 100% communications.

The stoplight system's efficiency was again dictated mostly by time. In the 2 and 3 vehicle scenarios efficiency starts off significantly lower than that of the other 3 systems. This is due to the longer time required to accomplish the pipeline tracking mission with the stoplight system (see Figure 8.2). As expected, efficiency dropped with the communication rate though not as steeply as the un-optimised DELPHÍS system.

The 4 vehicle pipeline tracking scenario showed somewhat different results than the previous two scenarios. Here the 3 DELPHÍS systems showed the same relationship to each other with the un-optimised system recording the worst efficiency, the optimised

system recording the best and the prediction failure system falling just below the optimised. In all 3 systems however efficiency decline was greater than in previous scenarios. In addition, the stoplight system showed an entirely different trend compared to its past performance. Rather than starting off significantly below the others in terms of efficiency and slowly decline it started off *better* and only barely dropped below the 100% efficiency mark.

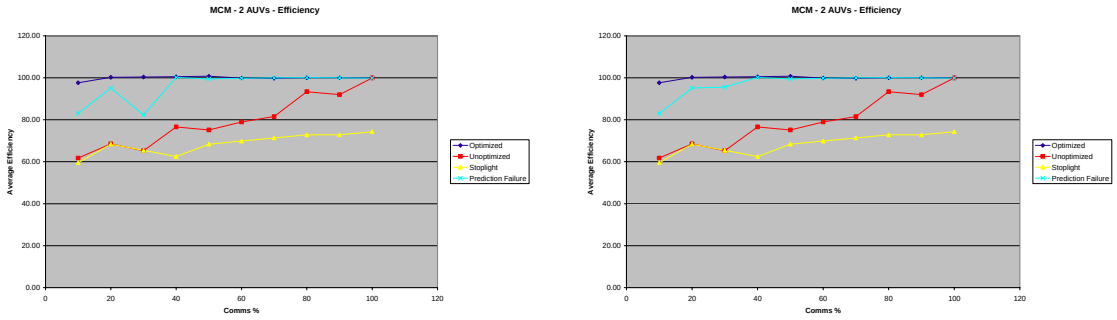
There are two main explanations for this change in relationship between the 4 multi-AUV systems. First of all like in the MCM vignette the more vehicles added to the system the harder it is to coordinate behaviours, especially in the lower communication rate environments. This in combination with the fact that the pipeline tracking mission vignette is more complicated than the MCM (9 legs as opposed to 5) is the most likely cause for the steeper efficiency decline in the 3 DELPHÍS systems. The cause of the stoplight systems efficiency improvement is much simpler. As it turns out the stoplight approach to coordinating 4 vehicles in this mission is most likely the best solution. In fact, as shown in Figure 7.27, the efficiency of the stoplight system is initially far above 100% due to the initial high mission speed described in the previous section. The stoplight system is still affected by the complexity of the mission which accounts for the dip in efficiency as communication makes coordination more and more difficult however it still remains very efficient.

8.3.6 Anomalous Data

Though most of the data explained in this section showed relatively clear trends there were some spikes and drops that stood out. At first glance they seem out of place however upon closer inspection they can be explained. A simple example can be found in the efficiency data recovered for 2 AUVs in the mine countermeasures mission. In Figure 7.13 (also shown in Figure 8.6*a*) the efficiencies are displayed for all 4 tested systems. The optimised and un-optimised DELPHÍS system show clear trends as does the stop light system. The DELPHÍS system with prediction failure however displays a strange drop in efficiency at 30% communications followed by a return to the expected trend at 20% communications.

At first this seems out of place but when the data used to calculate the efficiency of the mission is consulted the reasoning is clear. The time data (displayed in Figure 7.10)

shows a spike in the prediction failure data in the exact same place as the efficiency drop. Consulting the data it can be seen that of the 10 trials in the 30% communication tests 3 had 1 redundant goal. In addition in one of these 10 trials 4 goals were missed. The redundant goal trials resulted in longer mission times, and the combination of all three of these factors led to the dip in efficiency. When these 3 redundant goals are removed and the mission times adjusted in addition to removing the 4 missed goals the trend smoothes out as shown in Figure 8.6 where the original graph is shown in *a* and the adjusted graph in *b*.



a *b*

Figure 8.6 Adjusted MCM efficiency system comparison for 2 AUVs.

This example illustrates an important issue that came up in this research. By their very nature the numbers of redundant and missed goals as well as the mission time are completely random and can waver significantly between missions. Despite this however, they have direct effects on efficiency. Because this research uses relatively small trial sizes of 10 to keep experiment lengths manageable these random events can have large effects on the data (Though the trends are still visible). There are a few other examples of spikes and drops in the data due to this randomness and it is very likely that with larger numbers of trials the altitude and depth of these will decrease, if not disappear.

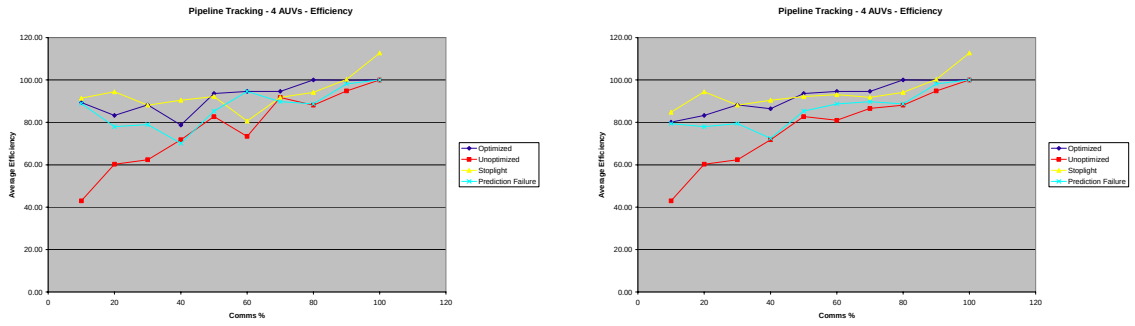
The example illustrated in Figure 8.6 is a mild example of this phenomenon however there are a few more significant ones. The most extreme example found in this research can be seen in the 4 AUV pipeline tracking mission vignette. As mentioned in section 8.3.5 the data in this scenario was significantly different to that of all the other runs. In this dataset all 4 systems recorded data that had anomalous spikes or drops resulting in the somewhat confusing graph shown in Figure 7.27. The lack of clear trends however can be explained when the data is looked at in more detail.

In the un-optimised DELPHÍS system following a relatively clear downward trend there is a bump in efficiency at 70% communications followed by a steep drop at 60%. This is again strange as efficiency then rises back up at 50% communications and continues its downward trend. When the mission time data is consulted the same phenomena are visible, though in reverse as lower time results in higher efficiency. The missed goal, redundancy and target acquisition data don't show any corresponding spikes/drops so it must be the mission time that is the issue. In fact when the time data for the un-optimised DELPHÍS system is consulted (see section B.3) it can be seen that at 70% the system never ran out of power (which would have resulted in a longer mission time) whereas at 60% it did on 3 separate occasions. This caused the jump in average mission time which consequently resulted in the drop in efficiency. As mentioned earlier in this section the number of times that missions will time out due to lack of batteries is random and with trial sizes of only 10, these events have major effects on efficiency.

The data for the stoplight as well as the optimised DELPHÍS system are very similar to that of the un-optimised system in that time was the biggest factor in determining the efficiency of the mission. Again when comparing the time data in Figure 7.24 to the efficiency data in Figure 7.27 it can be seen that the efficiency is essentially the inverse of the time in both systems (in addition to a minimal affect by the redundant and missed goal numbers). Like the un-optimised system when looking at the time data for each run in section B.3 the spikes and drops in the data are due to abnormally large and small numbers of missions where there was a battery time out.

The DELPHÍS system with prediction failure has different, but equally explainable data. As shown in Figure 7.27 there are three areas of the data that stand out. There are 2 spikes at 10% and 60% communications respectively and a drop at 40%. The spikes in efficiency are directly due to the phenomenon previously described where the number of battery related mission timeouts were abnormally low as can be seen in section B.3. The drop in efficiency at 40% communications however was due not to time but to an abnormal spike in missed goals (1 run had 6, compared to the others which averaged 2-3). As mentioned earlier in this section many of these data anomalies in this study would likely be averaged out were the trial sizes larger than 10. This is one of the main aspects of this work that the author would like to improve upon in the future.

When these anomalous times (in addition to the one missed goal run for the prediction failure DELPHÍS system) are pruned to simulate the expected smoothing of the data with larger trial sizes the result is a smoother graph showing clearer trends (see Figure 8.7).



a *b*

Figure 8.7 Adjusted pipeline tracking efficiency system comparison for 4 AUVs.

In addition to the randomness of some of the data there is another possible cause for these data anomalies. As mentioned in section 8.3.1 working in controlled simulation environments, though useful for testing, can cause very unique problems in terms of unlikely events. We saw that in the case of 4 AUVs executing the MCM mission using a stoplight controller there was a synchronisation issue that led to abnormally high initial redundancy. Because of the controlled simulation, this unlikely event was replicated for virtually every trial, something that would almost certainly not occur in the real world. It is possible that there are other, less obvious examples of this happening in these experiments that could cause spikes and drops in the data.

8.4 Real World Validation

In addition to the simulated experiments real word trials were conducted to validate the simulated results. This section will explain the results of these trials, focusing first on the AUV/Simulated AUV experiments and then on the multi-AUV data.

8.4.1 AUV / Simulated AUV Trials

As mentioned in section 7.4.2 the goal of the AUV/Simulated AUV trials was to validate the coordination ability of the DELPHÍS system without the added complication of two mobile robots. Three trials were conducted using an MCM vignette and two AUVs; Nessie III and a simulated AUV running on a computer on

shore. These trials served as an initial shakedown of the system and a number of bugs were discovered and fixed as a result. These bugs will be described first, followed by the results of the trials.

8.4.1.1 Discovered Software Bugs

Although minor, one of the bugs discovered upon consulting the logs (see Appendix C) after the missions was that the times of received broadcasts was not correctly recorded in the log files. This makes it challenging to clearly see when vehicles were and weren't communicating (however much can be learned from looking at the vehicle behaviours).

A second more serious bug was discovered in the way target data was added to the mission model. When working in simulation, target data was always exact (and in local frame) and therefore to determine if a transmitted target had previously been added to the mission its x and y coordinate values were checked to see if they were exactly the same. When the DELPHÍS system was applied to real vehicles however targets were notated in global frame. These coordinates were then converted back to local frame to be transmitted acoustically (Section 7.4.1) thereby losing a small bit of accuracy which was amplified when the coordinate was converted back to global on the receiving vehicle. This prevented the legacy "exact coordinate comparison" from differentiating between targets correctly, resulting in duplicates. This was most evident in the third trial after which the bug was discovered and repaired.

8.4.1.2 Trial 1

In the first trial the mission was limited such that only one vehicle could attempt the search. Because Nessie started first this goal was selected resulting in the simulated vehicle having to wait in a holding pattern until goals (discovered targets) became available. Once the first target was discovered the simulated vehicle began the investigation process. The second target however was investigated by Nessie. This was the result of the second target being discovered during a period of no communication between the vehicles. It can be seen on line 45 of the log file in Appendix C.1 that Nessie predicted that the simulated vehicle would attempt the recently discovered target. In the time between this prediction and the completion of the "Search" super-goal (waypoint 8) Nessie received an update from the simulated vehicle that showed that in fact the target was not being executed and was therefore reset to available in the mission

model. Before the information of the original target discovery could be broadcast Nessie began investigating the second target. This trial also illustrates accurate prediction of vehicle intention, specifically the simulated AUV prediction of Nessie actions as shown by the accurate predictions of waypoints 3, 4, 5 and 6 (illustrating a likely communication blackout period).

8.4.1.3 Trial 2

Trial 2 removed the target discovery aspect of the mission but also removed the limitation on the search so that multiple vehicles could concurrently achieve the goal. This resulted in the lawnmower being broken up into its component legs. The mission was completed as expected with Nessie executing a majority of the legs due to its higher speed. The simulated vehicle attempted execution of waypoint 7 but stopped, as can be seen in the aborted path in Figure 7.32. This behaviour was due to a period of no communication and a delayed and therefore incorrect prediction. On line 28 of the log file in Appendix C.2 it can be seen that Nessie predicted that the simulated vehicle would next attempt waypoint 3 when in fact it was seconds away from executing waypoint 7. Due to this incorrect prediction Nessie proceeded to finish its current goal and start waypoint 8, a goal in an execution lock with waypoint 7. During this execution Nessie sent a broadcast alerting the simulated AUV to the problem which resulted in it's aborting the goal (lines 33-35). Goal redundancy was avoided and the mission was completed without a problem.

8.4.1.4 Trial 3

In trial 3 the targets were re-added to the mission while the search remained unconstrained. This allowed both AUVs to detect targets and consequently resulted in the duplicate target bug mentioned earlier in this section. As can be seen in Figure 7.33 the search was broken up between the vehicles and targets were investigated as they were discovered. Due to their high priority targets took precedence over search legs and therefore were investigated as soon as they were available. However because of targets being incorrectly duplicated the vehicles investigated targets more than once. In addition it can be seen that Nessie attempted waypoint 7 while the simulated AUV was already executing that locked leg. As in trial 2, the DELPHÍS system was able to recognise this conflict and resolve it, avoiding any mission redundancy (lines 64-71 in Appendix C.3).

8.4.2 Multi-AUV Trials

Having fixed the bugs discovered in the AUV/Simulated AUV trials the DELPHÍS system was then used to coordinate two AUVs, REMUS and Nessie III. Like the first trial in the AUV/Simulated AUV experiments the lawnmower search pattern was locked so that only one AUV could achieve it. As mentioned earlier this was due to the fact that the search was of a large area and the Nessie III vehicle didn't have the speed to accomplish it. In all three trials REMUS was started first so that it would begin the search goal and discover simulated targets. Nessie was then started, the two vehicles registered with each other and then it waited until there were available goals.

8.4.2.1 Trial 1

In trial 1 the mission was executed exactly as expected however due to the last target's location being so close to the end of REMUS's last leg it was taken on by REMUS before being broadcast to Nessie. This resulted in the targets being split between the vehicles, as can be seen in Figure 7.35. An interesting behaviour was discovered in this trial and then witnessed in all the following multi-AUV trials done during this trip. When the vehicle executing the last available goal in the mission (in this trial this was REMUS executing the second target) finishes the goal it recognises the mission is complete and exits just after a final message is broadcast to notify other AUVs of its status. However in the real world experiments where communication was unreliable (particularly so for REMUS as will be explained later in section 8.4.2.3) this broadcast is often not received. Despite this, Nessie was able to predict that REMUS had finished the goal thereby rendering the mission complete. This behaviour is evident in lines 68-75 in the log file in Appendix D.1.

8.4.2.2 Trials 2 & 3

For the remainder of the trials (2 & 3) the second simulated target was moved north by about 40 metres so that the Nessie III vehicle would have a better chance of selecting it before REMUS. Although the data returned after the first trial was good, it was thought that by moving the target Nessie would be more active and this would act as a more difficult scenario for the DELPHÍS system to coordinate. This proved successful and in both trials 2 and 3 Nessie executed both targets while REMUS waited in a holding pattern following the completion of the lawnmower search (Figure 7.36). In both trials the prediction of the mission completion behaviour mentioned in the previous section was evident. In addition trial 3 showed more examples of prediction where Nessie was

able to correctly predict the actions of REMUS within reasonable degrees of time error. This can be seen in lines 55-56, 68-71 and 81-86 of the log file in Appendix D.3. In both trials 2 and 3 the mission completed successfully with no redundancy or missed goals.

8.4.2.3 REMUS Communication

Despite the good results of the multi-AUV trials there was a significant issue that was discovered with the way REMUS handled custom acoustic communication. As mentioned in section 7.4.1 the REMUS vehicle comes with an acoustic modem built in which was used for this research. To access this modem however the onboard Ocean Systems Laboratory PC104 computer must send a special user-message command to the onboard REMUS computer, which has exclusive access to the modem. The data contained in this message isn't sent until it is queried from another modem (in contrast to the modem on Nessie III which has full control and can send messages whenever). In order to enable the DELPHÍS system to send acoustic broadcasts a program was written that used a third WHOI MicroModem on shore to query REMUS and ask it to send the latest message.

In these multi-AUV trials the DELPHÍS system operating on the REMUS vehicle was programmed to send a broadcast message every second, while the polling program on shore was programmed to poll every 15 seconds. The hope was that in this manner the most recent message would be sent every time it was queried. What was discovered however was that although REMUS sent a message when queried (due to the unreliability of underwater communication on average messages were received every 37.88 seconds) the messages sent were not necessarily the most recent. In fact, in most missions there were only about 5 distinct messages sent by REMUS. This resulted in very few updates being sent as can be seen in Figure 8.8 where REMUS's path is very jagged (shown in yellow) in comparison to the mission (shown in blue).

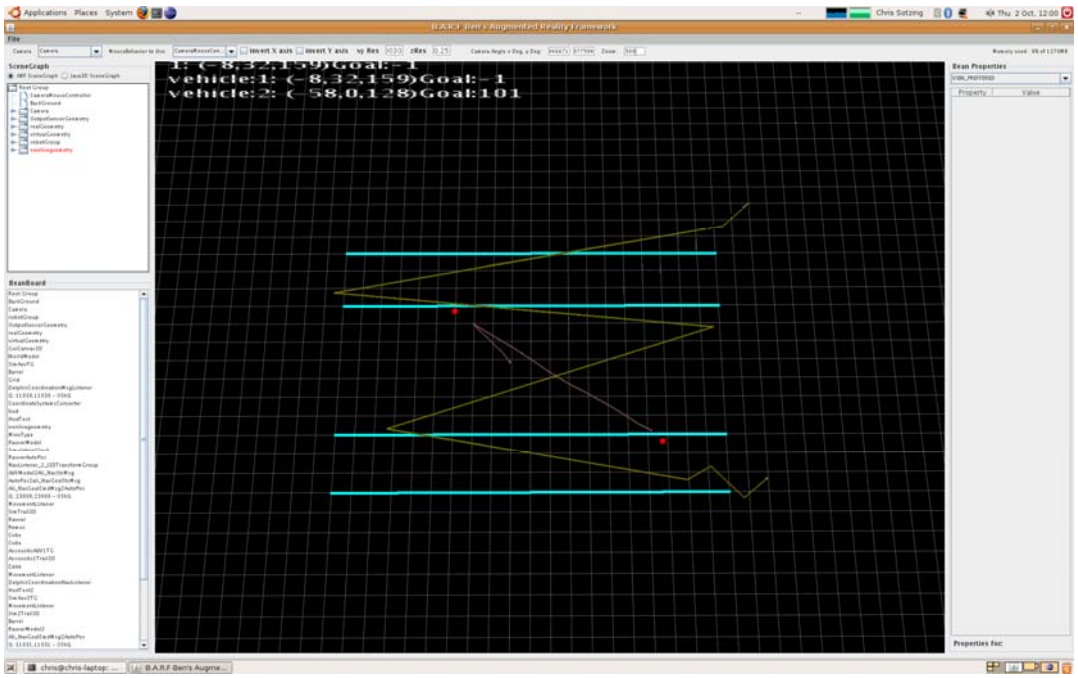


Figure 8.8 An ARF screenshot showing the mission progress of REMUS & Nessie III.

Although this was a limitation that meant that the average time between unique broadcasts was closer to 159 seconds the DELPHÍS system was still able to coordinate vehicle actions successfully. In addition because Nessie III was programmed to broadcast every 20 seconds (which in practice averaged out to be 28.23 seconds between received messages) REMUS and the shore monitoring computer were kept well up to date as can be seen by the much more detailed path of Nessie in Figure 8.8.

8.4.2.4 Simulation Validation

To validate the simulated findings of this study the multi-AUV mission run at Loch Earn with the real AUVs REMUS and Nessie III was also run in simulation. Using simulated versions of both vehicles the exact same mission used in the real tests was run. The Nessie vehicle was programmed to broadcast information every 29 seconds to mimic the average 28.23 seconds seen in the trials and REMUS was limited to broadcast every 159 seconds to represent the average time between unique broadcasts. Aside from these modifications all the code used was the exact same as in the aforementioned real world experiments. The logs from this experiment can be found in Appendix D.4.

As was expected the results from this simulated experiment was virtually identical to the data returned from the in water trials (particularly trials 2 and 3 as this experiment used the updated target position). It can be seen that as REMUS executed the

lawnmower search, Nessie waited in a holding pattern until targets were available. Once these targets were discovered and broadcast (both were broadcast at the same time due to communication lag, lines 38-40 in the log file) they were executed in turn. This simulation also mimicked the prediction behaviour first explained in section 8.4.2.1 when Nessie finished the second target. The only difference between the simulated data and the real mission was the shorter simulated mission time. This was down to the fact that the Nessie simulation was slightly quicker than the real vehicle. This affected the REMUS time as well since it would wait until the target investigation was complete before exiting.

8.5 AUV Group Size

As mentioned in section 8.3 one of the biggest effects on efficiency aside from communication is the number of AUVs working in the mission together. In large missions having too few vehicles can result in long mission times. However large group sizes can also be hazardous, particularly in bad communication environments, due to the number of concurrent decisions that are constantly being made. The challenge is to have just the right number of vehicles for a mission to minimize the time required to complete it while also minimising the number of simultaneous vehicle decisions.

The current state of the art in multi-AUV coordination easily illustrates this issue. In stoplight systems where vehicle actions are decided before the mission takes place, the wrong number of vehicles is extremely important. In the 2 AUV mine countermeasures mission for instance 2 vehicles took a long time to accomplish the mission and as a result were significantly less efficient than the other systems. On the other side of the spectrum was the 4 AUV pipeline tracking mission where just enough vehicles were used resulting in good efficiency that rivalled even that of the optimised DELPHÍS system.

These examples illustrate a setback in the stoplight system that is handled by the DELPHÍS system. In virtually all examples the DELPHÍS system maintained a very high efficiency throughout, regardless of the group size. This was because of the nature of the system where vehicles make mission decisions on the fly and can optimise/recover from conflicts when necessary. As group size increases the stoplight

system is more and more affected by communication loss (this is also true of the un-optimised DELPHÍS system and prediction failure DELPHÍS system to a lesser extent). The DELPHÍS system was designed to handle this situation and in result is not affected by the size of the collective in the same manner.

8.6 Key Performance Indicator

When looking at this data it is important to look back upon the original metrics used to calculate multi-vehicle efficiency and determine which had the greatest effect on the results. In this research the most influential metric, or key performance indicator (KPI), was mission time. Although there were only minor benefits over the stoplight system in terms of goal redundancy, missed goals and target acquisition the DELPHÍS system was far better at keeping mission times down due to its ability to optimise on the fly. Conversely the stoplight system was unable to resolve mission conflicts and wasted a lot of time, often resulting in an inability to complete the mission. This had a major effect on efficiency and shows a clear benefit of the mission optimisation techniques of this research including agent prediction and dynamic goal re-selection. It also illustrates the need for a more robust control architecture that can handle the kind of coordination errors that are likely in practice.

8.7 Conclusion

Having tested this research against the current state of the art in multi-AUV coordination it has been shown that by using the DELPHÍS system mission efficiency is increased. Optimisation techniques such as agent prediction of intent and dynamic mission execution have allowed for more robust mission execution and have resulted in maximising all of the efficiency metrics, especially mission time which has proven to be the KPI. Although there was some anomalous data this can be easily explained due to relatively small sample sizes and with more experiments these would likely disappear. Finally, the use of the system to coordinate real vehicles has proven its ability to work in the uncertainty of the marine environment and its accurate repetition in simulation has proved the validity of the simulated results.

Chapter 9

Conclusion

9.1 Summary

This research has investigated the use of a multi-agent based control architecture to coordinate multiple autonomous underwater vehicles and to increase efficiency over the state of the art. This chapter will present the achievements and contributions of this study as well as the conclusions that can be made from them. The next section will detail the achievements of the work followed by a section describing the novel contributions. This is followed by future work and finally a section explaining the recommendations that can be made from the results of this work.

9.2 Achievements

Based on the results shown in Chapter 7 and then discussed in Chapter 8 a number of conclusions can be made, both about the different multi-AUV coordination architectures tested and about multi-AUV operations themselves. In section 1.2 at the beginning of this thesis the research objectives were listed. In relation to these objectives the achievements of this research can be summarised as follows:

- Successfully designed, created and demonstrated the DELPHÍS system, a multi-AUV control architecture able to coordinate multiple AUVs in poor communication environments.

- Proved that as communication rates drop this research is able to coordinate multiple vehicles more efficiently than the current state of the art.

- Determined the key performance indicator (KPI) of multi-AUV operations to be mission time.

- Showed that multi-AUV systems are superior to single vehicle systems in the most common vignettes.

- Validated this research by using it to successfully coordinate the REMUS and Nessie III AUVs in a mine countermeasures (MCM) mission.

The first objective was to create a multi-agent control architecture for an autonomous underwater vehicle that was able to work in conjunction with other vehicles to coordinate behaviours. This objective has been achieved with the design, creation and testing of the DELPHÍS system which using its multi-agent based architecture has proved itself able to efficiently coordinate up to 4 AUVs (and very likely many more). In addition its novel combination of agent prediction of intent, dynamic goal execution and mission optimisation techniques allow it to remain efficient in communication poor environments.

The second objective in this research was to compare the DELPHÍS system to the current state of the art in multi-AUV coordination architectures (stoplight systems) and determine whether its functionality would render it more or less efficient. A stoplight system was simulated and along with three versions of the DELPHÍS system was tested on two of the most common multi-AUV mission vignettes in differing communication environments that attempted to simulate real world conditions. Based on the data recovered from these experiments it has been shown that the efficiency of the fully optimised DELPHÍS system surpassed that of the others in most scenarios, being no less than on par in one (4 AUV pipeline tracking). In addition it consistently returned shorter mission times than the other systems in virtually all cases, a major benefit seeing as how aside from mission accuracy the faster a mission can be completed the better.

The third objective was to investigate the metrics used to calculate coordination efficiency and determine how these affected efficiency as a whole. Results have shown that although missed goals, redundant goals and target acquisition have a clear effect on efficiency it is mission time that is the key performance indicator (KPI). The more efficient the system the less time it will take to accomplish the mission. This research has shown that the DELPHÍS system is able to keep mission time generally constant as

the communication rate decreases while the stoplight system returns progressively longer times.

The fourth objective was to determine the benefit of multi-AUV systems over single vehicle approaches and based upon the results the benefits of multiple autonomous underwater vehicle missions over single vehicle ones is clear. As shown in sections 7.2.4 and 7.3.4, multi-AUV systems cut mission time down by significant amounts and in a world where time is money (and battery life is finite) this is extremely important. As AUV collectives grow in number however it was shown that the need for a good coordination architecture becomes paramount. More vehicles mean more concurrent decisions which mean more chances for conflict. Without the ability to foresee and also rectify these conflicts multi-AUV systems will be forever limited. The DELPHÍS system proposes a solution to this problem and the results obtained in this study validate its usefulness.

Finally, having demonstrated the DELPHÍS system in real world trials with the REMUS and Nessie III AUV, this work proved itself as a viable option for coordinating multiple AUVs in actual situations. These trials also served to validate the results obtained in simulation and strengthen the claims that the DELPHÍS system is more efficient than the current state of the art in multi-AUV coordination.

9.3 Novel Contributions

In addition to the achieved objectives presented in the previous section this work is responsible for and enabled by a number of novel techniques that are new to autonomous underwater vehicle systems and help it to achieve its goal of efficient multi-vehicle coordination. These contributions are summarised below:

- Prediction of intent to facilitate coordination when communication cannot be depended upon.

- Dynamic mission execution to eliminate the need for and limitation of pre-scripted goal order.

- Communication via a simple acoustic broadcast system that allows for simple group size scaling during mission run time.

- Mission optimisation tools to reconcile conflicts when they occur.

One of the main contributions of this research is the use of agent prediction (via the recursive modelling method (RMM)) to make logical assumptions about vehicle intent when communication isn't available. By utilising the local decision making structure with information of other vehicles AUVs are able to make accurate predictions and consequently make accurate decisions even when out of contact with the collective. This has shown to have a major effect on keeping mission errors down and therefore maximising mission efficiency.

Unlike the rigid pre-scripted goal execution utilised by the current state of the art in multi-AUV coordination this work, through the use of the BIIMAPS system, is able to query the mission plan and determine the most suitable task to execute at any given time. This has resulted in both the simplification of mission planning (since only one plan is required instead of one plan per vehicle) as well as the ability to execute missions in the most optimum order given the state of the world.

The use of a simple acoustic broadcast communication system has allowed for vehicles to pass information in a way that avoids the need for message acknowledgement, a challenge in environments where acoustic messages are often lost. In addition because broadcasts include each vehicle's mission history this has enabled new vehicles to enter the mission at any time without any prior knowledge programmed into the mission.

Regardless of the control architecture, in the marine environment some coordination errors are inevitable. To handle this eventuality this research has employed mission optimisation tools to recognise these errors and reconcile them so that the mission can continue un-phased. These techniques include the ability to recognise mission conflicts such as two vehicles attempting the same goal as well as the ability to roll back mission plans when confronted with incorrect predictions. This has shown to enable the system to handle long intervals of no communication and the resulting conflicts that occur when communication returns.

9.4 Future Work

Despite these achievements, like all research there are a number of areas which could be improved upon and or studied further. An important next step to this research would be the incorporation of a deliberative layer mission planner like that found in [53]. The optimisation techniques employed by the DELPHÍS system would allow most issues to be solved in the executive layer however any unsolvable missions could then be passed back to the planner to be re-planned. Like the incorporation of a high level mission planner another addition to the architecture would be a unified ontological world model. Currently there are a number of databases in the DELPHÍS system including the AUV database and the mission representation. In the future it would be prudent to take all of these databases and concatenate them into one world model shared by all modules in the architecture.

In addition to these architecture improvements there are a number of ways that the DELPHÍS system itself could be improved. One of the most important and powerful aspects of the system is its ability to predict the actions of other AUVs. This prediction ability can be improved in a number of ways. First of all prediction could be extended to not just the next move but the possible next *few* moves. This would require attaching a certainty value to predictions as the farther forward actions are guessed the less certain they become. This could allow for more accurate vehicle coordination and possibly even improve goal selection in good communications where vehicles could choose their own goal based on where other vehicles might go in the future.

Another possible direction for further research is the application of machine learning techniques to optimise coordination over time. A good example of a place where learning would be beneficial is the prediction just mentioned. If vehicle prediction included certainty values the system could learn at which point to trust its predictions and when to ignore them. The OBSERVER system [79] is an example of a system where agents are able to learn the outcomes of predictions and apply this knowledge to future ones.

9.5 Recommendations

Based on this research there are a number of recommendations that can be made about multi-AUV operations from an operational standpoint, whether or not the DELPHÍS

system is being utilised. First and foremost for missions like MCM and pipeline tracking where there are distinct search and target identification aspects it has been found to be prudent to break up the search task between vehicles. This is clear in both the MCM and pipeline tracking vignettes were the efficiency of the 2 vehicle stoplight system was severely limited due to the time it took for one vehicle to search the area. When the search tasks were broken up, either dynamically using the DELPHÍS system or manually with a stoplight system, the efficiency improved. Thus, if resources allow, the more the search can be divided the better.

Another related recommendation has to do with group size. Whether or not a dynamic system like the DELPHÍS system is used, the size of the collective is important. Too few vehicles and the mission may take too long, lowering efficiency. It can be said that the size of the AUV collective depends directly upon the complexity and size of the mission. Too many vehicles and there may be too few tasks to go around resulting in vehicles constantly waiting in holding patterns. A good example from this research can be found in the pipeline tracking data. As mentioned in Chapter 8 for this vignette 4 vehicles is most likely the optimal group size, as shown by the stoplight data. With 3 vehicles the search isn't broken up enough and the data suggests that a group size of 5 would result in one vehicle having little if anything to do.

When looking at this research and these recommendations the obvious question is whether or not control architectures like the DELPHÍS system are worth it. Given the optimal number of vehicles the stoplight system does an equally good job of coordinating the mission as shown by the 4 vehicle pipeline tracking data. Though this may be true there are a number of reasons that the use of a dynamic, intelligent system like DELPHÍS is a better option. First of all mission definition is far simpler. Users only have to define one mission and don't have to worry about what each vehicle will do to accomplish said mission or even how large the group will be since this is all handled by the system at runtime. Another reason that the DELPHÍS system is a better choice is its ability to optimise on the fly. This optimisation takes many forms from vehicle intent prediction to the ability to add more vehicles as required while the mission is being accomplished. These benefits have been shown to minimise mission errors and consequently maximise coordination efficiency in realistic environments. This functionality is unavailable to current multi-AUV coordination systems thereby proving the worth of this research.

In the end the choice of the DELPHÍS system over the current state of the art simplifies to flexibility. Though stoplight systems can successfully coordinate multiple vehicles in good conditions this research has shown that when these conditions deteriorate so too does mission coordination efficiency. The DELPHÍS system can maintain efficiency in a much wider range of conditions and this makes it a good choice for multi-AUV operations.

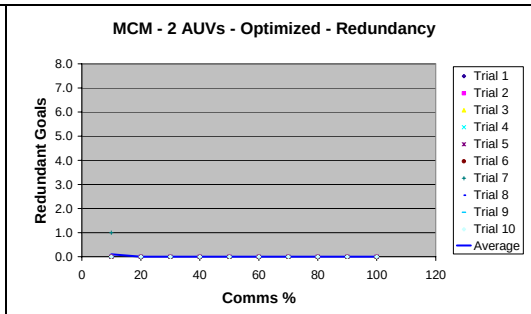
Appendix A

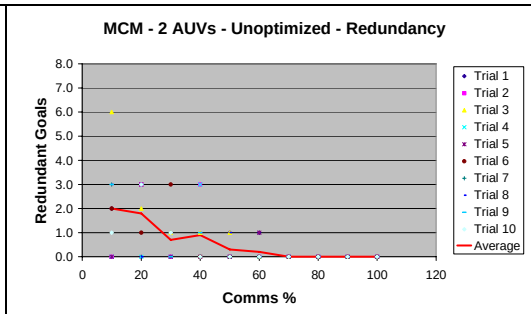
Mine Countermeasures Data

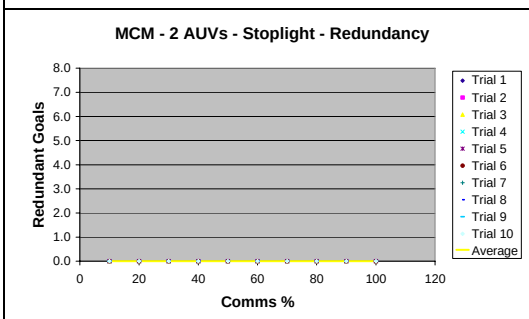
This appendix contains the individual efficiency metric data for the mine countermeasures mission. Data is presented for each of the four systems (DELPHÍS system, DELPHÍS system un-optimised, DELPHÍS system with prediction failure and stoplight system) for groups of 2-4 vehicles.

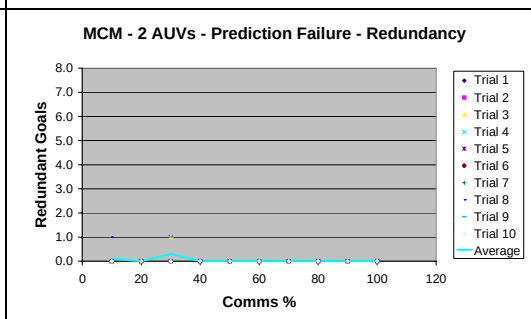
A.1 MCM – Redundancy

2 AUVs

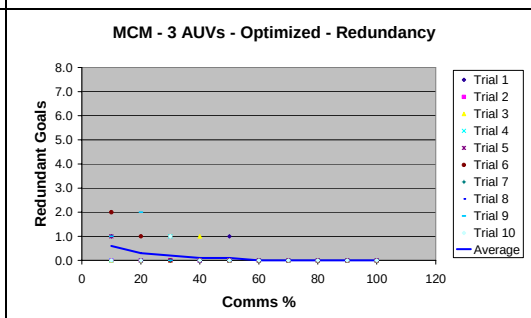


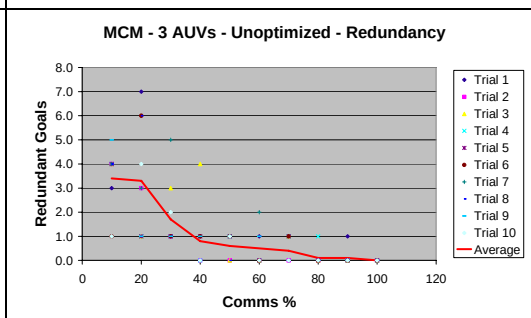


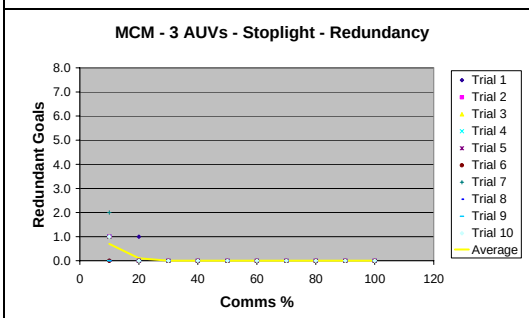


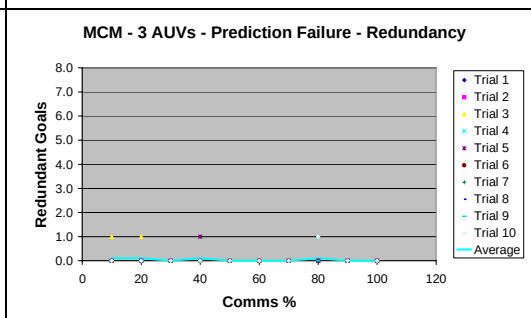


3 AUVs

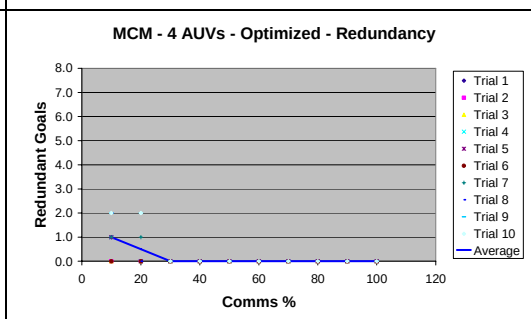


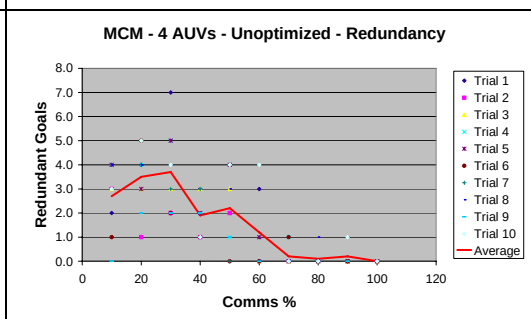


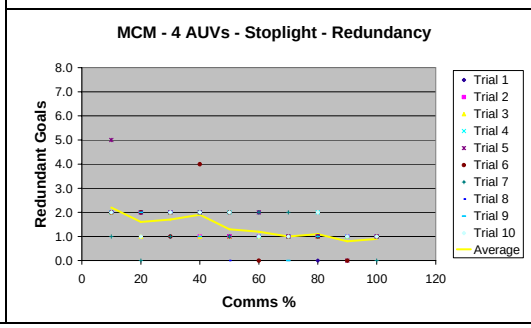


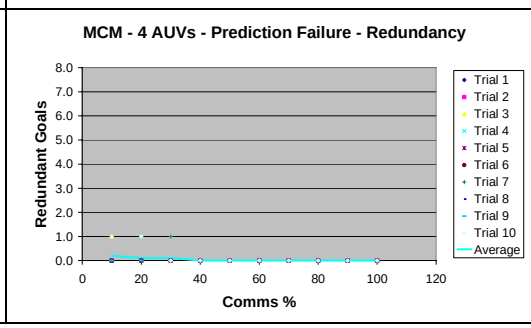


4 AUVs



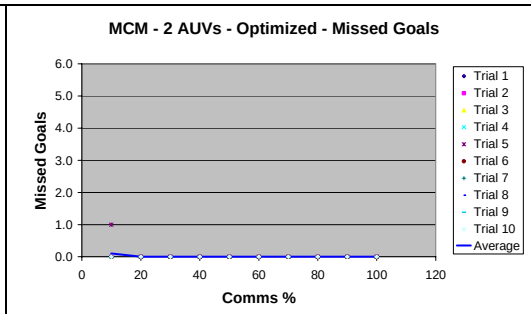


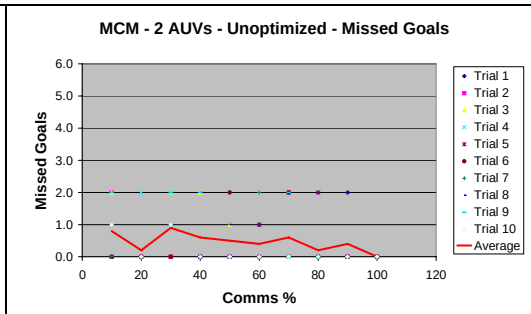


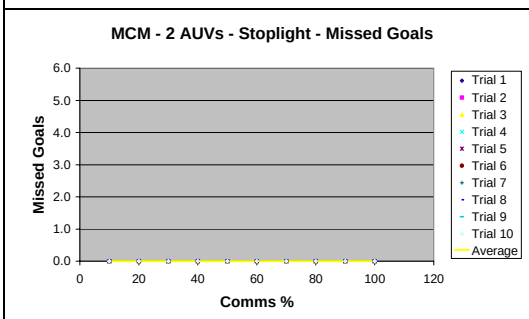


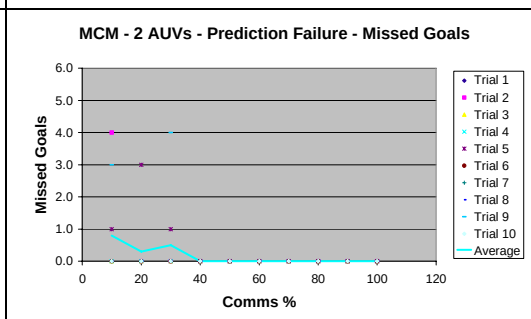
A.2 MCM – Missed Goals

2 AUVs

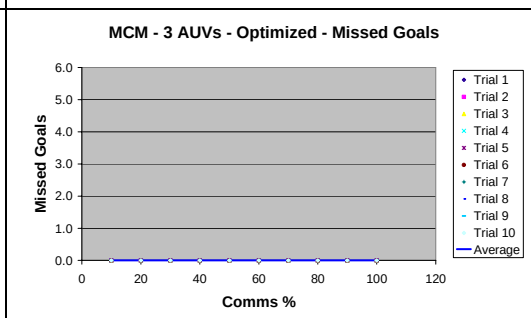


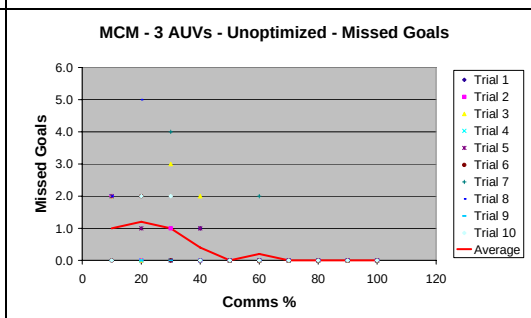


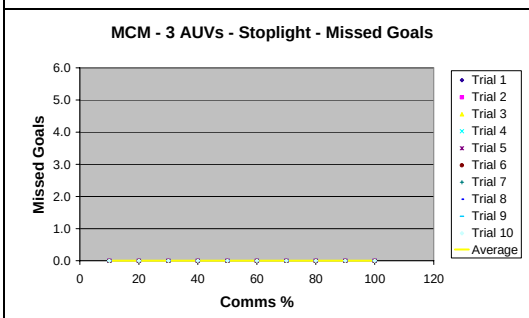


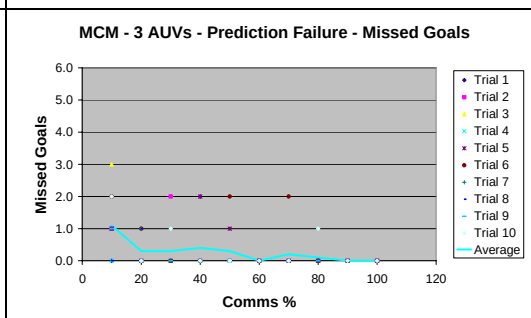


3 AUVs

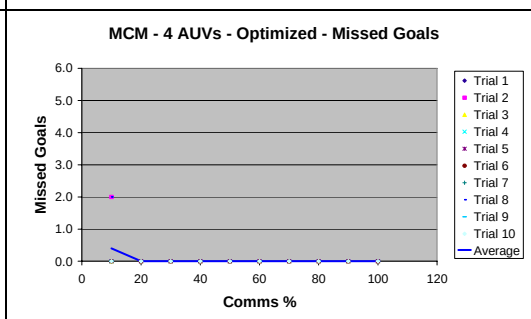


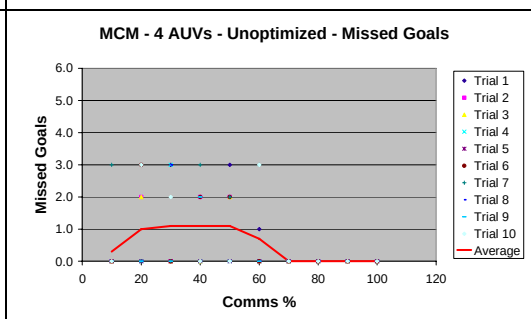


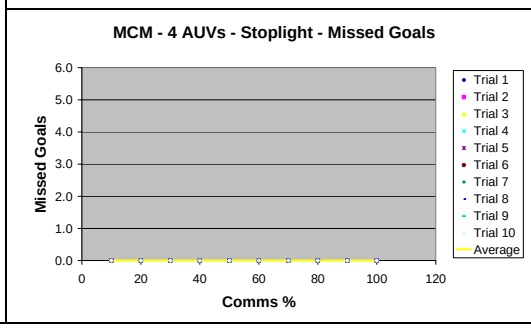


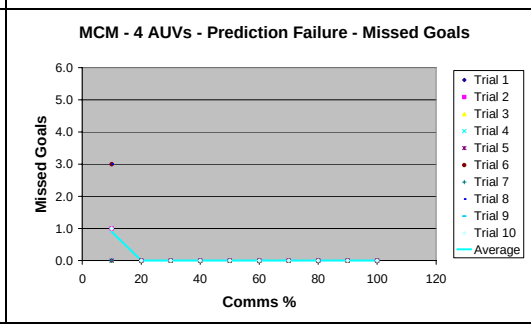


4 AUVs



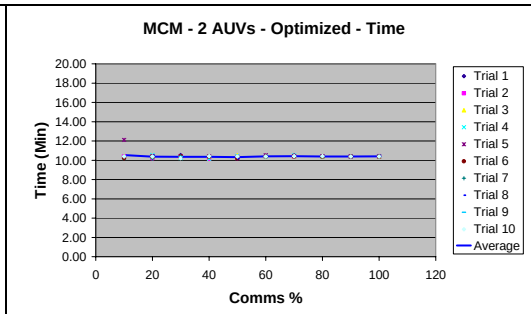


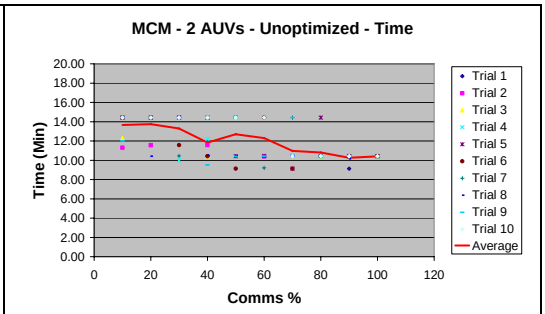


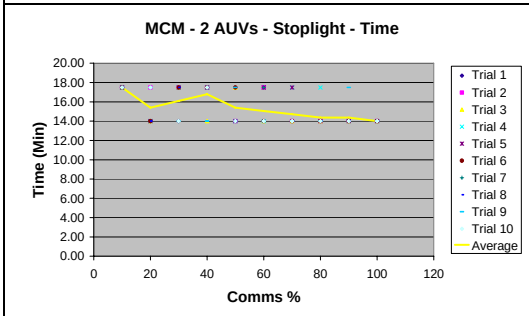


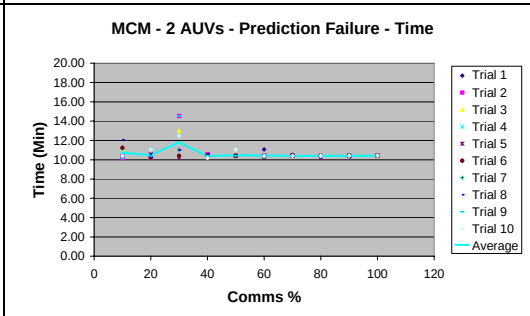
A.3 MCM – Time

2 AUVs

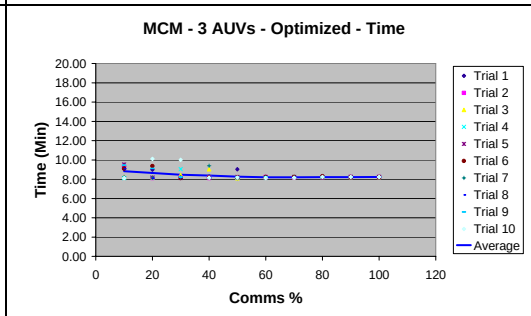


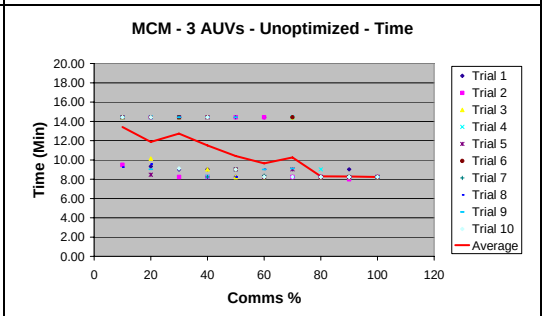


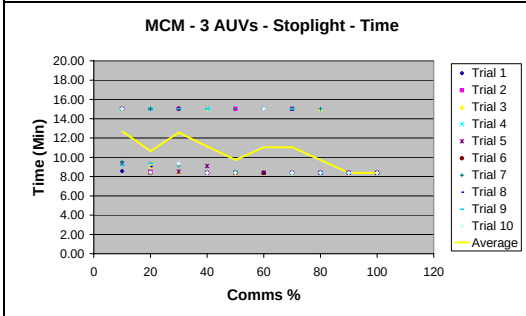


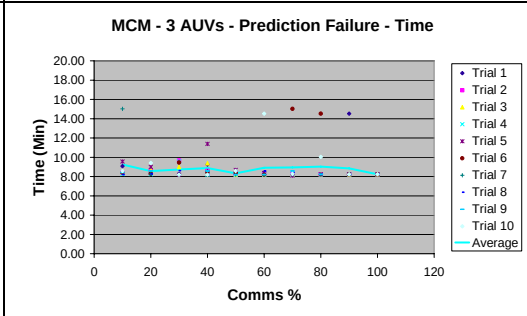


3 AUVs

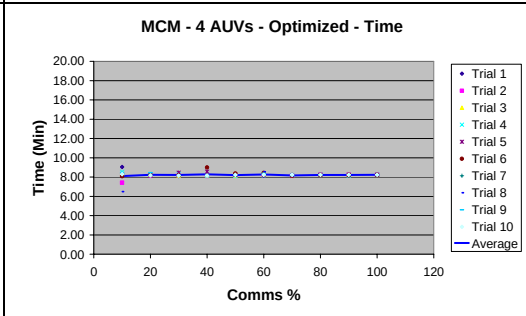


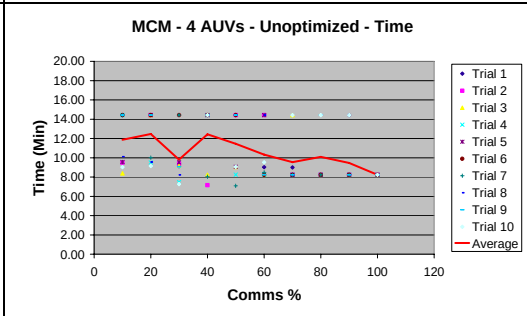


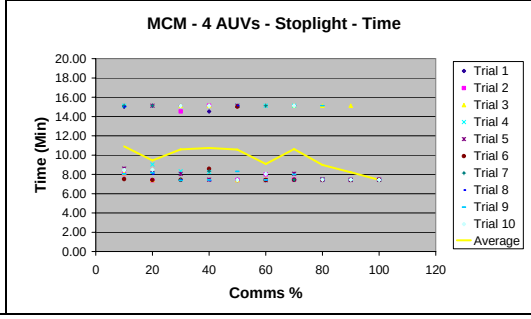


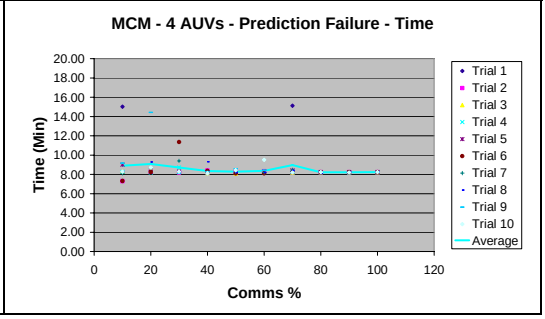


4 AUVs



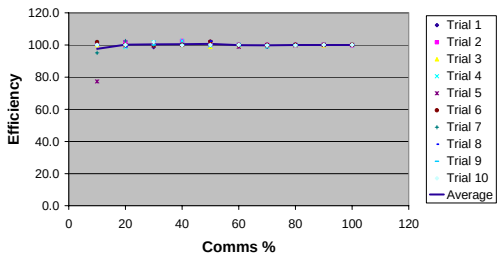


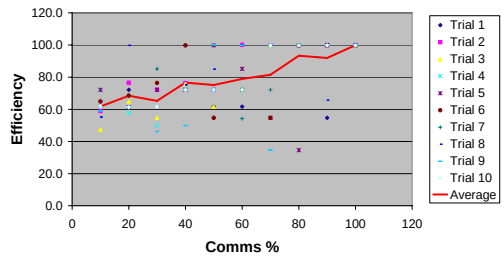


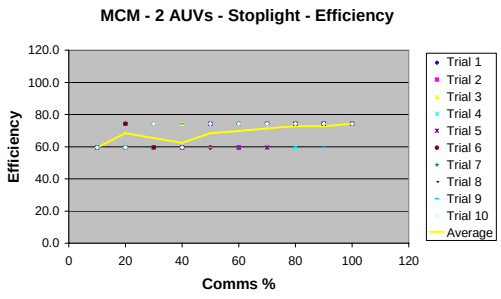


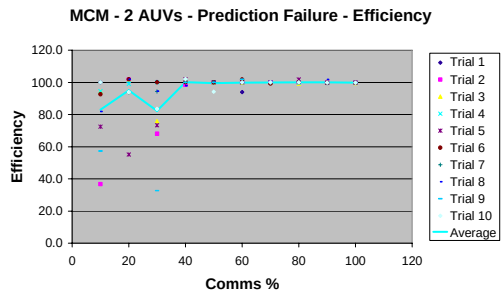
A.4 MCM – Efficiency

2 AUVs

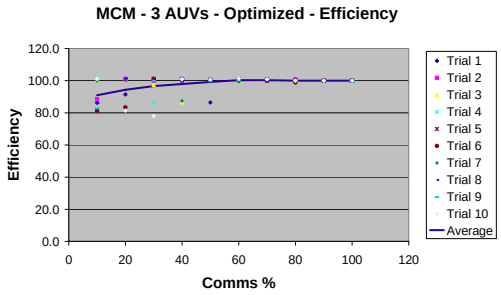


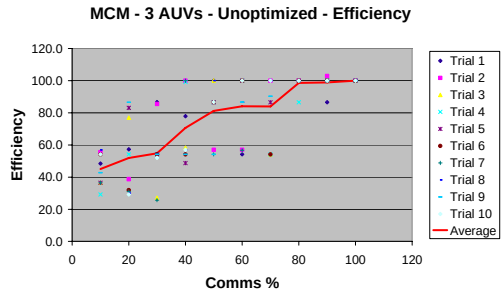


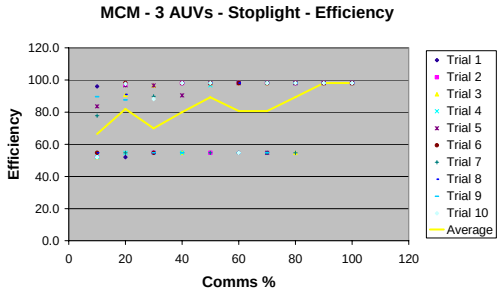


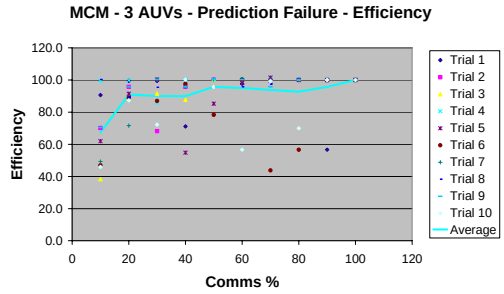


3 AUVs

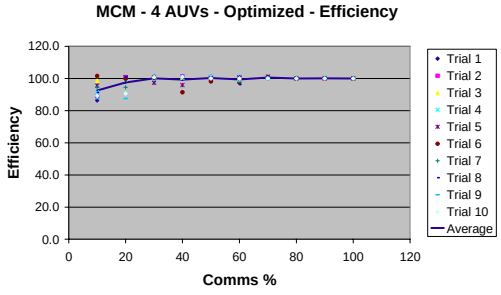


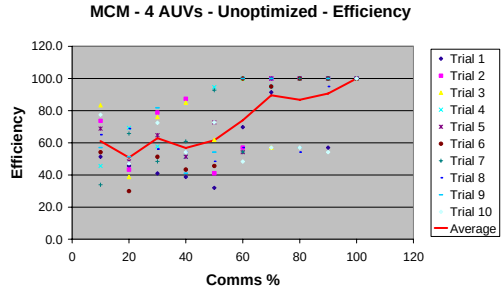


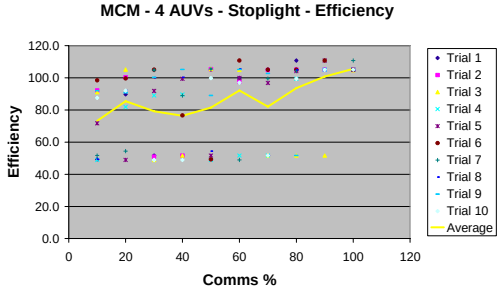


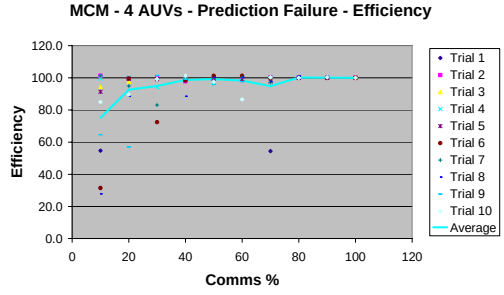


4 AUVs









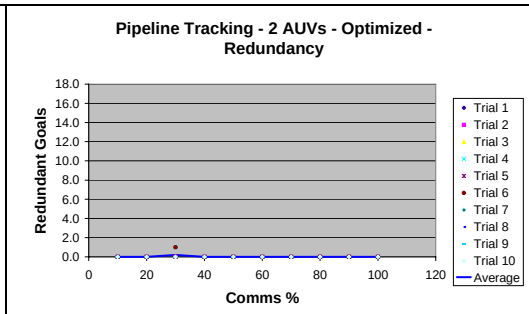
Appendix B

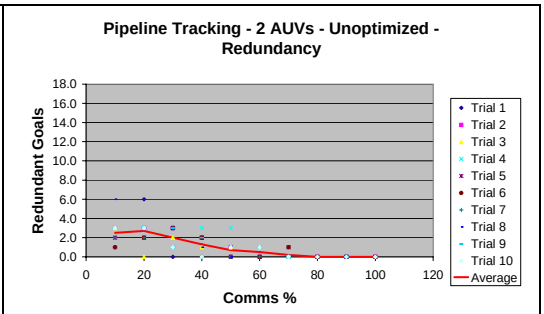
Pipeline Tracking Data

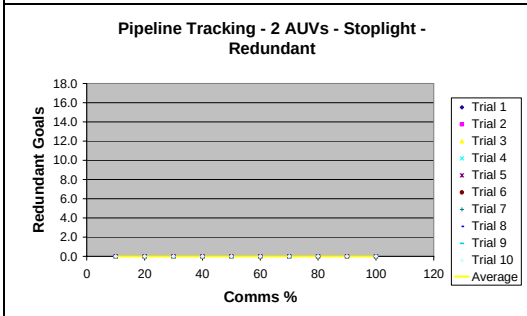
This appendix contains the individual efficiency metric data for the pipeline tracking mission. Data is presented for each of the four systems (DELPHÍS system, DELPHÍS system un-optimised, DELPHÍS system with prediction failure and stoplight system) for groups of 2-4 vehicles.

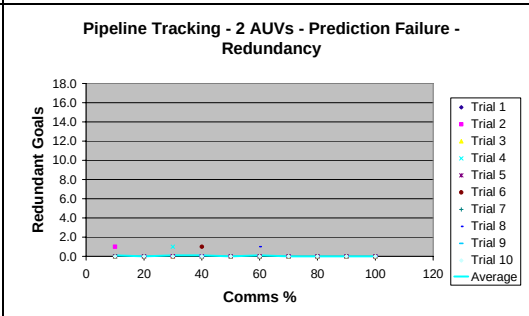
B.1 Pipeline Tracking – Redundancy

2 AUVs

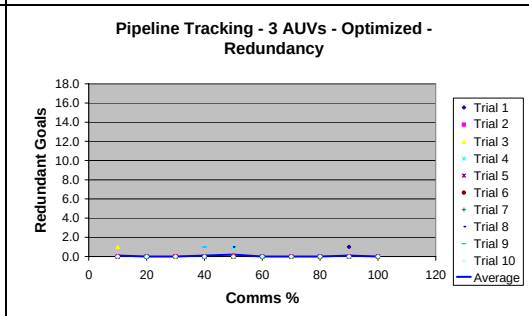


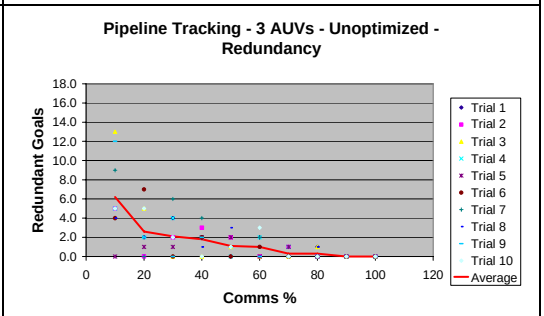


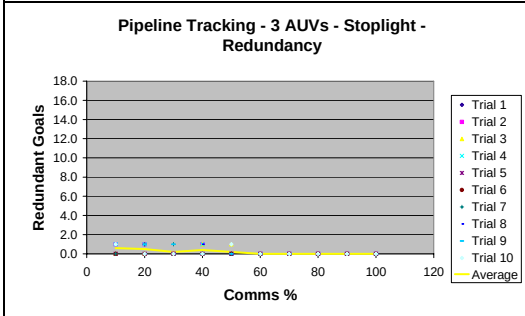


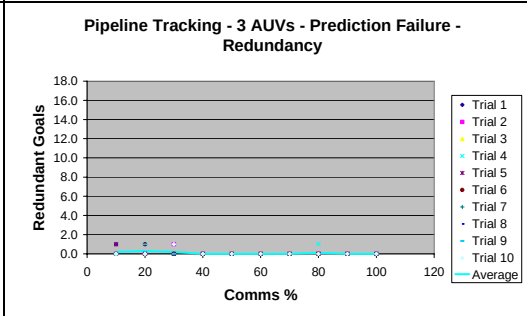


3 AUVs

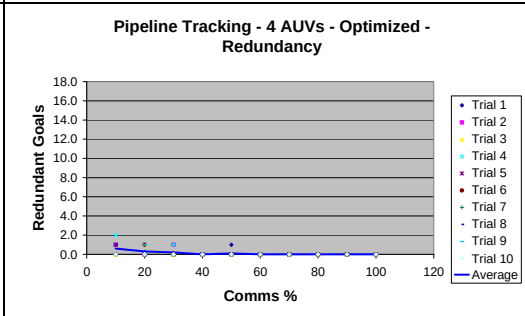


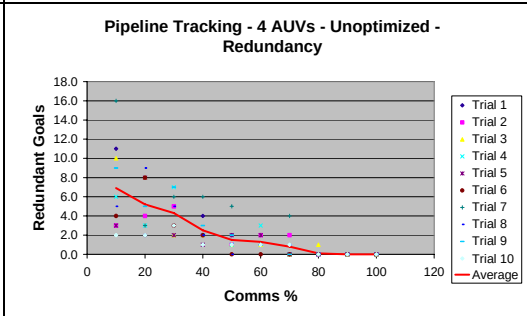


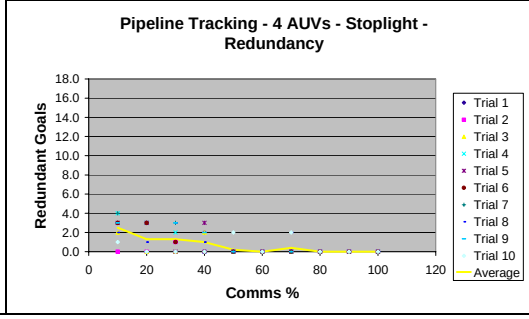


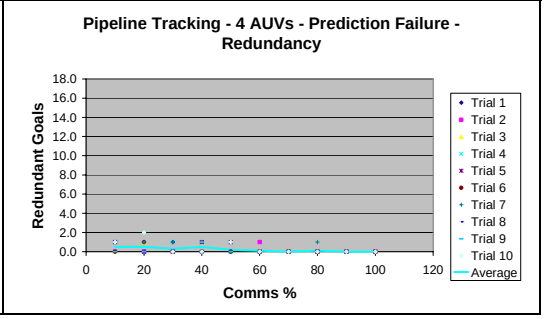


4 AUVs



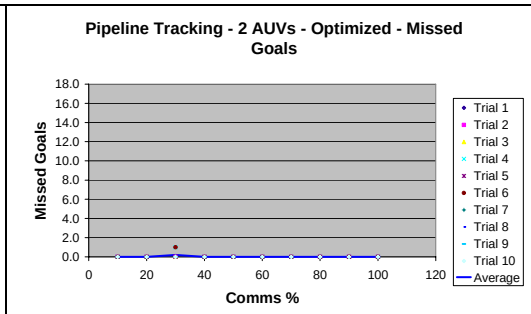


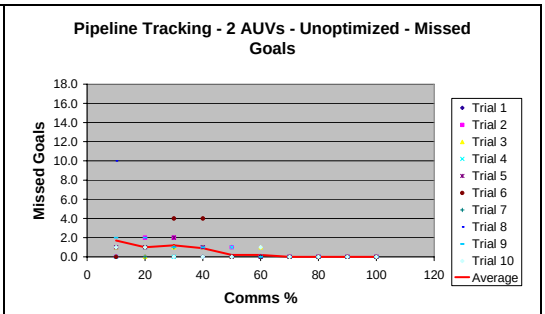


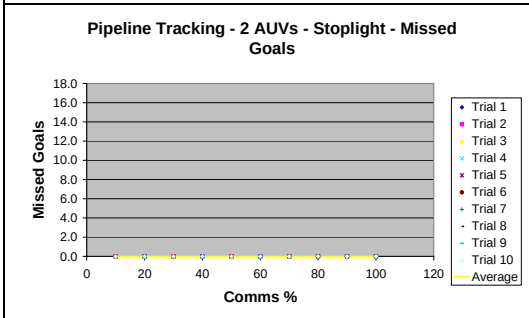


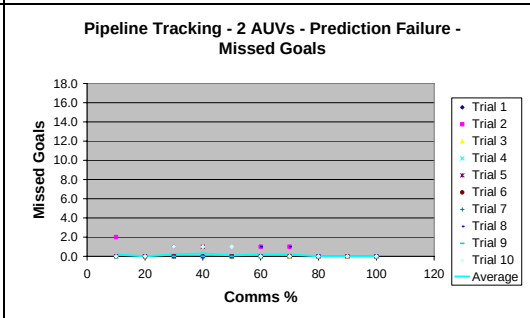
B.2 Pipeline Tracking – Missed Goals

2 AUVs

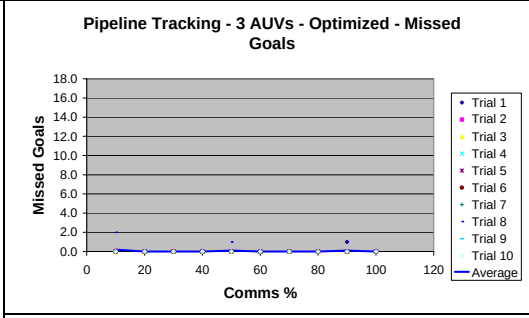


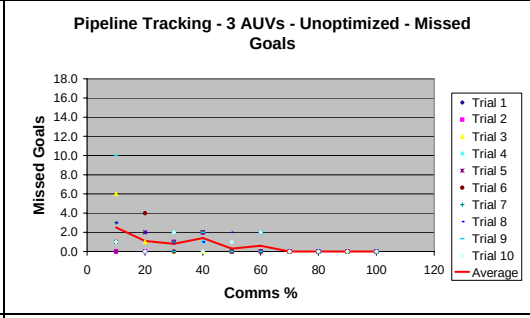


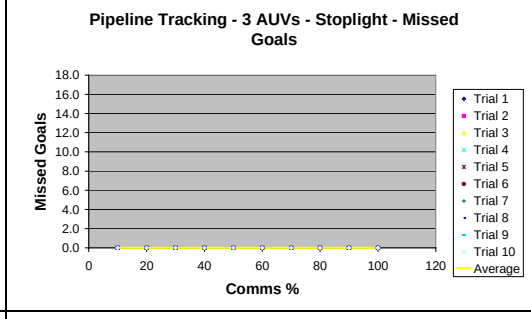


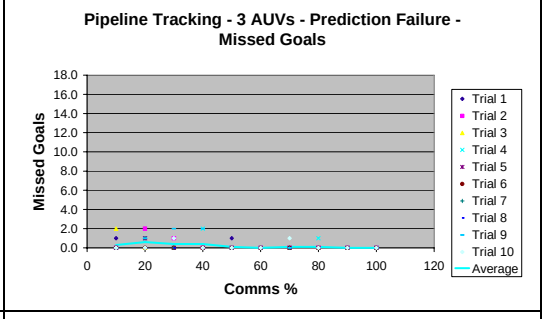


3 AUVs

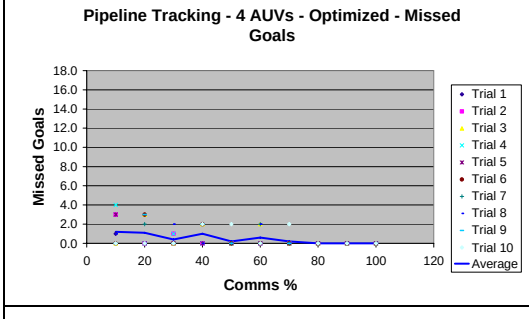


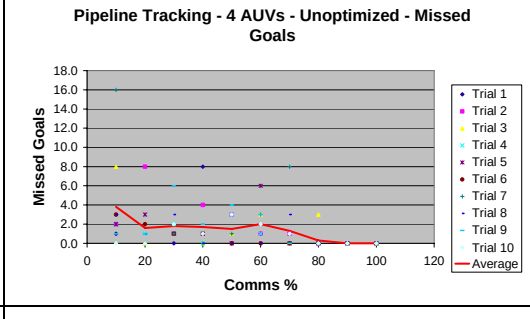


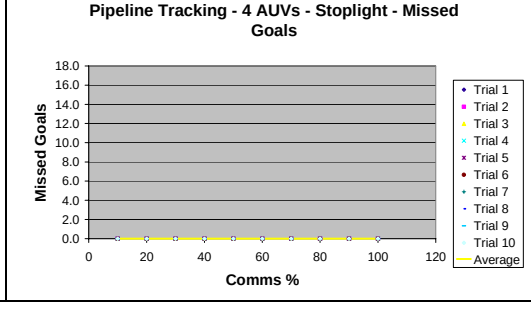


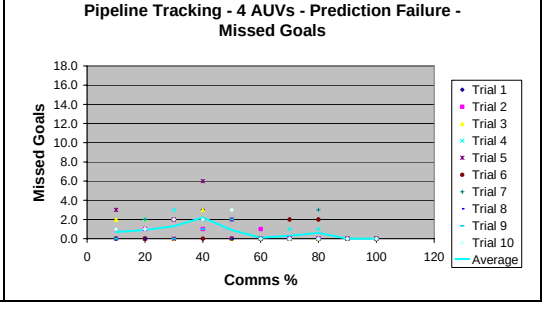


4 AUVs



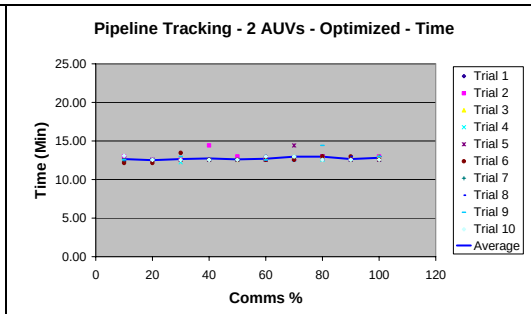


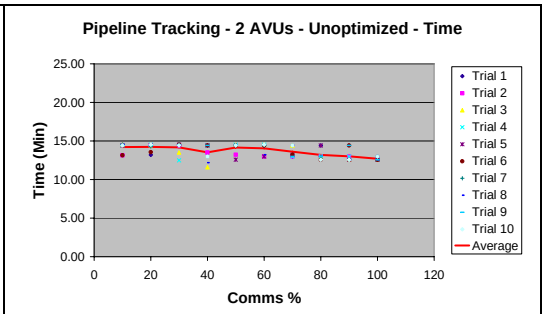


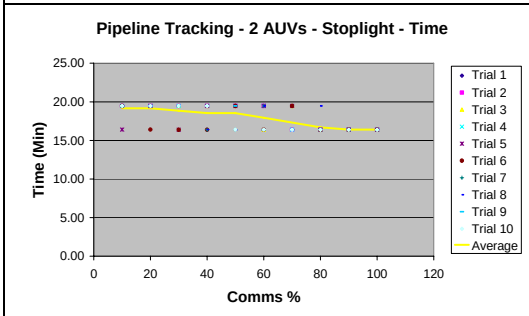


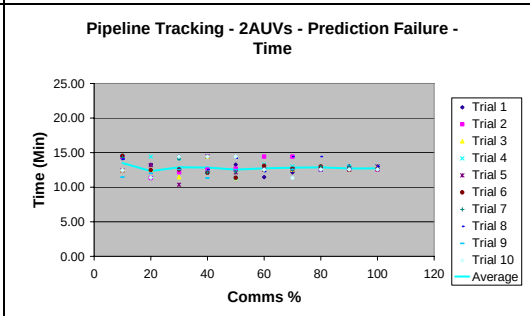
B.3 Pipeline Tracking – Time

2 AUVs

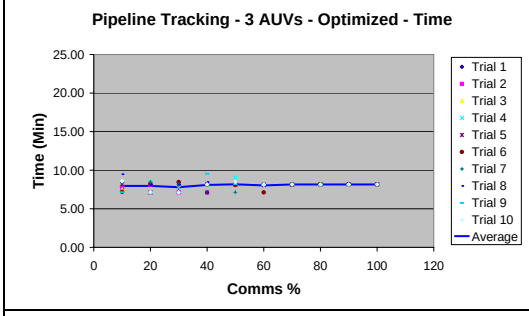


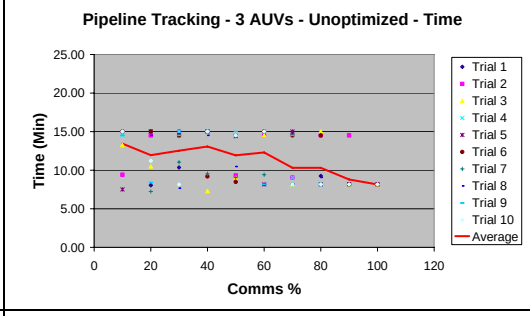


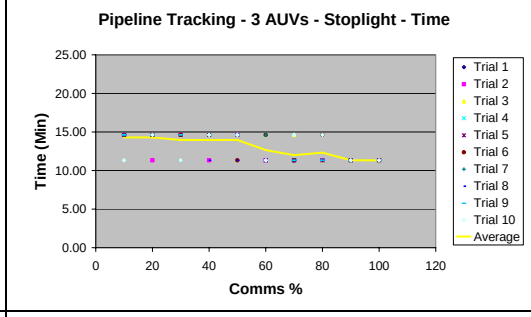


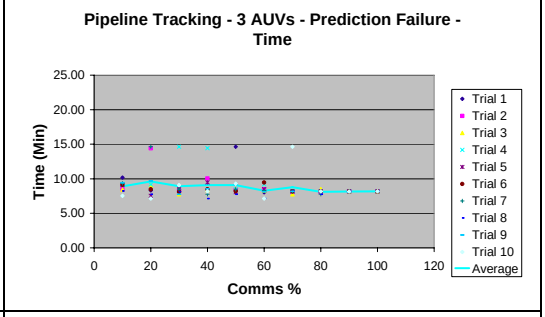


3 AUVs

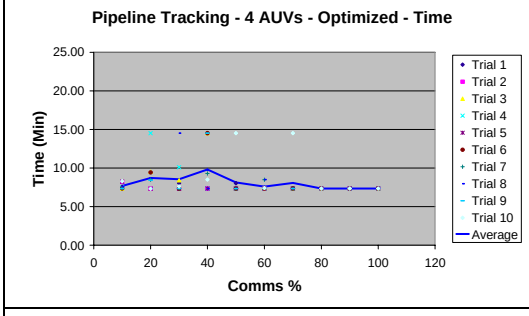


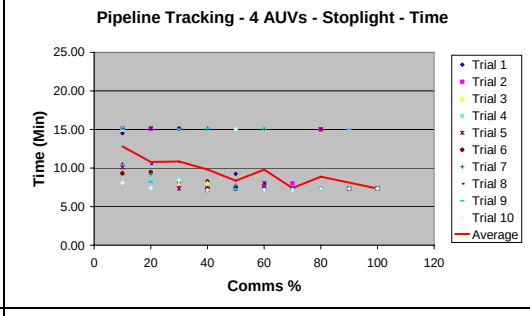


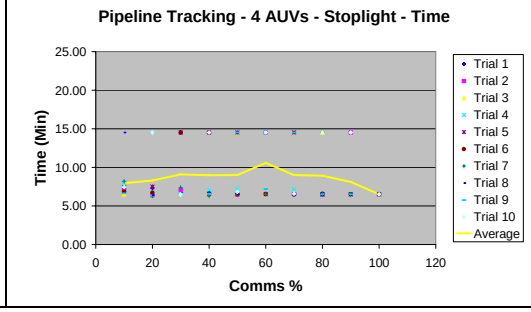


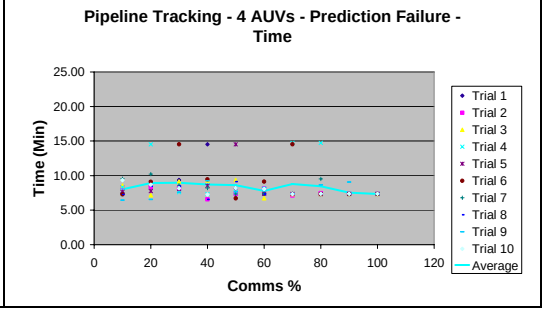


4 AUVs



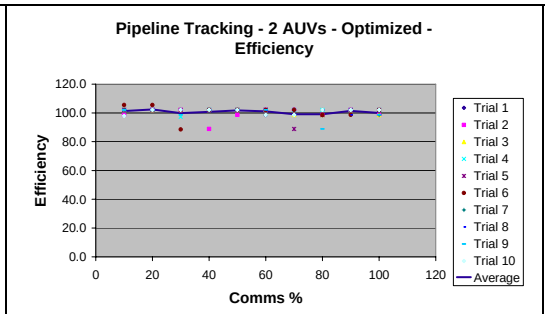


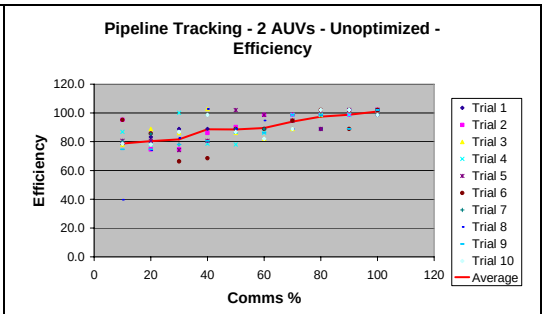


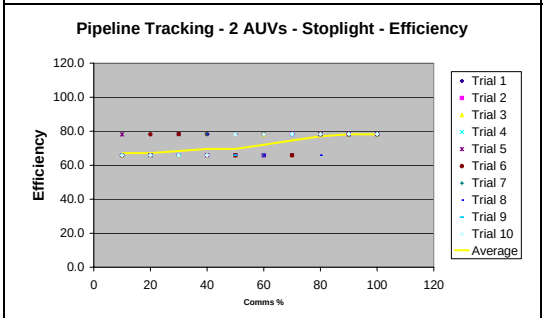


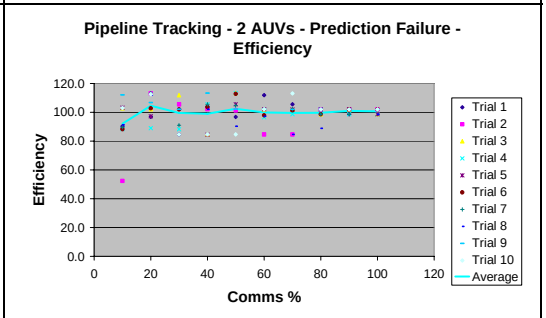
B.4 Pipeline Tracking – Efficiency

2 AUVs

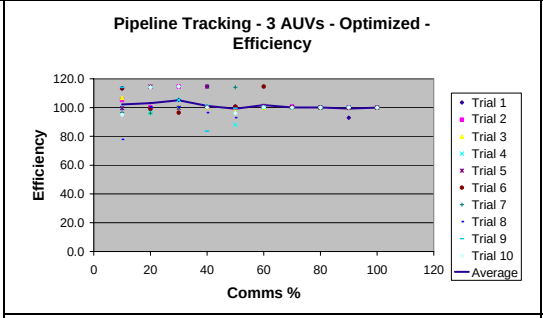


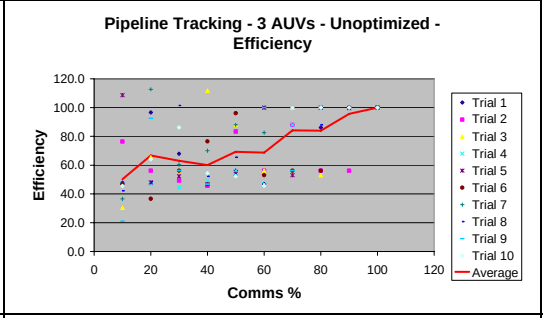


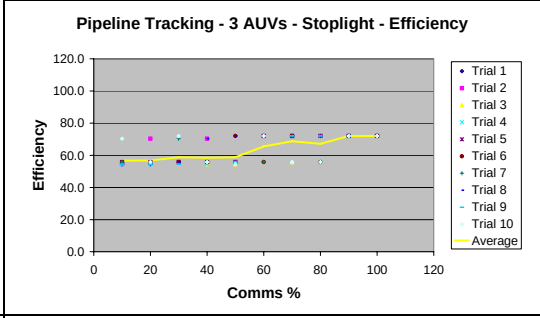


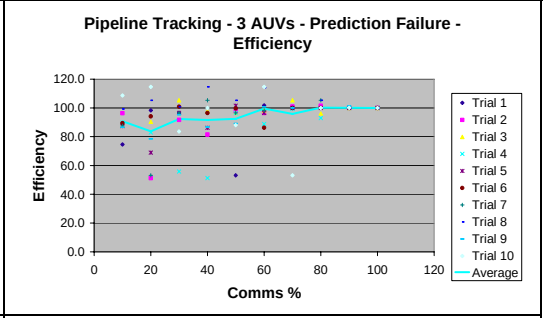


3 AUVs

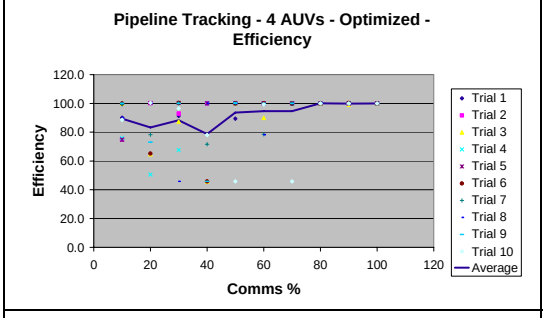


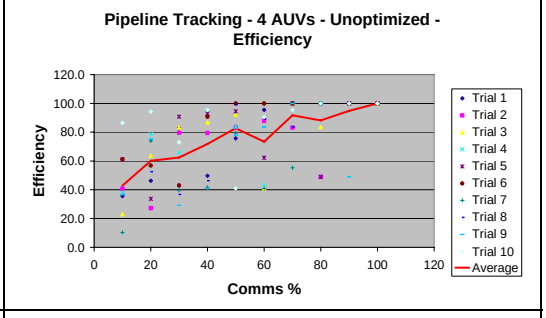


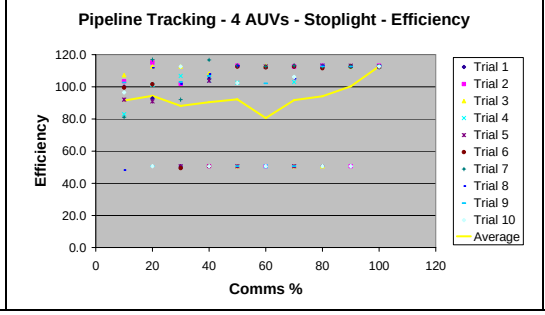


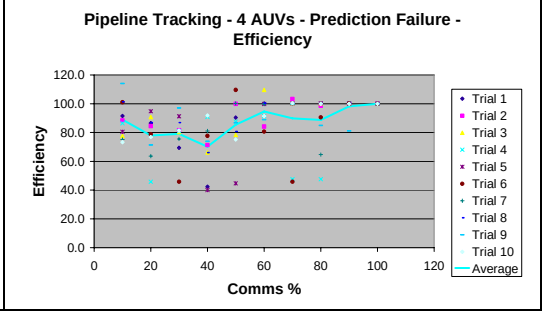


4 AUVs









Appendix C

Threipmuir Reservoir AUV/Simulated AUV Logs

This appendix contains the log data for the Threipmuir Reservoir trials where the DELPHÍS system was used to coordinate a mine countermeasures mission using Nessie III and a simulated AUV. In these logs the Nessie III events are annotated in green and the simulated events in blue.

C.1 Trial 1

Line	Time	Simulated AUV	Nessie
1	[2008-09-18 11:58:31]		Running on inPort 15000 and outPort 15000
2	[2008-09-18 11:58:31]		Importing Message Types.
3	[2008-09-18 11:58:33]		Initializing Message Listener.
4	[2008-09-18 11:58:34]	Running on inPort 11030 and outPort 11030	
5	[2008-09-18 11:58:34]	Importing Message Types.	
6	[2008-09-18 11:58:35]	Initializing Message Listener.	
7	[2008-09-18 11:58:35]		Waiting until control is available.
8	[2008-09-18 11:58:36]	Waiting until control is available.	
9	[2008-09-18 11:58:44]		Control available, starting mission.
10	[2008-09-18 11:58:44]		Initializing Status Beacon.
11	[2008-09-18 11:58:49]	Registered new AUV with id 2	
12	[2008-09-18 11:59:06]		Mission start:
13	[2008-09-18 11:59:06]		executing Waypoint 1
14	[2008-09-18 11:59:36]		Target discovered at x:55.865345 y:-3.31974 z:0.0
15	[2008-09-18 11:59:37]	Control available, starting mission.	
16	[2008-09-18 11:59:37]	Initializing Status Beacon.	
17	[2008-09-18 11:59:40]		finished Waypoint 1
18	[2008-09-18 11:59:40]		executing Waypoint 2
19	[2008-09-18 11:59:43]		Registered new AUV with id 1
20	[2008-09-18 12:00:14]	Mission start:	
21	[2008-09-18 12:00:18]	executing Target[55.86535,-3.3197522,0.0]	
22	[2008-09-18 12:01:29]	finished Target[55.86535,-3.3197522,0.0]	
23	[2008-09-18 12:01:30]	AUV 2 is in position	
24	[2008-09-18 12:01:31]	Predicting that AUV 2 will next do goal Waypoint 3	
25	[2008-09-18 12:01:39]		finished Waypoint 2
26	[2008-09-18 12:01:39]		executing Waypoint 3
27	[2008-09-18 12:01:40]	AUV 2 is in position	
28	[2008-09-18 12:01:41]	Predicting that AUV 2 will next do goal Waypoint 4	
29	[2008-09-18 12:01:54]		finished Waypoint 3
30	[2008-09-18 12:01:54]		executing Waypoint 4
31	[2008-09-18 12:02:06]	AUV 2 is in position	
32	[2008-09-18 12:02:07]	Predicting that AUV 2 will next do goal Waypoint 4	
33	[2008-09-18 12:03:23]		finished Waypoint 4
34	[2008-09-18 12:03:23]		executing Waypoint 5
35	[2008-09-18 12:03:39]		AUV 1 is in position
36	[2008-09-18 12:03:39]		finished Waypoint 5
37	[2008-09-18 12:03:39]		executing Waypoint 6
38	[2008-09-18 12:03:40]		Predicting that AUV 1 will wait until another goal is available
39	[2008-09-18 12:04:08]	AUV 2 is in position	
40	[2008-09-18 12:04:09]	Predicting that AUV 2 will next do goal Waypoint 5	
41	[2008-09-18 12:04:29]	AUV 2 is in position	
42	[2008-09-18 12:04:30]	Predicting that AUV 2 will next do goal Waypoint 6	
43	[2008-09-18 12:04:33]		Target discovered at x:55.865246 y:-3.319626 z:0.0
44	[2008-09-18 12:04:34]		AUV 1 is in position
45	[2008-09-18 12:04:35]		Predicting that AUV 1 will next do goal Target[55.865246,-3.319626,0.0]
46	[2008-09-18 12:05:28]		AUV 1 is in position
47	[2008-09-18 12:05:29]		Predicting that AUV 1 will wait until another goal is available
48	[2008-09-18 12:05:42]		finished Waypoint 6
49	[2008-09-18 12:05:42]		executing Waypoint 7
50	[2008-09-18 12:05:58]		finished Waypoint 7
51	[2008-09-18 12:05:58]		executing Waypoint 8
52	[2008-09-18 12:07:28]		finished Waypoint 8
53	[2008-09-18 12:07:28]		executing Target[55.865246,-3.319626,0.0]
54	[2008-09-18 12:07:46]	AUV 2 is in position	
55	[2008-09-18 12:07:47]	Predicting that AUV 2 will wait until another goal is available	
56	[2008-09-18 12:07:47]	Mission Accomplished. It's Miller Time!!	
57	[2008-09-18 12:07:47]	Time: 7:33	
58	[2008-09-18 12:08:41]		finished Target[55.865246,-3.319626,0.0]
59	[2008-09-18 12:08:41]		Mission Accomplished. It's Miller Time!!
60	[2008-09-18 12:08:41]		Time: 9:32

C.2 Trial 2

Line	Time	Simulated Vehicle	Nessie
1	[2008-09-18 12:13:34]		Running on inPort 15000 and outPort 15000
2	[2008-09-18 12:13:34]		Importing Message Types.
3	[2008-09-18 12:13:37]		Initializing Message Listener.
4	[2008-09-18 12:13:38]		Waiting until control is available.
5	[2008-09-18 12:14:08]	Running on inPort 11030 and outPort 11030	
6	[2008-09-18 12:14:08]	Importing Message Types.	
7	[2008-09-18 12:14:09]	Initializing Message Listener.	
8	[2008-09-18 12:14:10]	Waiting until control is available.	
9	[2008-09-18 12:14:16]		Control available, starting mission.
10	[2008-09-18 12:14:16]		Initializing Status Beacon.
11	[2008-09-18 12:14:22]	Registered new AUV with id 2	
12	[2008-09-18 12:14:39]		Mission start:
13	[2008-09-18 12:14:39]		executing Waypoint 1
14	[2008-09-18 12:14:58]	Control available, starting mission.	
15	[2008-09-18 12:14:58]	Initializing Status Beacon.	
16	[2008-09-18 12:15:04]		Registered new AUV with id 1
17	[2008-09-18 12:15:14]		finished Waypoint 1
18	[2008-09-18 12:15:14]		executing Waypoint 2
19	[2008-09-18 12:15:35]	Mission start:	
20	[2008-09-18 12:15:38]	executing Waypoint 4	
21	[2008-09-18 12:16:59]	finished Waypoint 4	
22	[2008-09-18 12:16:59]	executing Waypoint 3	
23	[2008-09-18 12:17:13]		finished Waypoint 2
24	[2008-09-18 12:17:13]		executing Waypoint 6
25	[2008-09-18 12:17:45]		finished Waypoint 6
26	[2008-09-18 12:17:45]		executing Waypoint 5
27	[2008-09-18 12:18:15]		AUV 1 is in position
28	[2008-09-18 12:18:16]		Predicting that AUV 1 will next do goal Waypoint 3
29	[2008-09-18 12:19:02]	finished Waypoint 3	
30	[2008-09-18 12:19:02]	executing Waypoint 7	
31	[2008-09-18 12:19:16]		finished Waypoint 5
32	[2008-09-18 12:19:16]		executing Waypoint 8
33	[2008-09-18 12:19:23]	Oh no, I'm in an execution lock with AUV:2	
34	[2008-09-18 12:19:23]	...aborting current goal and selecting another	
35	[2008-09-18 12:19:23]	stopped Waypoint 7	
36	[2008-09-18 12:19:23]	Re-evaluating available goals.	
37	[2008-09-18 12:19:32]		AUV 1 is in position
38	[2008-09-18 12:19:33]		Predicting that AUV 1 will wait until another goal is available
39	[2008-09-18 12:19:33]		finished Waypoint 8
40	[2008-09-18 12:19:33]		executing Waypoint 7
41	[2008-09-18 12:19:45]	AUV 2 is in position	
42	[2008-09-18 12:19:46]	Predicting that AUV 2 will next do goal Waypoint 7	
43	[2008-09-18 12:21:31]		finished Waypoint 7
44	[2008-09-18 12:21:32]		Mission Accomplished. It's Miller Time!!
45	[2008-09-18 12:21:32]		Time: 6:51
46	[2008-09-18 12:22:08]	AUV 2 is in position	
47	[2008-09-18 12:22:09]	Mission Accomplished. It's Miller Time!!	
48	[2008-09-18 12:22:09]	Time: 6:34	

C.3 Trial 3

Line	Time	Simulated AUV	Nessie
1	[2008-09-18 12:29:16]	Running on inPort 11030 and outPort 11030	
2	[2008-09-18 12:29:16]	Importing Message Types.	
3	[2008-09-18 12:29:16]	Initializing Message Listener.	
4	[2008-09-18 12:29:17]	Waiting until control is available.	
5	[2008-09-18 12:29:35]		Running on inPort 15000 and outPort 15000
6	[2008-09-18 12:29:35]		Importing Message Types.
7	[2008-09-18 12:29:38]		Initializing Message Listener.
8	[2008-09-18 12:29:39]		Waiting until control is available.
9	[2008-09-18 12:29:39]		Control available, starting mission.
10	[2008-09-18 12:29:39]		Initializing Status Beacon.
11	[2008-09-18 12:29:45]	Registered new AUV with id 2	
12	[2008-09-18 12:30:02]		Mission start:
13	[2008-09-18 12:30:02]		executing Waypoint 1
14	[2008-09-18 12:30:32]		Target discovered at x:55.865345 y:-3.31974 z:0.0 (Mine 1 - Nessie)
15	[2008-09-18 12:30:34]	Control available, starting mission.	
16	[2008-09-18 12:30:34]	Initializing Status Beacon.	
17	[2008-09-18 12:30:36]		finished Waypoint 1
18	[2008-09-18 12:30:36]		executing Waypoint 2
19	[2008-09-18 12:30:40]		Registered new AUV with id 1
20	[2008-09-18 12:31:11]	Mission start:	
21	[2008-09-18 12:31:11]	executing Waypoint 4	
22	[2008-09-18 12:32:23]	finished Waypoint 4	
23	[2008-09-18 12:32:23]	executing Waypoint 3	
24	[2008-09-18 12:32:30]	Target discovered at x:55.865345 y:-3.31974 z:0.0 (Mine 1 - Sim)	
25	[2008-09-18 12:32:37]		finished Waypoint 2
26	[2008-09-18 12:32:37]		executing Target[55.865345,-3.31974,0.0] (Mine 1 - Nessie)
27	[2008-09-18 12:33:14]	Target discovered at x:55.865246 y:-3.319626 z:0.0 (Mine 2 - Sim)	
28	[2008-09-18 12:33:57]		finished Target[55.865345,-3.31974,0.0] (Mine 1 - Nessie)
29	[2008-09-18 12:33:57]		executing Target[55.86535,-3.3197522,0.0] (Mine 1 - Sim)
30	[2008-09-18 12:34:01]		finished Target[55.86535,-3.3197522,0.0] (Mine 1 - Sim)
31	[2008-09-18 12:34:01]		executing Waypoint 5
32	[2008-09-18 12:34:26]	finished Waypoint 3	
33	[2008-09-18 12:34:26]	executing Target[55.865246,-3.319626,0.0] (Mine 2 - Sim)	
34	[2008-09-18 12:34:27]		finished Waypoint 5
35	[2008-09-18 12:34:27]		executing Waypoint 6
36	[2008-09-18 12:35:20]		Target discovered at x:55.865246 y:-3.319626 z:0.0 (Mine 2 - Nessie)
37	[2008-09-18 12:35:25]		AUV 1 is in position
38	[2008-09-18 12:35:26]		Predicting that AUV 1 will wait until another goal is available
39	[2008-09-18 12:35:27]		AUV 1 is in position
40	[2008-09-18 12:35:28]		Predicting that AUV 1 will next do goal Target[55.865246,-3.319626,0.0] (Mine 2 - Nessie)
41	[2008-09-18 12:35:42]	finished Target[55.865246,-3.319626,0.0] (Mine 2 - Sim)	
42	[2008-09-18 12:35:42]	executing Target[55.865345,-3.31974,0.0] (Mine 1 - Sim)	
43	[2008-09-18 12:36:31]		finished Waypoint 6
44	[2008-09-18 12:36:31]		executing Target[55.865246,-3.319626,0.0] (Mine 2 - Sim)
45	[2008-09-18 12:36:44]	finished Target[55.865345,-3.31974,0.0] (Mine 1 - Sim)	
46	[2008-09-18 12:36:44]	executing Target[55.86535,-3.3197522,0.0] (Mine 1 - Nessie)	
47	[2008-09-18 12:36:47]	finished Target[55.86535,-3.3197522,0.0] (Mine 1 - Nessie)	
48	[2008-09-18 12:36:47]	executing Waypoint 5	
49	[2008-09-18 12:37:17]		finished Target[55.865246,-3.319626,0.0] (Mine 2 - Sim)
50	[2008-09-18 12:37:17]		executing Waypoint 7
51	[2008-09-18 12:37:18]	There is a more suitable goal available: Waypoint 8	
52	[2008-09-18 12:37:18]	stopped Waypoint 5	
53	[2008-09-18 12:37:18]	Re-evaluating available goals.	
54	[2008-09-18 12:37:18]	executing Waypoint 8	
55	[2008-09-18 12:37:48]		AUV 1 is in position
56	[2008-09-18 12:37:49]		Predicting that AUV 1 will wait until another goal is available
57	[2008-09-18 12:37:57]	finished Waypoint 8	
58	[2008-09-18 12:37:57]	executing Waypoint 7	
59	[2008-09-18 12:38:03]	Oh no! AUV [2] is going for the same goal as me!	
60	[2008-09-18 12:38:03]	but I've already started the execution lock so I'm going to finish it	
61	[2008-09-18 12:38:03]	continuing Waypoint 7	
62	[2008-09-18 12:38:19]		finished Waypoint 7
63	[2008-09-18 12:38:19]		executing Waypoint 8
64	[2008-09-18 12:38:33]	Oh no, I'm in an execution lock with AUV:2	
65	[2008-09-18 12:38:33]	...but I'm in better position so I'm keeping it	
66	[2008-09-18 12:38:36]		AUV 1 is in position
67	[2008-09-18 12:38:37]		Predicting that AUV 1 will next do goal Waypoint 5
68	[2008-09-18 12:38:38]		Oh no! AUV [1] is going for the same goal as me!
69	[2008-09-18 12:38:38]		other AUV is already in the execution lock... aborting current goal and selecting another
70	[2008-09-18 12:38:39]		stopped Waypoint 8
71	[2008-09-18 12:38:39]		Re-evaluating available goals.
72	[2008-09-18 12:40:00]	finished Waypoint 7	
73	[2008-09-18 12:40:00]	Mission Accomplished. It's Miller Time!!	
74	[2008-09-18 12:40:00]	Time: 8:48	
75	[2008-09-18 12:40:11]		Mission Accomplished. It's Miller Time!!
76	[2008-09-18 12:40:11]		Time: 10:06

Appendix D

Loch Earn Multi-AUV Logs

This appendix contains the log data for the Loch Earn trials where the DELPHÍS system was used to coordinate a mine countermeasures mission using the REMUS and Nessie III AUVs. In these logs the REMUS events are annotated in blue and the Nessie III events in blue.

D.1 Trial 1

Line	Time	REMUS	Nessie
1	[2008-10-02 10:55:00]	Running on inPort 11030 and outPort 11030	
2	[2008-10-02 10:55:00]	Importing Message Types.	
3	[2008-10-02 10:55:07]	Initializing Message Listener.	
4	[2008-10-02 10:55:08]	Waiting until control is available.	
5	[2008-10-02 10:55:36]		Running on inPort 15000 and outPort 15000
6	[2008-10-02 10:55:36]		Importing Message Types.
7	[2008-10-02 10:55:38]		Initializing Message Listener.
8	[2008-10-02 10:55:40]		Waiting until control is available.
9	[2008-10-02 11:06:35]	Control available, starting mission.	
10	[2008-10-02 11:06:35]	Initializing Status Beacon.	
11	[2008-10-02 11:06:37]	Mission start:	
12	[2008-10-02 11:06:37]	executing Waypoint 1	
13	[2008-10-02 11:06:55]	finished Waypoint 1	
14	[2008-10-02 11:06:55]	executing Waypoint 2	
15	[2008-10-02 11:07:56]		Got a status update from: AUV 1
16	[2008-10-02 11:07:56]		Registered new AUV with id 1
17	[2008-10-02 11:08:33]		Control available, starting mission.
18	[2008-10-02 11:08:33]		Initializing Status Beacon.
19	[2008-10-02 11:08:35]	finished Waypoint 2	
20	[2008-10-02 11:08:35]	executing Waypoint 3	
21	[2008-10-02 11:08:43]	Got a status update from: AUV 2	
22	[2008-10-02 11:08:43]	Registered new AUV with id 2	
23	[2008-10-02 11:08:47]	finished Waypoint 3	
24	[2008-10-02 11:08:47]	executing Waypoint 4	
25	[2008-10-02 11:09:03]		Mission start:
26	[2008-10-02 11:09:19]	Target discovered at lat:56.38314 lon:-4.273212	
27	[2008-10-02 11:09:43]	Got a status update from: AUV 2	
28	[2008-10-02 11:10:26]	finished Waypoint 4	
29	[2008-10-02 11:10:26]	executing Waypoint 5	
30	[2008-10-02 11:10:26]		Got a status update from: AUV 1
31	[2008-10-02 11:10:26]		Got a mine coordinate (id:100) from AUV 1
32	[2008-10-02 11:10:27]		executing Target[56.383144,-4.273196,5.0]
33	[2008-10-02 11:10:34]		finished Target[56.383144,-4.273196,5.0]
34	[2008-10-02 11:11:01]	finished Waypoint 5	
35	[2008-10-02 11:11:01]	executing Waypoint 6	
36	[2008-10-02 11:11:03]	Got a status update from: AUV 2	
37	[2008-10-02 11:11:11]		Got a status update from: AUV 1
38	[2008-10-02 11:11:11]		Got a mine coordinate (id:100) from AUV 1
39	[2008-10-02 11:11:46]	Got a status update from: AUV 2	
40	[2008-10-02 11:12:11]		Got a status update from: AUV 1
41	[2008-10-02 11:12:11]		Got a mine coordinate (id:100) from AUV 1
42	[2008-10-02 11:12:26]		Got a status update from: AUV 1
43	[2008-10-02 11:12:26]		Got a mine coordinate (id:100) from AUV 1
44	[2008-10-02 11:12:38]	finished Waypoint 6	
45	[2008-10-02 11:12:38]	executing Waypoint 7	
46	[2008-10-02 11:12:45]	Got a status update from: AUV 2	
47	[2008-10-02 11:12:50]	finished Waypoint 7	
48	[2008-10-02 11:12:50]	executing Waypoint 8	
49	[2008-10-02 11:13:03]	Got a status update from: AUV 2	
50	[2008-10-02 11:13:26]		Got a status update from: AUV 1
51	[2008-10-02 11:13:26]		Got a mine coordinate (id:100) from AUV 1
52	[2008-10-02 11:13:41]		Got a status update from: AUV 1
53	[2008-10-02 11:13:41]		Got a mine coordinate (id:100) from AUV 1
54	[2008-10-02 11:13:45]	Got a status update from: AUV 2	
55	[2008-10-02 11:13:55]	Target discovered at lat:56.382256 lon:-4.2720876	
56	[2008-10-02 11:14:04]	Got a status update from: AUV 2	
57	[2008-10-02 11:14:24]	Got a status update from: AUV 2	
58	[2008-10-02 11:14:26]		Got a status update from: AUV 1
59	[2008-10-02 11:14:26]		Got a mine coordinate (id:100) from AUV 1
60	[2008-10-02 11:14:29]	finished Waypoint 8	
61	[2008-10-02 11:14:29]	executing Target[56.382256,-4.2720876,5.0]	
62	[2008-10-02 11:14:41]		Got a status update from: AUV 1
63	[2008-10-02 11:14:41]		Got a mine coordinate (id:100) from AUV 1
64	[2008-10-02 11:14:45]	Got a status update from: AUV 2	
65	[2008-10-02 11:14:57]		Got a status update from: AUV 1
66	[2008-10-02 11:14:57]		Got a mine coordinate (id:100) from AUV 1
67	[2008-10-02 11:15:04]	Got a status update from: AUV 2	
68	[2008-10-02 11:15:06]	finished Target[56.382256,-4.2720876,5.0]	
69	[2008-10-02 11:15:06]	Mission Accomplished. It's Miller Time!!	
70	[2008-10-02 11:15:06]	Time: 8:28	
71	[2008-10-02 11:15:57]		Got a status update from: AUV 1
72	[2008-10-02 11:15:57]		Got a mine coordinate (id:100) from AUV 1
73	[2008-10-02 11:15:57]		Got a mine coordinate (id:101) from AUV 1
74	[2008-10-02 11:16:34]		Predicting that AUV 1 has completed Target[56.382256,-4.272079,5.0]
75	[2008-10-02 11:16:34]		Mission Accomplished. It's Miller Time!!
76	[2008-10-02 11:16:34]		Time: 8:08

D.2 Trial 2

Line	Time	REMUS	Nessie
1	[2008-10-02 11:42:22]	Running on inPort 11030 and outPort 11030	
2	[2008-10-02 11:42:22]	Importing Message Types.	
3	[2008-10-02 11:42:28]	Initializing Message Listener.	
4	[2008-10-02 11:42:29]	Waiting until control is available.	
5	[2008-10-02 11:45:54]		Running on inPort 15000 and outPort 15000
6	[2008-10-02 11:45:54]		Importing Message Types.
7	[2008-10-02 11:45:56]		Initializing Message Listener.
8	[2008-10-02 11:45:58]		Waiting until control is available.
9	[2008-10-02 11:51:25]	Control available, starting mission.	
10	[2008-10-02 11:51:25]	Initializing Status Beacon.	
11	[2008-10-02 11:51:26]	Mission start:	
12	[2008-10-02 11:51:26]	executing Waypoint 1	
13	[2008-10-02 11:51:45]		Got a status update from: AUV 1
14	[2008-10-02 11:51:45]		Registered new AUV with id 1
15	[2008-10-02 11:51:46]	finished Waypoint 1	
16	[2008-10-02 11:51:46]	executing Waypoint 2	
17	[2008-10-02 11:52:12]		Control available, starting mission.
18	[2008-10-02 11:52:12]		Initializing Status Beacon.
19	[2008-10-02 11:52:22]	Got a status update from: AUV 2	
20	[2008-10-02 11:52:22]	Registered new AUV with id 2	
21	[2008-10-02 11:52:39]	Got a status update from: AUV 2	
22	[2008-10-02 11:52:42]		Mission start:
23	[2008-10-02 11:52:59]	Got a status update from: AUV 2	
24	[2008-10-02 11:53:04]		Got a status update from: AUV 1
25	[2008-10-02 11:53:15]		Got a status update from: AUV 1
26	[2008-10-02 11:53:22]	Got a status update from: AUV 2	
27	[2008-10-02 11:53:26]	finished Waypoint 2	
28	[2008-10-02 11:53:26]	executing Waypoint 3	
29	[2008-10-02 11:53:39]	finished Waypoint 3	
30	[2008-10-02 11:53:39]	executing Waypoint 4	
31	[2008-10-02 11:53:39]	Got a status update from: AUV 2	
32	[2008-10-02 11:53:59]	Got a status update from: AUV 2	
33	[2008-10-02 11:54:05]		Got a status update from: AUV 1
34	[2008-10-02 11:54:10]	Target discovered at lat:56.38314 lon:-4.273212	
35	[2008-10-02 11:54:15]		Got a status update from: AUV 1
36	[2008-10-02 11:54:39]	Got a status update from: AUV 2	
37	[2008-10-02 11:54:45]		Got a status update from: AUV 1
38	[2008-10-02 11:54:59]	Got a status update from: AUV 2	
39	[2008-10-02 11:55:04]		Got a status update from: AUV 1
40	[2008-10-02 11:55:15]		Got a status update from: AUV 1
41	[2008-10-02 11:55:18]	finished Waypoint 4	
42	[2008-10-02 11:55:18]	executing Waypoint 5	
43	[2008-10-02 11:55:22]	Got a status update from: AUV 2	
44	[2008-10-02 11:55:39]	Got a status update from: AUV 2	
45	[2008-10-02 11:55:45]		Got a status update from: AUV 1
46	[2008-10-02 11:55:45]		Got a mine coordinate (id:100) from AUV 1
47	[2008-10-02 11:55:46]		executing Target[56.383144,-4.273196,5.0]
48	[2008-10-02 11:55:53]	finished Waypoint 5	
49	[2008-10-02 11:55:53]	executing Waypoint 6	
50	[2008-10-02 11:55:59]	Got a status update from: AUV 2	
51	[2008-10-02 11:56:04]		Got a status update from: AUV 1
52	[2008-10-02 11:56:04]		Got a mine coordinate (id:100) from AUV 1
53	[2008-10-02 11:56:05]	Target discovered at lat:56.382496 lon:-4.271436	
54	[2008-10-02 11:56:15]		Got a status update from: AUV 1
55	[2008-10-02 11:56:15]		Got a mine coordinate (id:100) from AUV 1
56	[2008-10-02 11:56:22]	Got a status update from: AUV 2	
57	[2008-10-02 11:56:39]	Got a status update from: AUV 2	
58	[2008-10-02 11:56:45]		Got a status update from: AUV 1
59	[2008-10-02 11:56:45]		Got a mine coordinate (id:100) from AUV 1
60	[2008-10-02 11:57:15]		Got a status update from: AUV 1
61	[2008-10-02 11:57:15]		Got a mine coordinate (id:100) from AUV 1
62	[2008-10-02 11:57:28]		finished Target[56.383144,-4.273196,5.0]
63	[2008-10-02 11:57:31]	finished Waypoint 6	
64	[2008-10-02 11:57:31]	executing Waypoint 7	
65	[2008-10-02 11:57:39]	Got a status update from: AUV 2	
66	[2008-10-02 11:57:43]	finished Waypoint 7	
67	[2008-10-02 11:57:43]	executing Waypoint 8	
68	[2008-10-02 11:57:45]		Got a status update from: AUV 1
69	[2008-10-02 11:57:45]		Got a mine coordinate (id:100) from AUV 1
70	[2008-10-02 11:57:45]		Got a mine coordinate (id:101) from AUV 1
71	[2008-10-02 11:57:46]		executing Target[56.3825,-4.2714314,5.0]
72	[2008-10-02 11:57:59]	Got a status update from: AUV 2	
73	[2008-10-02 11:58:23]	Got a status update from: AUV 2	
74	[2008-10-02 11:58:30]		Got a status update from: AUV 1
75	[2008-10-02 11:58:30]		Got a mine coordinate (id:100) from AUV 1
76	[2008-10-02 11:58:30]		Got a mine coordinate (id:101) from AUV 1
77	[2008-10-02 11:58:39]	Got a status update from: AUV 2	
78	[2008-10-02 11:58:45]		Got a status update from: AUV 1
79	[2008-10-02 11:58:45]		Got a mine coordinate (id:100) from AUV 1
80	[2008-10-02 11:58:45]		Got a mine coordinate (id:101) from AUV 1
81	[2008-10-02 11:58:59]	Got a status update from: AUV 2	
82	[2008-10-02 11:59:15]		Got a status update from: AUV 1
83	[2008-10-02 11:59:15]		Got a mine coordinate (id:100) from AUV 1
84	[2008-10-02 11:59:15]		Got a mine coordinate (id:101) from AUV 1
85	[2008-10-02 11:59:21]	finished Waypoint 8	
86	[2008-10-02 11:59:22]	Got a status update from: AUV 2	
87	[2008-10-02 11:59:34]		Got a status update from: AUV 1
88	[2008-10-02 11:59:34]		Got a mine coordinate (id:100) from AUV 1
89	[2008-10-02 11:59:34]		Got a mine coordinate (id:101) from AUV 1
90	[2008-10-02 11:59:41]	Got a status update from: AUV 2	
91	[2008-10-02 11:59:59]	Got a status update from: AUV 2	
92	[2008-10-02 12:00:15]		Got a status update from: AUV 1
93	[2008-10-02 12:00:15]		Got a mine coordinate (id:100) from AUV 1
94	[2008-10-02 12:00:15]		Got a mine coordinate (id:101) from AUV 1

95	[2008-10-02 12:00:22]	Got a status update from: AUV 2	
96	[2008-10-02 12:00:30]		Got a status update from: AUV 1
97	[2008-10-02 12:00:30]		Got a mine coordinate (id:100) from AUV 1
98	[2008-10-02 12:00:30]		Got a mine coordinate (id:101) from AUV 1
99	[2008-10-02 12:00:39]	Got a status update from: AUV 2	
100	[2008-10-02 12:00:45]		Got a status update from: AUV 1
101	[2008-10-02 12:00:45]		Got a mine coordinate (id:100) from AUV 1
102	[2008-10-02 12:00:45]		Got a mine coordinate (id:101) from AUV 1
103	[2008-10-02 12:00:59]	Got a status update from: AUV 2	
104	[2008-10-02 12:01:15]		Got a status update from: AUV 1
105	[2008-10-02 12:01:15]		Got a mine coordinate (id:100) from AUV 1
106	[2008-10-02 12:01:15]		Got a mine coordinate (id:101) from AUV 1
107	[2008-10-02 12:01:22]	Got a status update from: AUV 2	
108	[2008-10-02 12:01:30]		Got a status update from: AUV 1
109	[2008-10-02 12:01:30]		Got a mine coordinate (id:100) from AUV 1
110	[2008-10-02 12:01:30]		Got a mine coordinate (id:101) from AUV 1
111	[2008-10-02 12:01:39]	Got a status update from: AUV 2	
112	[2008-10-02 12:01:59]	Got a status update from: AUV 2	
113	[2008-10-02 12:02:23]	Got a status update from: AUV 2	
114	[2008-10-02 12:02:39]	Got a status update from: AUV 2	
115	[2008-10-02 12:02:45]		Got a status update from: AUV 1
116	[2008-10-02 12:02:45]		Got a mine coordinate (id:100) from AUV 1
117	[2008-10-02 12:02:45]		Got a mine coordinate (id:101) from AUV 1
118	[2008-10-02 12:02:59]	Got a status update from: AUV 2	
119	[2008-10-02 12:03:15]		Got a status update from: AUV 1
120	[2008-10-02 12:03:15]		Got a mine coordinate (id:100) from AUV 1
121	[2008-10-02 12:03:15]		Got a mine coordinate (id:101) from AUV 1
122	[2008-10-02 12:03:22]	Got a status update from: AUV 2	
123	[2008-10-02 12:03:39]	Got a status update from: AUV 2	
124	[2008-10-02 12:03:59]	Got a status update from: AUV 2	
125	[2008-10-02 12:04:20]		finished Target[56.3825,-4.2714314,5.0]
126	[2008-10-02 12:04:20]		Mission Accomplished. It's Miller Time!!
127	[2008-10-02 12:04:20]		Time: 11:35
128	[2008-10-02 12:04:30]	Predicting that AUV 2 has completed Target[56.382496,-4.271436,5.0]	
129	[2008-10-02 12:04:31]	Mission Accomplished. It's Miller Time!!	
130	[2008-10-02 12:04:31]	Time: 13:02	

D.3 Trial 3

Line	Time	REMUS	Nessie
1	[2008-10-02 12:19:00]	Running on inPort 11030 and outPort 11030	
2	[2008-10-02 12:19:00]	Importing Message Types.	
3	[2008-10-02 12:19:05]	Initializing Message Listener.	
4	[2008-10-02 12:19:07]	Waiting until control is available.	
5	[2008-10-02 12:21:28]		Running on inPort 15000 and outPort 15000
6	[2008-10-02 12:21:28]		Importing Message Types.
7	[2008-10-02 12:21:30]		Initializing Message Listener.
8	[2008-10-02 12:21:31]		Waiting until control is available.
9	[2008-10-02 12:26:24]	Control available, starting mission.	
10	[2008-10-02 12:26:24]	Initializing Status Beacon.	
11	[2008-10-02 12:26:26]	Mission start:	
12	[2008-10-02 12:26:26]	executing Waypoint 1	
13	[2008-10-02 12:26:44]	finished Waypoint 1	
14	[2008-10-02 12:26:44]	executing Waypoint 2	
15	[2008-10-02 12:26:46]		Got a status update from: AUV 1
16	[2008-10-02 12:26:46]		Registered new AUV with id 1
17	[2008-10-02 12:27:01]		Control available, starting mission.
18	[2008-10-02 12:27:01]		Initializing Status Beacon.
19	[2008-10-02 12:27:01]		Got a status update from: AUV 1
20	[2008-10-02 12:27:11]	Got a status update from: AUV 2	
21	[2008-10-02 12:27:11]	Registered new AUV with id 2	
22	[2008-10-02 12:27:16]		Got a status update from: AUV 1
23	[2008-10-02 12:27:31]	Got a status update from: AUV 2	
24	[2008-10-02 12:27:31]		Mission start:
25	[2008-10-02 12:27:34]		Got a status update from: AUV 1
26	[2008-10-02 12:28:11]	Got a status update from: AUV 2	
27	[2008-10-02 12:28:16]		Got a status update from: AUV 1
28	[2008-10-02 12:28:25]	finished Waypoint 2	
29	[2008-10-02 12:28:25]	executing Waypoint 3	
30	[2008-10-02 12:28:31]	Got a status update from: AUV 2	
31	[2008-10-02 12:28:37]	finished Waypoint 3	
32	[2008-10-02 12:28:37]	executing Waypoint 4	
33	[2008-10-02 12:29:01]		Got a status update from: AUV 1
34	[2008-10-02 12:29:09]	Target discovered at lat:56.38314 lon:-4.273212	
35	[2008-10-02 12:29:11]	Got a status update from: AUV 2	
36	[2008-10-02 12:29:31]	Got a status update from: AUV 2	
37	[2008-10-02 12:29:34]		Got a status update from: AUV 1
38	[2008-10-02 12:30:01]		Got a status update from: AUV 1
39	[2008-10-02 12:30:11]	Got a status update from: AUV 2	
40	[2008-10-02 12:30:16]	finished Waypoint 4	
41	[2008-10-02 12:30:16]	executing Waypoint 5	
42	[2008-10-02 12:30:31]	Got a status update from: AUV 2	
43	[2008-10-02 12:30:34]		Got a status update from: AUV 1
44	[2008-10-02 12:30:34]		Got a mine coordinate (id:100) from AUV 1
45	[2008-10-02 12:30:34]		executing Target[56.383144,-4.273196,5.0]
46	[2008-10-02 12:30:51]	finished Waypoint 5	
47	[2008-10-02 12:30:51]	executing Waypoint 6	
48	[2008-10-02 12:31:01]		Got a status update from: AUV 1
49	[2008-10-02 12:31:01]		Got a mine coordinate (id:100) from AUV 1
50	[2008-10-02 12:31:02]	Target discovered at lat:56.382496 lon:-4.271436	
51	[2008-10-02 12:31:11]	Got a status update from: AUV 2	
52	[2008-10-02 12:31:31]	Got a status update from: AUV 2	
53	[2008-10-02 12:31:34]		Got a status update from: AUV 1
54	[2008-10-02 12:31:34]		Got a mine coordinate (id:100) from AUV 1
55	[2008-10-02 12:32:15]		Predicting that AUV 1 has completed Waypoint 5
56	[2008-10-02 12:32:16]		Predicting that AUV 1 will next do goal Waypoint 6
57	[2008-10-02 12:32:29]	finished Waypoint 6	
58	[2008-10-02 12:32:29]	executing Waypoint 7	
59	[2008-10-02 12:32:31]	Got a status update from: AUV 2	
60	[2008-10-02 12:32:34]		Got a status update from: AUV 1
61	[2008-10-02 12:32:34]		Got a mine coordinate (id:100) from AUV 1
62	[2008-10-02 12:32:34]		Got a mine coordinate (id:101) from AUV 1
63	[2008-10-02 12:32:41]	finished Waypoint 7	
64	[2008-10-02 12:32:41]	executing Waypoint 8	
65	[2008-10-02 12:33:07]		finished Target[56.383144,-4.273196,5.0]
66	[2008-10-02 12:33:07]		executing Target[56.3825,-4.2714314,5.0]
67	[2008-10-02 12:33:11]	Got a status update from: AUV 2	
68	[2008-10-02 12:33:16]		Predicting that AUV 1 has completed Waypoint 6
69	[2008-10-02 12:33:17]		Predicting that AUV 1 will next do goal Waypoint 7
70	[2008-10-02 12:33:27]		Predicting that AUV 1 has completed Waypoint 7
71	[2008-10-02 12:33:28]		Predicting that AUV 1 will next do goal Waypoint 8
72	[2008-10-02 12:33:31]	Got a status update from: AUV 2	
73	[2008-10-02 12:34:11]	Got a status update from: AUV 2	
74	[2008-10-02 12:34:20]	finished Waypoint 8	
75	[2008-10-02 12:34:31]	Got a status update from: AUV 2	
76	[2008-10-02 12:35:11]	Got a status update from: AUV 2	
77	[2008-10-02 12:35:16]		Got a status update from: AUV 1
78	[2008-10-02 12:35:16]		Got a mine coordinate (id:100) from AUV 1
79	[2008-10-02 12:35:16]		Got a mine coordinate (id:101) from AUV 1
80	[2008-10-02 12:35:31]	Got a status update from: AUV 2	
81	[2008-10-02 12:35:52]		Predicting that AUV 1 has completed Waypoint 8
82	[2008-10-02 12:35:53]		Predicting that AUV 1 will wait until another goal is available
83	[2008-10-02 12:35:55]		Predicting that AUV 1 has completed Waypoint 8
84	[2008-10-02 12:35:56]		Predicting that AUV 1 will wait until another goal is available
85	[2008-10-02 12:35:58]		Predicting that AUV 1 has completed Waypoint 8
86	[2008-10-02 12:35:59]		Predicting that AUV 1 will wait until another goal is available
87	[2008-10-02 12:36:01]		Got a status update from: AUV 1
88	[2008-10-02 12:36:01]		Got a mine coordinate (id:100) from AUV 1
89	[2008-10-02 12:36:01]		Got a mine coordinate (id:101) from AUV 1
90	[2008-10-02 12:36:11]	Got a status update from: AUV 2	
91	[2008-10-02 12:36:31]	Got a status update from: AUV 2	

92	[2008-10-02 12:37:01]		Got a status update from: AUV 1
93	[2008-10-02 12:37:01]		Got a mine coordinate (id:100) from AUV 1
94	[2008-10-02 12:37:01]		Got a mine coordinate (id:101) from AUV 1
95	[2008-10-02 12:37:11]	Got a status update from: AUV 2	
96	[2008-10-02 12:37:31]	Got a status update from: AUV 2	
97	[2008-10-02 12:38:01]		Got a status update from: AUV 1
98	[2008-10-02 12:38:01]		Got a mine coordinate (id:100) from AUV 1
99	[2008-10-02 12:38:01]		Got a mine coordinate (id:101) from AUV 1
100	[2008-10-02 12:38:11]	Got a status update from: AUV 2	
101	[2008-10-02 12:38:31]	Got a status update from: AUV 2	
102	[2008-10-02 12:38:34]		Got a status update from: AUV 1
103	[2008-10-02 12:38:34]		Got a mine coordinate (id:100) from AUV 1
104	[2008-10-02 12:38:34]		Got a mine coordinate (id:101) from AUV 1
105	[2008-10-02 12:39:01]		Got a status update from: AUV 1
106	[2008-10-02 12:39:01]		Got a mine coordinate (id:100) from AUV 1
107	[2008-10-02 12:39:01]		Got a mine coordinate (id:101) from AUV 1
108	[2008-10-02 12:39:11]	Got a status update from: AUV 2	
109	[2008-10-02 12:39:31]	Got a status update from: AUV 2	
110	[2008-10-02 12:39:34]		Got a status update from: AUV 1
111	[2008-10-02 12:39:34]		Got a mine coordinate (id:100) from AUV 1
112	[2008-10-02 12:39:35]		Got a mine coordinate (id:101) from AUV 1
113	[2008-10-02 12:39:44]		finished Target[56.3825,-4.2714314,5.0]
114	[2008-10-02 12:39:44]		Mission Accomplished. It's Miller Time!!
115	[2008-10-02 12:39:44]		Time: 12:09
116	[2008-10-02 12:40:01]	Predicting that AUV 2 has completed Target[56.382496, -4.271436,5.0]	
117	[2008-10-02 12:40:02]	Mission Accomplished. It's Miller Time!!	
118	[2008-10-02 12:40:02]	Time: 13:34	

D.4 Loch Earn Simulation

Line	Time	Simulated REMUS	Simulated Nessie
1	[2008-10-14 13:30:58]	Running on inPort 11030 and outPort 11030	
2	[2008-10-14 13:30:58]	Importing Message Types.	
3	[2008-10-14 13:30:59]	Initializing Message Listener.	
4	[2008-10-14 13:31:00]	Waiting until control is available.	
5	[2008-10-14 13:31:13]		Running on inPort 11032 and outPort 11032
6	[2008-10-14 13:31:13]		Importing Message Types.
7	[2008-10-14 13:31:14]		Initializing Message Listener.
8	[2008-10-14 13:31:15]		Waiting until control is available.
9	[2008-10-14 13:33:32]	Control available, starting mission.	
10	[2008-10-14 13:33:32]	Initializing Status Beacon.	
11	[2008-10-14 13:33:32]		Got a status update from: AUV 1
12	[2008-10-14 13:33:32]		Registered new AUV with id 1
13	[2008-10-14 13:33:33]	Mission start:	
14	[2008-10-14 13:33:33]	executing Waypoint 1	
15	[2008-10-14 13:33:51]	finished Waypoint 1	
16	[2008-10-14 13:33:51]	executing Waypoint 2	
17	[2008-10-14 13:35:31]	finished Waypoint 2	
18	[2008-10-14 13:35:31]	executing Waypoint 3	
19	[2008-10-14 13:35:41]	finished Waypoint 3	
20	[2008-10-14 13:35:41]	executing Waypoint 4	
21	[2008-10-14 13:36:11]	Target discovered at lat:56.38314 lon:-4.273212	
22	[2008-10-14 13:36:11]		Got a status update from: AUV 1
23	[2008-10-14 13:36:14]	Got a status update from: AUV 2	
24	[2008-10-14 13:36:14]	Registered new AUV with id 2	
25	[2008-10-14 13:36:14]		Control available, starting mission.
26	[2008-10-14 13:36:14]		Initializing Status Beacon.
27	[2008-10-14 13:36:43]	Got a status update from: AUV 2	
28	[2008-10-14 13:36:58]		Mission start:
29	[2008-10-14 13:37:12]	Got a status update from: AUV 2	
30	[2008-10-14 13:37:18]	finished Waypoint 4	
31	[2008-10-14 13:37:18]	executing Waypoint 5	
32	[2008-10-14 13:37:41]	Got a status update from: AUV 2	
33	[2008-10-14 13:37:54]	finished Waypoint 5	
34	[2008-10-14 13:37:54]	executing Waypoint 6	
35	[2008-10-14 13:38:04]	Target discovered at lat:56.382496 lon:-4.271436	
36	[2008-10-14 13:38:10]	Got a status update from: AUV 2	
37	[2008-10-14 13:38:39]	Got a status update from: AUV 2	
38	[2008-10-14 13:38:50]		Got a status update from: AUV 1
39	[2008-10-14 13:38:50]		Got a mine coordinate (id:100) from AUV 1
40	[2008-10-14 13:38:50]		Got a mine coordinate (id:101) from AUV 1
41	[2008-10-14 13:38:50]		executing Target 100 [56.383144,-4.273196,5.0]
42	[2008-10-14 13:39:08]	Got a status update from: AUV 2	
43	[2008-10-14 13:39:08]	Got a mine coordinate (id:100) from AUV 2	
44	[2008-10-14 13:39:08]	Got a mine coordinate (id:101) from AUV 2	
45	[2008-10-14 13:39:33]	finished Waypoint 6	
46	[2008-10-14 13:39:33]	executing Waypoint 7	
47	[2008-10-14 13:39:37]	Got a status update from: AUV 2	
48	[2008-10-14 13:39:37]	Got a mine coordinate (id:100) from AUV 2	
49	[2008-10-14 13:39:37]	Got a mine coordinate (id:101) from AUV 2	
50	[2008-10-14 13:39:43]	finished Waypoint 7	
51	[2008-10-14 13:39:43]	executing Waypoint 8	
52	[2008-10-14 13:40:06]	Got a status update from: AUV 2	
53	[2008-10-14 13:40:06]	Got a mine coordinate (id:100) from AUV 2	
54	[2008-10-14 13:40:06]	Got a mine coordinate (id:101) from AUV 2	
55	[2008-10-14 13:40:16]		finished Target 100 [56.383144,-4.273196,5.0]
56	[2008-10-14 13:40:16]		executing Target 101 [56.3825,-4.2714314,5.0]
57	[2008-10-14 13:40:35]	Got a status update from: AUV 2	
58	[2008-10-14 13:40:35]	Got a mine coordinate (id:100) from AUV 2	
59	[2008-10-14 13:40:35]	Got a mine coordinate (id:101) from AUV 2	
60	[2008-10-14 13:41:04]	Got a status update from: AUV 2	
61	[2008-10-14 13:41:04]	Got a mine coordinate (id:100) from AUV 2	
62	[2008-10-14 13:41:04]	Got a mine coordinate (id:101) from AUV 2	
63	[2008-10-14 13:41:20]	finished Waypoint 8	
64	[2008-10-14 13:41:29]		Got a status update from: AUV 1
65	[2008-10-14 13:41:29]		Got a mine coordinate (id:100) from AUV 1
66	[2008-10-14 13:41:29]		Got a mine coordinate (id:101) from AUV 1
67	[2008-10-14 13:41:33]	Got a status update from: AUV 2	
68	[2008-10-14 13:41:33]	Got a mine coordinate (id:100) from AUV 2	
69	[2008-10-14 13:41:33]	Got a mine coordinate (id:101) from AUV 2	
70	[2008-10-14 13:42:02]	Got a status update from: AUV 2	
71	[2008-10-14 13:42:02]	Got a mine coordinate (id:100) from AUV 2	
72	[2008-10-14 13:42:02]	Got a mine coordinate (id:101) from AUV 2	
73	[2008-10-14 13:42:31]	Got a status update from: AUV 2	
74	[2008-10-14 13:42:31]	Got a mine coordinate (id:100) from AUV 2	
75	[2008-10-14 13:42:31]	Got a mine coordinate (id:101) from AUV 2	
76	[2008-10-14 13:43:00]	Got a status update from: AUV 2	
77	[2008-10-14 13:43:00]	Got a mine coordinate (id:100) from AUV 2	
78	[2008-10-14 13:43:00]	Got a mine coordinate (id:101) from AUV 2	
79	[2008-10-14 13:43:29]	Got a status update from: AUV 2	
80	[2008-10-14 13:43:29]	Got a mine coordinate (id:100) from AUV 2	
81	[2008-10-14 13:43:29]	Got a mine coordinate (id:101) from AUV 2	
82	[2008-10-14 13:43:58]	Got a status update from: AUV 2	
83	[2008-10-14 13:43:58]	Got a mine coordinate (id:100) from AUV 2	
84	[2008-10-14 13:43:58]	Got a mine coordinate (id:101) from AUV 2	
85	[2008-10-14 13:44:08]		Got a status update from: AUV 1
86	[2008-10-14 13:44:08]		Got a mine coordinate (id:100) from AUV 1
87	[2008-10-14 13:44:08]		Got a mine coordinate (id:101) from AUV 1
88	[2008-10-14 13:44:27]	Got a status update from: AUV 2	
89	[2008-10-14 13:44:27]	Got a mine coordinate (id:100) from AUV 2	
90	[2008-10-14 13:44:27]	Got a mine coordinate (id:101) from AUV 2	
91	[2008-10-14 13:44:31]		finished Target 101 [56.3825,-4.2714314,5.0]
92	[2008-10-14 13:44:31]		Mission Accomplished. It's Miller Time!!
93	[2008-10-14 13:44:31]		Time: 7:33

94	[2008-10-14 13:45:15]	Predicting that AUV 2 has completed Target 101 [56.382496,-4.271436,5.0]	
95	[2008-10-14 13:45:15]	Mission Accomplished. It's Miller Time!!	
96	[2008-10-14 13:45:15]	Time: 11:42	

References

[1] G. Agha and C. Hewitt. Actors: A conceptual foundation for concurrent object-oriented programming. In B. Shriver and P. Wegner, editors, *Research Directions in Object Oriented Programming*, pages 49–74. MIT Press, 1987.

[2] J. S. Albus and D. R. Blidberg. Control system architecture for multiple autonomous undersea vehicles (mauv). In *Proceedings of the Fifth International Symposium on Unmanned, Untethered Submersible Technology*, pages 22–24, 1987.

[3] J. S. Albus, R. Lumia, J. Fiala, and A. Wavering. Nasrem: The nasa/nbs standard reference model for telerobot control system architecture. Technical report, Robot Systems Division: National Institute of Standards and Technology, 1994.

[4] M. Alighanbari and J. P. How. Decentralized task assignment for unmanned aerial vehicles. In *Proceedings of the 44th IEEE Conference on Decision and Control (CDC-ECC '05)*, pages 5668–5673, 2005.

[5] R. C. Arkin. *Towards Cosmopolitan Robots: Intelligent Navigation in Extended Man-Made Environments*. PhD thesis, University of Massachusetts Amherst, 1987.

[6] R. C. Arkin and T. R. Balch. Aura: Principles and practice in review. *JETAI*, 9(2-3):175–189, 1997.

[7] H. Asama, A. Matsumoto, and Y. Ishida. Design of an autonomous and distributed robot system: Actress. In *Proceedings of the IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS 89)*, pages 283–290, 1989.

[8] R. S. Aylett, A. M. Coddington, R. A. Ghanea-Hercock, and D. P. Barnes. Heterogeneous agents for multi-robot cooperation. In *IEE Colloquium on Design and Development of Autonomous Agents*, volume 3, pages 1–7, 1995.

[9] D. Barnett, S. McClaran, E. Nelson, M. McDermott, and G. Williams. Architecture of the texas a&m autonomous underwater vehicle controller. In *Proceedings of the 1996 Symposium on Autonomous Underwater Vehicle Technology (AUV '96)*, pages 231–237, 1996.

[10] G. Beni and J. Wang. Swarm intelligence. In *Proceedings of the 7th Annual Meeting Robotics Society Japan*, pages 425–428, 1989.

[11] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2:14–23, 1986.

[12] A. Burns and A. Wellings. *Real-Time Systems and Programming Languages*. Pearson - Addison Wesley, third edition, 2001.

[13] M. D. Byington and B. E. Bishop. Cooperative robot swarm locomotion using genetic algorithms. In *40th Southeastern Symposium on System Theory (SSST)*, pages 252–256, 2008.

[14] A. Cai, T. Fukada, F. Arai, T. Ueyama, and A. Sakai. Hierarchical control architecture for cellular robotic system - simulations and experiments. In *Proceedings of the 1995 IEEE International Conference on Robotics and Automation*, volume 1, pages 1191–1196, 1995.

[15] Y. U. Cao, A. S. Fukunaga, and A. B. Kahng. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4(21):7–27, March 1997.

[16] J. Cartwright, N. Johnson, B. C. Davis, Z. Qiang, A. Linares, A. Enoch, G. Lemaitre, H. Roth, and Y. Petillot. Nessie iii autonomous underwater vehicle for sauc-e 2008. In *Proceedings of The 10th Unmanned Underwater Vehicle Showcase*, 2008.

[17] R. Charton, A. Boyer, and F. Charpillet. Learning of mediation strategies for heterogeneous agents cooperation. In *Proceedings of the 15th IEEE*

International Conference on Tools with Artificial Intelligence, pages 330–337, 2003.

[18] D. D. Corkill. Design alternatives for parallel and distributed blackboard systems. In R. D. V. Jagannathan and L. S. Baum, editors, *Blackboard Architectures and Applications*, pages 99–136. Academic Press, 1989.

[19] D. D. Corkill. Blackboard systems. *AI Expert*, 6:40–47, 1991.

[20] B. C. Davis, P. Patrón, and D. M. Lane. An augmented reality architecture for the creation of hardware-in-the-loop & hybrid simulation test scenarios for unmanned underwater vehicles. In *Proceedings of IEEE Oceans*, pages 1–6, 2007.

[21] Y. Dongyong, T. Qiong, F. Luping, Z. Xiao, and J. Jingping. Cooperative multi-agent transport based on coevolution. In *SICE 2004 Annual Conference*, volume 2, pages 1301–1304, 2004.

[22] DSTL. Sauc-e http://www.dstl.gov.uk/news_events/competitions/sauce/-index.php, 2008.

[23] E. H. Durfee. Blissful ignorance: Knowing just enough to coordinate well. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, pages 406–413, 1995.

[24] D. Edwards, T. Bean, D. Odell, and M. Anderson. A leader-follower algorithm for multiple auv formations. In *Autonomous Underwater Vehicles (IEEE/OES)*, pages 40–46, 2004.

[25] L. D. Erman, F. Hayes-Roth, V. R. Lesser, and D. R. Reddy. The hearsay-ii speech understanding system: Integrating knowledge to resolve uncertainty. *ACM Computing Survey*, 12:213–253, 1980.

[26] R. Eustice, H. Singh, J. Leonard, M. Walter, and R. Ballard. Visually navigating the rms titanic with slam information filters. In *Proceedings of Robotics Science and Systems*, 2005.

[27] J. Evans, C. C. Sotzing, P. Patrón, and D. M. Lane. Cooperative planning architectures for multi-vehicle autonomous operations. In *Proceedings of the 1st SEAS DTC Technical Conference*, 2006.

[28] E. Fiorelli, N. E. Leonard, P. Bhatta, D. Paley, R. Bachmayer, and D. M. Fratantoni. Multi-auv control and adaptive sampling in monterey bay. In *Proceedings of the IEEE/OES Autonomous Underwater Vehicles*, pages 134–147, 2004.

[29] T. Fukada and S. Nakagawa. Dynamically reconfigurable robotic system. In *Proceedings of the 1988 IEEE International Conference onRobotics and Automation*, volume 3, pages 1581–1586, 1988.

[30] E. Gat. On three-layer architectures. In *Artificial Intelligence and Mobile Robots*. MIT/AAAI Press, 1998.

[31] R. Ghabcheloo, A. P. Aguiar, A. Pascoal, and C. Silvestre. Coordinated path-following control of multiple auvs in the presence of communication failures and time delays. In *7th IFAC Conference on Maneouvring and Control of Marine Craft*, 2006.

[32] R. P. Goldman, K. Z. Haigh, D. J. Musliner, and M. J. Pelican. Macbeth: A multi-agent contraint-based planner. In *Proceedings of the 21st Digital Avionics Systems Conference*, volume 2, pages 7E3–1–7E3–8, 2002.

[33] R. Hartley and F. Pipitone. Experiments with the subsumption architecture. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1652–1658, 1991.

[34] M. J. Huber and E. H. Durfee. Deciding when to commit to action during observation-based coordination. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS)*, pages 163–170, 1995.

[35] K. Hwang, H. C.-H. Hsu, and A. Liu. A homogeneous agent architecture for robot navigation. In *Proceedings of the 2003 International Conference on Neural Networks and Signal Processing*, volume 1, pages 310–315, 2003.

[36] M. J. B. Krieger, J.-B. Billeter, and L. Keller. Ant-like task allocation and recruitment in cooperative robots. *Nature*, 406(6799):992–995, August 2000.

[37] K. H. Low, W. K. Leow, and M. H. A. Jr. Continuous-spaced action selection for single and multi-robot tasks using cooperative extended kohonen maps. In *2004 IEEE International Conference on Networking, Sensing and Control*, volume 1, pages 198–203, 2004.

[38] J. Mackelburg. Auss (advanced unmanned search system). In *Proceedings of the 2nd International Symposium on Unmanned Untethered Submersible Technology,*, volume 2, pages 5–8, 1981.

[39] S. Mandutianu, F. Hadaegh, and P. Elliot. Multi-agent system for formation flying missions. In *Proceedings of the IEEE Aerospace Conference*, volume 6, pages 2793–2802, 2001.

[40] J. E. Manley. Multiple auv missions in the national oceanic and atmospheric administration. In *Proceedings of IEEE/OES Autonomous Underwater Vehicles*, pages 20–25, 2004.

[41] V. Matellán and D. Borrajo. Abc2 an agenda based multi-agent model for robots control and cooperation. *Journal of Intelligent and Robotic Systems*, 32:93–114, 2001.

[42] P. McDowell, B. Bourgeois, and S. S. Iyengar. Formation maneuvering using passive acoustic communications. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA '04)*, volume 4, pages 3843–3848, 2004.

[43] R. K. Mehra, J. D. Boskovic, and S. Li. Autonomous formation flying of multiple ucavs under communication failure. In *IEEE 2000 Position Location and Navigation Symposium*, pages 371–378, 2000.

[44] F. Mondada, L. M. Gambardella, D. Floreano, S. Nolfi, J.-L. Deneubourg, and M. Dorigo. The cooperation of swarm-bots. *IEEE Robotics and Automation Magazine*, 12:21–28, 2005.

[45] M. N. Nicolescu and M. J. Mataric. A hierarchical architecture for behavior-based robots. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 227–233, 2002.

[46] W. Nodland, T. Ewart, W. Bendiner, and J. M. E. Aagaard. Spurv ii-an unmanned, free-swimming submersible developed for oceanographic research. In *Proceedings of Oceans*, volume 13, pages 92–98, 1981.

[47] W. E. Nodland. A general description of the self-propelled underwater research vehicle. Technical report, Applied Physics Laboratory, University of Washington, 1968.

[48] Ocean Systems Laboratory. Oceanshell: An embedded library for distributed applications and communications. Technical report, Heriot-Watt University, 2008.

[49] U. Y. Ogras, O. H. Dagci, and U. Ozguner. Cooperative control of mobile robots for target search. In *Proceedings of the IEEE International Conference Mechatronics (ICM '04)*, pages 123–128, 2004.

[50] L. E. Parker. Alliance: An architecture for fault tolerant multirobot cooperation. *IEEE Transactions on Robotics and Automation*, 14:220–240, 1998.

[51] L. E. Parker. Current state of the art in distributed autonomous mobile robotics. In *Proceedings of the Fifth International Symposium on Distributed Autonomous Robotic Systems*, 2000.

[52] H. V. D. Parunak, S. Brueckner, R. Matthews, J. Sauter, and S. Brophy. Real-time evolutionary agent characterization and prediction. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, 2007.

[53] P. Patrón, E. Miguelañez, Y. R. Petillot, and D. M. Lane. Fault tolerant adaptive mission planning with semantic knowledge representation for autonomous underwater vehicles. In *Proceedings of IEEE IROS 2008*, 2008.

[54] G. A. S. Pereria, M. F. M. Campos, and V. Kumar. Decentralized algorithms for multi-robot manipulation via caging. *The International Journal of Robotics Research*, 23:783–795, 2004.

[55] M. Powers and T. Balch. Value-based communication preservation for mobile robotics. In *7th International Symposium on Distributed Autonomous Robotic Systems*, 2004.

[56] P. Ridao, J. Batlle, J. Amat, and G. N. Roberts. Recent trends in control architectures for autonomous underwater vehicles. *International Journal of Systems Science*, 30(9):1033–1056, 1999.

[57] P. Ridao, J. Yuh, J. Batlle, and K. Sugihara. On auv control architecture. In *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)*, volume 2, pages 855–860, 2000.

[58] S. Sariel, T. Balch, and J. Stack. Distributed multi-auv coordination in naval mine countermeasure missions. Technical Report GIT-GVU-06-04, Georgia Institute of Technology, 2006.

[59] J. Shao and L. Wang. Platform for cooperation of multiple robotic fish - robofish water polo. In *46th IEEE Conference on Decision and Control*, pages 1423–1428, 2007.

[60] R. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun, and H. Younes. Coordination for multi-robot exploration and mapping. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 852–858, 2000.

[61] R. G. Simmons. Structured control for autonomous robots. *IEEE Transactions on Robotics and Automation*, 10(1):34–43, 1994.

[62] R. G. Smith. The contract net protocol: High level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, 29(12):1104–1113, 1980.

[63] R. S. Smith and F. Y. Hadeagh. Control topologies for deep space formation flying spacecraft. In *Proceedings of the 2002 American Control Conference*, volume 4, pages 2836–2841, 2002.

[64] C. C. Sotzing, W. M. Htay, and C. B. Congdon. Gencem: A genetic algorithms approach to coordinated exploration and mapping with multiple automous robots. In *Proceedings of IEEE International Congress on Evolutionary Computation (CEC-05)*, volume 3, pages 2317–2324, 2005.

[65] C. C. Sotzing, N. Johnson, and D. M. Lane. Improving multi-auv coordination with hierarchical blackboard-based plan representation. In *Proceedings of the 27th Workshop of the UK Planning and Scheduling Special Interest Group (PlanSIG)*, pages 110–117, 2008.

[66] D. J. Stilwell and B. E. Bishop. Platoons of underwater vehicles. *IEEE Control Systems Magazine*, 20:45–52, 2000.

[67] R. P. Stokey, L. E. Freitag, and M. D. Grund. A compact control language for auv acoustic communication. In *Proceedings of Oceans 2005 - Europe*, volume 2, pages 1133–1137, 2005.

[68] P. Stone and M. Veloso. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robotics*, 8(3):345–383, 2000.

[69] A. W. Stroupe, R. Ravichandran, and T. Balch. Value-based action selection for exploration and mapping with robot teams. In *Proceedings of the IEEE International Conference on Robotics & Automation (ICRA)*, 2004.

[70] S. Thrun. Robotic mapping: A survey. In *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.

[71] J. Y. Tien, G. H. Purcell, L. R. Amaro, L. E. Young, M. Aung, J. M. Srinivasan, E. D. Archer, A. M. Vozoff, and Y. Chong. Technology validation of the autonomous formation flying sensor for precision formation flying. In *Proceedings of the 2003 IEEE Aerospace Conference*, volume 1, pages 1–12, 2003.

[72] G. Trimble. Autonomous operation of the explosive ordnance disposal robotic work package using the cetus untethered underwater vehicle. In *Proceedings of the 1996 Symposium on Autonomous Underwater Vehicle Technology*, pages 21–27, 1996.

[73] United States Navy. The navy unmanned undersea vehicle (uuv) master plan. Technical report, United States Navy, 2004.

[74] J. Vazquez and I. Tena-Ruiz. Decentralised simultaneous localisation and mapping for auvs. In *Proceedings of Oceans 2007 - Europe*, 2007.

[75] S. Verret. Current state of the art in multirobot systems. Technical report, Defence Research and Development Canada, 2005.

[76] J. M. Vidal and E. H. Durfee. Recursive agent modelling using limited rationality. In *Proceedings of the First International Conference on Multi-Agent Systems*, pages 376–383, 1995.

[77] C. von Alt. Autonomous underwater vehicles. In *Autonomous Underwater Lagrangian Platforms and Sensors Workshop*, March 2003.

[78] C. von Alt, B. Allen, T. Austin, N. Forrester, R. Goldsborough, M. Purcell, and R. Stokey. Hunting for mines with remus: A high performance, affordable, free swimming underwater robot. In *OCEANS '01*, volume 1, pages 117–122, 2001.

[79] X. Wang. Planning while learning operators. In *Proceedings of the Third International Conference on Artificial Intelligence Planning Systems*, pages 229–236, 1996.

[80] G. Weiss. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. The MIT Press, 1999.

[81] J. Wen, H. Xing, X. Luo, and J. Yan. Multi-agent based distributed control system for an intelligent robot. In *Proceedings of the 2004 IEEE International Conference on Services Computing (SCC 2004)*, pages 633–637, 2004.

[82] L. L. Whitcomb. Underwater robotics: Out of the research laboratory and into the field. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '00)*, volume 1, pages 709–716, 2000.

[83] S. Yu, T. Ura, and N. Yoshiaki. Multi-auv based cooperative observations. In *Proceedings of IEEE/OES Autonomous Underwater Vehicles*, pages 7–13, 2004.

[84] J. Yuh, S. K. Choi, C. Ikehara, G. H. Kim, G. McMurty, M. Ghasemi-Nejhad, N. Sarkar, and K. Sugihara. Design of a semi-autonomous underwater vehicle for intervention missions (sauvim). In *Proceedings of the 1998 International Symposium on Underwater Technology*, pages 63–68, 1998.

[85] S. Zelinski, T. J. Koo, and S. Sastry. Hybrid system design for formations of autonomous vehicles. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, 2003.

[86] D. Zhang, Y. Fang, G. Xie, J. Yu, and L. Wang. A coordination method for multiple biomimetic robotic fish box-pushing. In *IEEE International Conference on Mechatronics and Automation*, volume 2, pages 940–945, 2005.

[87] G. Zhijun, Y. Guozheng, D. Guoqing, and H. Heng. Research of communication mechanism of multi-agent robot systems. In *Proceedings of the 2001 International Symposium on Micromechatronics and Human Science (MHS 2001)*, pages 75–79, 2001.