Distributed Communication in Swarms of Autonomous Underwater Vehicles

Felix Stephan Schill

A thesis submitted
for the degree of Doctor of Philosophy
of the Australian National University

July 2007

THE AUSTRALIAN NATIONAL UNIVERSITY

Department of Information Engineering
Research School of Information Sciences and Engineering
The Australian National University

Statement of originality

The work presented in this thesis is the result of original research done by myself, in collaboration with others, while enrolled as a Doctor of Philosophy student in the Department of Systems Engineering, later renamed to the Department of Information Engineering, at the Research School of Information Sciences and Engineering at The Australian National University. It has not been submitted for any other degree or award in any other university or educational institution.

Parts of this thesis are based on work described in the following publications, that appeared in refereed journals and conference proceedings, and which I completed while I was a Doctor of Philosophy student:

1. Felix Schill, Uwe R. Zimmer, and Jochen Trumpf. Visible spectrum optical communication and distance sensing for underwater applications. In Proc. ACRA 2004, 2004.
2. Felix Schill, Jochen Trumpf, and Uwe R. Zimmer. Towards optimal TDMA scheduling for robotic swarm communication. In Proceedings Towards Autonomous Robotic Systems, 2005.
3. Felix Schill and Uwe R. Zimmer. Distributed dynamical omnicast routing. Complex Systems (intl. Journal), 16(4):299-316, 2006.
4. Felix Schill and Uwe R. Zimmer. Effective communication in schools of submersibles. In Proceedings IEEE OCEANS'06, 2006.
5. Felix Schill and Uwe R. Zimmer. Pruning local schedules for efficient swarm communication. In Proceedings of the International Symposium on Underwater Technology, Tokyo, Japan, 2007.
6. Ram Somaraju and Felix Schill. A communication module and TDMA scheduling for a swarm of small submarines. Tr. J. of Electrical Engineering and Computer Sciences, Special Issue on Swarm Robotics, 2007.

Felix Schill

Abstract

Effective communication mechanisms are a key requirement for schools of submersible robots and their meaningful deployment. Large schools of identical submersibles require a fully distributed communication system which scales well and optimises for "many-to-many" communication (omnicast, also known as gossiping). As an additional constraint, communication channels under water are typically very low bandwidth and short range. This thesis discusses possible electric and electro-magnetic wireless communication channels suitable for underwater environments. Theoretical findings on the omnicast communication problem are presented, as well as the implementation of a distributed time division multiple access (TDMA) scheduling algorithm in simulation and in hardware. It is shown theoretically and in simulation that short range links in a robotic swarm are actually an advantage, compared to links that cover large parts of the network. Experiments were carried out on custom-developed digital long-wave radio and optical link modules. The results of the experiments are used to revisit the initial assumptions on communication in multi-hop wireless networks.

Acknowledgements

I would like to thank my supervisors Uwe Zimmer and Jochen Trumpf for their help, guidance and support. Their door was always open, and they always found the time to give advice and to discuss interesting research. I am grateful that I was given the opportunity to learn a lot about science, robotics, mathematics, electronics and many other areas, and that I had the freedom and flexibility to engage in many small side projects.

I would also like to thank Uwe Zimmer and Navinda Kottege for interesting discussions over lunch, good company on international trips, movie screenings, and generally the friendly atmosphere that made the last four years an enlightening and enjoyable experience. Many thanks go to Andrew Dankers; the last four years almost seemed too short for the countless ideas and projects we came up with and sometimes realised. I wonder how many robots, flying machines, reinvented wheels and other contraptions would exist if we hadn't been so busy writing our theses.

I would like to thank Helen Lindsay for supporting me, especially in the busy last few months before submission, and for proof-reading. To my friends in the ANU Mountaineering Club, thank you for making my stay in Australia a great experience. Also many thanks to all my friends and colleagues at RSISE; I enjoyed the friendly and open environment.

Most importantly I would like to thank my parents Peter and Angelika. They fuelled my interest in science, education and technology from early on, and they always encouraged and supported me to pursue the path I have taken.

Contents

List of Figures

Chapter 1

Introduction

How is the weather in Berlin? What is the wind speed in Sydney, or the temperature in Fiji? Questions like this can now be easily answered by anyone with internet access. Even more so, every high and low pressure system world wide is mapped; a fine meshed network of weather stations measures pressure distributions, wind speeds, humidity, cloud coverage and many other values with high resolution and global coverage in 10 minute intervals. These "on-location" measurements are augmented by remote sensing from satellites or ground based weather radars. Weather monitoring is just one example where distributed sensing and wide area measurements have become standard. Other examples are hyper-spectral imaging from satellites for monitoring forest growth or parasites, wide-area video surveillance, and traffic monitoring.

All these examples leave out the oceans completely. Most of the knowledge about the vast amounts of water in the earth's oceans concentrates on the boundaries - the surface, the seafloor, and the coast lines. Hardly anything is known about the properties of the water as such. This can partially be explained by less public interest in the oceans. However, more detailed knowledge about the oceans is crucial for understanding climate change, managing sustainable fishing, monitoring pollution, or monitoring coral reefs and endangered species, and will

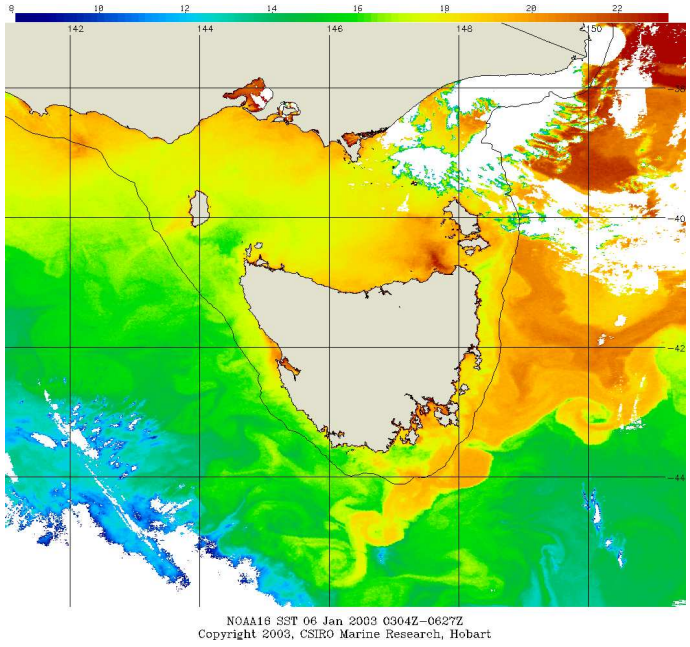

Figure 1.1: Sea surface temperature off the coast of Australia from a satellite infrared image (Image and copyright by CSIRO Marine and Atmospheric Research; reproduced with permission)

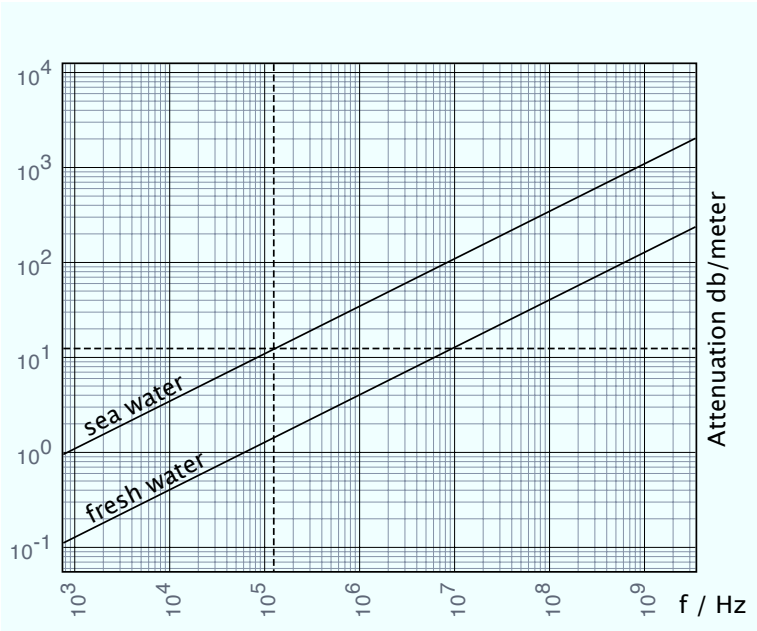

Figure 1.2: Approximate attenuation of radio waves in fresh water and sea water

become increasingly important in the future. There is great interest in being able to measure properties such as temperature, salinity, currents, pollution, nutrients, concentration of gases (i.e. oxygen or methane), not only in a single spot at a time but in given areas or volumes of water. Such a spatially distributed map of measurements makes it possible to compute gradients, find sources of emission, or, if measurements are taken over some time span, it allows to measure rates of change, to track plumes, or to observe correlations to time of the day. At the present moment it is not possible to obtain such distributed measurements below the surface.

1.1 Spatio-temporal oceanographic sensing

A major problem with obtaining wide-area measurements from the oceans is that remote sensing under water is much more difficult than above water. Satellite based sensing usually employs electro-magnetic waves from high-frequency radio across infrared, visible light, ultraviolet to X-Ray and gamma radiation. Water is typically opaque for most electro-magnetic waves, with only narrow bands having limited transparency. Attenuation is typically hundreds or thousands of decibel per metre for radio waves (figure 1.2). Very low frequencies are less affected but even for long-wave radio ranges are limited to 10-100 metres. For light there is a narrow gap for wave lengths from 400 nm to 500 nm where attenuation is low. Very clean water can offer visibility of 100 metres or more for blue light. Unfortunately these wave lengths are severely affected by scattering from suspended particles, reducing the overall visibility to less than 10 metres in many areas. This limits measurements that can be performed from satellites or planes to the surface of the oceans. In the far infrared region (heat radiation) water is opaque and acts as a black body radiation source. This makes it possible to measure the surface temperature using infrared cameras. Figure 1.1 shows an example. There is no similar way of remotely sensing the temperature a few metres below the surface.

From boats or submarines it is possible to use sonar (acoustic) sensing to obtain measurements from a distance. Sonar measurements mainly give information about solid bodies in the water volume, such as rocks, the seafloor, vessels and fish. Again, this limits remote sensing to obtaining information about the boundaries of a water volume. Over larger distances distortions occur due to

changes in the speed of sound in water depending on temperature, salinity, pressure and other factors. It is theoretically possible to measure differences in time of flight and phases, in order to reconstruct some of these parameters. Obviously this can only work if parameters are estimated for every point (or a dense approximation) of the volume covered by sound rays. To achieve a sufficiently high resolution, a large number of sound emitters and receivers is needed, which have to be spatially separated. It is conceivable that a very large acoustic array could be able to reconstruct a three-dimensional map of temperature, salinity and pressure over a volume of water. However, there is no result known to the author where this has been achieved, and it is expected to be extremely difficult.

Since remote sensing for obtaining spatially distributed measurements of volumes of water is not possible, the only alternative is sensing on location. In a dynamic environment it is not sufficient to scan an area or volume with a single sensor, because the changes are not negligible over the duration of an entire scan. This means that measurements taken at different locations but at different times are not comparable. It follows that a large number of sensors is required which have to be placed at known positions (at least relative to the other sensors).

There is also a problem of scale and resolution. As can be seen in figure 1.1, the water body can be quite turbulent. In this particular example this means that there are significant temperature variations even at very small scales. With limited resources there is a trade-off between resolution and area coverage. The larger the monitored area is, the lower the resolution will be, assuming the same number of sensors. The vastness of the world's oceans means that resources will probably always be limited to a very small body of water. A possible solution is to install a coarse resolution system (i.e. a submarine cable network with sensors, or satellite-based sea surface surveillance), and to employ higher resolution sensor networks or robotic swarms to obtain a higher resolution in small areas of interest. For wide-area coarse resolution sensing, practical distances between sensors are around 1 kilometre. Finer-scale sensing systems would therefore have to cover the areas between these sensors. A sensor distance of 10 metres or less is realistic. If even finer resolution is required, mobile sensor nodes (i.e. robotic swarms) can be used to adapt the resolution to the requirements, and to achieve the area coverage by following a covering trajectory.

1.2 Currently used technology

Distributed sensing in oceanic environments still has only a marginal role today, but the importance has been recognized by the scientific community.

Principally there are two possibilities for obtaining distributed measurements - sensor networks and swarms or formations of propelled vehicles. Both have advantages and disadvantages depending on the application. Sensor networks have already been deployed in military applications and small scale scientific installations. The first large scale sensor network was probably the SOSUS (*SOund SUrveillance System*) of the U.S. Navy [58]. This system was used for early detection of military submarines crossing the Atlantic by detecting and tracking their acoustic emissions.

The U.S. National Science Foundation allocated several hundred million dollars for the Ocean Observatories Initiative in 2005, with the stated goal of installing wide-ranging coastal and global sensor networks using submarine cables and moored buoys [7]. Scientific drivers of the initiative include research on climate variability, ecosystems, ocean food, ocean dynamics, global geodynamics, fluid rock interactions and sub-seafloor biosphere. The vision is an interactive subsea sensing network, combining live updates from permanently installed sensor networks, remote controlled underwater vehicles and groups of autonomous vehicles. All data will be accessible through the internet and the system will be able to respond to requests from remotely connected researchers. As of 2007 the first coastal cabled networks are being set up.

There have been initial experiments with small formations up to twelve vehicles (Autonomous glider project). The formations were not able to communicate during dives. Up to the present day there has been no successful trial with a larger swarm of autonomous vehicles. The following sections will look at the details of sensor networks and swarms; the difficulties, advantages and disadvantages.

1.2.1 Sensor networks

A sensor network is an installation of a large number of relatively simple sensor nodes, which are connected by a communication network. This network can be wired or wireless. Wired networks offer the advantage that the sensor nodes can be supplied with power, which allows for indefinite operation. Wired networks

also offer very high data bandwidths. Installation of a wired network can be difficult in marine environments, and the wires are prone to damage caused by corrosion, ship anchors, marine lifeforms such as shellfish, rocks or landslides. Wireless networks in underwater environments have very low range, need a lot of power and are usually severely bandwidth limited. The problems of wireless communication in water will be explained in detail in the next chapter.

Wired sensor networks can cover large areas and can operate for a very long period of time. Often sensors are embedded into a long cable which provides power and communication. The cable can then be towed behind a boat, or deployed on the seafloor. This technique is commonly used for sonar arrays. These arrays only offer high resolution along the length of the cable. To cover an area or a volume, a lot of cables would have to be deployed in parallel, which can be impractical. Other topologies have been proposed, i.e. tree-like layouts.

Wireless sensor networks can be deployed covering areas and volumes with even resolution, provided that the distance between nodes is within the communication range. They are therefore more flexible, but limited in battery life, communication bandwidth and range. Depending on the maximum battery life, deployment and retrieval can be a significant time overhead compared to the actual time of sensing.

1.2.2 Swarms

A swarm can be defined as a group of individuals that aggregate and travel in the same direction. This definition does not necessarily require the direction of travel being strictly identical for all individuals at all times, but that to an extent the swarm stays within a bounded volume after formation. Further characteristics are decentralised control, self-organisation, and some form of collective or emergent behaviour. In the context of robotics, groups of robots travelling as a group in the same direction are sometimes referred to as a formation. The main differences are that formations typically have a limited number of individuals, and that their positions with respect to the group are rigidly defined. In a swarm the individuals are interchangeable, and the precise positions are not defined, but emerge from behavioural rules. Hence the exact number of individuals does not change the general behaviour or shape of the swarm, and the swarm can scale to large numbers of individuals.

Swarms share some aspects of sensor networks. The nodes are fairly simple, there is a large number of them, and they maintain communication amongst each other. The main mechanical difference is the ability to change position relative to other nodes or the world. It is usually assumed that nodes can actively manoeuvre by some means of propulsion. However, a swarm is more than a number of sensor nodes that can automatically deploy themselves in the manner that a sensor network would be deployed by external forces. A swarm can act as a sensor network, and achieve in principal the same results. Additionally, the mobility allows nodes to react adaptively to measured data. This opens up a range of applications that can not be addressed by passive sensor networks, such as gradient following to find a maximum or minimum, isocline adaptation, adaptable sensing resolution by changing the node density and the size of the swarm, to name a few.

For practical reasons communication in swarms has to be wireless. This implies that swarm nodes have to carry their own power source, which limits the usable mission time. Propulsion requires significantly more energy than sensing and basic computing. This means that the endurance of a swarm is typically an order of magnitude less than a comparable battery-powered sensor network. However, depending on the application, swarm nodes can position themselves and then deactivate the propulsion, effectively turning the swarm robots into passive sensor nodes.

Robotic swarms are more powerful and flexible than sensor networks, and can be deployed more quickly and easily. This makes them suitable for short term measurements (up to one day) or for applications where online adaptation is required. For long term observations (weeks or months) a sensor network is usually better suited.

From a technological perspective most techniques and technologies that apply to sensor networks also apply to robotic swarms, but swarms add additional complexity due to the mobility of nodes. This thesis will concentrate on the more general concept of robotic swarms. It is hoped that results that apply to robotic swarms will also be beneficial to the field of sensor networks.

1.3 Preconditions for swarming

Given a group of robots, which preconditions have to be met, and which capabilities do these robots need to form a swarm? In the distributed behavioural model published by Craig W. Reynolds in 1987 [39], it is assumed that swarming robots can sense the relative direction and distance to (some of) their neighbours in the swarm. For swarm formation and control this localised relative position sensing is already sufficient. If applications are taken into account, then nodes also have to be able to communicate at least locally in order to exchange measured sensor data for estimating gradients, extrema and averages and also to coordinate global behaviour changes.

1.4 Communication problems in underwater swarms

Swarms of autonomous underwater vehicles have particular requirements regarding the communication between swarm nodes, that are special either with respect to the underwater environment, to swarming, or a combination of both. The above mentioned high attenuation of electro-magnetic radiation in water severely impedes radio communication. The vast majority of available wireless communication technology is designed for use in air or vacuum, and is based on high frequency radio waves. This makes it impossible to make use of commercially available technology for underwater applications.

There are commercially available sonar modems for underwater communication, which make use of acoustic waves that travel relatively large distances. Most of these modules are rather large, expensive and power hungry. This only allows operation on larger vehicles. As will be discussed in chapter 2, it is beneficial in many regards to miniaturise robotic swarming vehicles as far as possible for reasons of practicability, handling, deployment and cost. Unfortunately this prohibits the use of many of the available acoustic modems. Furthermore, current acoustic modems are designed for point-to-point operation, but not for large-scale multi-node networks.

A problem of acoustic communication is that sound travels relatively large distances in water. Due to multi-path propagation, reflections and echoes, the signal can only be decoded in a range around the sender that is relatively small compared to the distance at which the acoustic emissions can still be detected.

In other words, a vehicle has a large range of interference, but only a relatively short range for usable communication. This is not so much a problem for point-to-point connections to a single vehicle. For large swarms, however, this means that the noise level gets very high.

With regard to multiple channel access, there are further problems with acoustic communication. The slow propagation speed of sound in water introduces delays and time shifts over larger distances. In addition to the time delay for the actual communication, echoes cause a further delay until the channel is free of interference again. These delays get increasingly worse for large geometric swarm sizes and large numbers of vehicles. Particular advantages and disadvantages of communication channels are described in more detail in chapter 3.

Communication requirements of a coordinated group of robots is a topic that is rarely discussed in literature. The distribution of information, timing and latency requirements, occurrence of communication and other criteria are very different to more common network setups, such as mobile phone or computer networks. Many aspects are discussed in literature on sensor networks. An additional constraint of robotic swarms is the constantly changing network topology. Optimising a communication network for swarms has impacts on many levels, from the physical medium and channel access to the highest levels of data abstraction. The problem of channel access is addressed in chapters 4 and 5. Higher levels of abstraction, such as decentralised mutual information exchange, consensus or swarm control are not addressed this thesis. The author believes that these topics do not fall into the domain of communication in swarms, but rather into the domain of swarm control and specific applications.

The problem of channel access in swarms and reliable real time communication in changing network topologies also applies to robotic swarms on land or in the air. For most applications the problem does not pose such severe constraints, as communication links for air are orders of magnitude faster than what is required for reliable control of the swarm. It is generally possible to use off-the-shelf hardware and reserve a few percent of the bandwidth for real time data exchange. While not strictly real time, latencies and time jitter are so low that they are often ignored. Due to the high available bandwidth, network reconfigurations can also be achieved in very short time. The combination of swarm communication and the underwater environment results in a more

severe problem. The available bandwidth and communication range are so low that they have to be fully utilised. Strategies on managing and sharing the limited communication resources have a large impact on the overall performance and viability of a swarm. A recurring theme of this thesis is the combination of available communication channels and effective strategies for sharing and managing them in swarm applications.

1.5 Contributions of this thesis

The key problem addressed in this thesis is highly parallel, scalable and fully distributed real time communication in underwater swarms. Constraints are: slow and short range communication links, only limited local knowledge of the network topology, rapid changes in the network topology and the impossibility of centralised control. It is argued that swarms require many-to-many communication (*omnicast*) for meaningful applications. This thesis provides a solution for a communication algorithm together with suitable hardware, that allows a large number of submarines to form and maintain a network ad hoc; once formed the network enables the robots to exchange information locally and globally with known upper bounds and with predictable timing (chapter 4 and 5).

Underwater wireless communication is subject to severe constraints. In order to show the feasability of wireless communication in underwater swarms, it is subject of this thesis to investigate possible wireless communication channels. Emphasis is put on electric and electro-magnetic channels to find out if these modalities can offer a suitable alternative to acoustic communication. A long-wave communication module and an optical communication module are given as examples for short range underwater communication. The little-known return current effect is revisited, and high-resolution measurements of the current density in freshwater are presented. The prior knowledge of return current communication in seawater is extended by presenting more conclusive experimental results, and by demonstrating the effect in freshwater. Altogether three possible communication links (optical, radio and return current) are considered to be useful for short range communication over up to 20 m distance (Chapter 3).

A recurring theme of this thesis are the implications of using short range links in a multi-hop swarm network. Obviously communication links have to reach far

enough to reach at least direct neighbours in a swarm. It is of great interest how long the range of individual communication links should be for a given geometric swarm size and density. It is shown that short range links have advantages in a swarming context.

Lastly the question is posed how multi-hop wireless networks should be modelled for the purpose of swarm communication. The graph-based network model which is commonly used in literature is investigated and used to derive theoretical results for many-to-many communication. An upper bound for *omnicast* which is linear in the number of nodes is presented. The experiments with long-wave radio modules revealed that the real behaviour of these communication links differs from the graph-based model; however, solutions which are valid for the graph based model retain their validity. A geometric/radiometric model is proposed, which can be used to improve the performance of a proposed real time TDMA scheduling algorithm. Results from a realistic distributed real time simulation are presented.

Chapter 2

Submarine system overview

In principle swarming can be implemented with almost any kind of underwater vehicle. However, there are a few practical constraints with regards to vehicle size, cost, ease of deployment and retrieval and manoeuvrability which determine if a submersible robot is suitable or practical for swarming applications. This chapter introduces the various types of underwater vehicles and describes a new design that is especially adapted for swarming.

2.1 Overview

Many types of underwater vehicles have been built to the present day, and there are a number of commercially available models for offshore industry, research, marine biology and military uses. Most distinctively, underwater vehicles can be categorised as manned submarines and unmanned underwater vehicles (UUVs); the unmanned vehicles are either Remotely Operated Vehicles (ROVs) or Autonomous Underwater Vehicles (AUVs).

Remotely Operated Vehicles are typically remote controlled from an operator's desk on a ship, to which they are connected via an umbilical cable. The umbilical

Figure 2.1: A Magellan ROV being launched into the Ocean

Figure 2.2: A SuperScorpio ROV is being loaded into a Galaxy for a submarine rescue mission in Russia. The trailer in the foreground carries the umbilical cable.

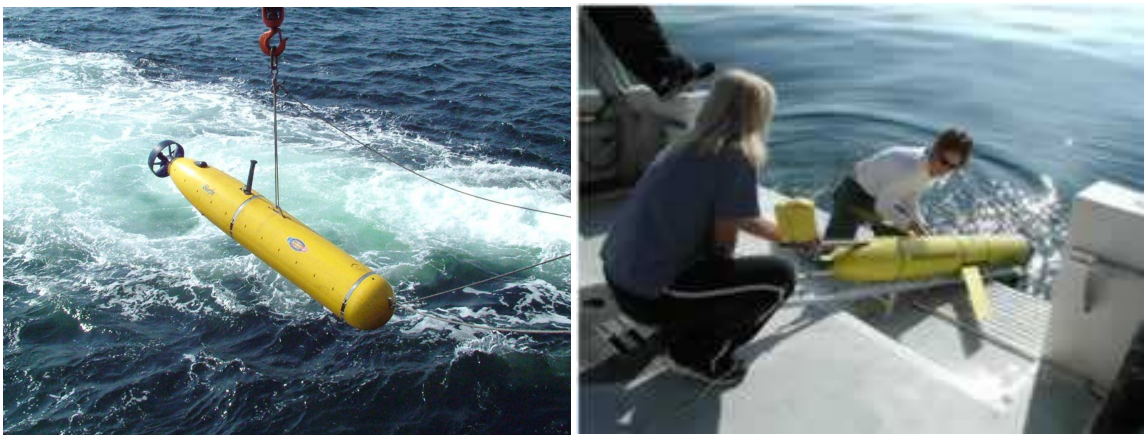

Figure 2.3: Left: A Bluefin submarine being launched (Photo by Bluefin). Right: NOAA Researchers launch a Glider

cable supplies control commands to the motors of the ROV, transfers sensor data back to the operator (often including video feeds), and usually also supplies power to the vehicle. The cable can also act as an emergency cable to retrieve the vehicle in case of mechanical faults. Typically, the cable is very robust, a few hundred or thousand metres long, and can easily weigh several hundred kilograms. The vehicle must have strong propulsion to counter the weight and the hydrodynamic drag of the cable. ROVs come in a large variety of shapes and sizes. The smallest ones can be as light as a few kilograms, and can only carry a camera or a small sensor suite. Due to the small size, they can only drag a short, thin cable, and are therefore typically operated close to the ship, or a larger submersible acting as a base platform. Workclass ROVs are quite large and heavy, ranging from 500 kg to several tonnes, and often have manipulator arms and extensive sensor arrays. Examples of commercially available ROVs are the *Double Eagle* by *Saab*, the *Hercules* by *Subsea 7*, *Super Scorpio* by *Perry Tritech* and the *Magellan 725* by *Oceaneering*.

Autonomous Underwater Vehicles (AUVs) are similar to ROVs, but do not require remote control and therefore also require no cable. Most AUVs are scientific robots. They are mostly built as a one-off model for a specific research project; only a few are sold in larger numbers. Popular models are the *Remus* by Woods Hole Oceanographic Institute, the models by *Bluefin*, or the much larger *HUGIN 3000* by *Kongsberg Gruppen*. AUVs are equipped with on board

computing for navigation and mission planning, carry their own power supply and a range of sensors (depth sensors, inertial navigation, compass, temperature, often also sonar sensor and cameras). Once deployed, the AUV carries out the mission autonomously without human interaction and returns to the surface for pickup after completion. Missions can last for several hours, or even several weeks in some special cases.

There are two principally different designs for underwater vehicles. Most ROVs are an open frame design with multiple thrusters that allow decoupled multi-dimensional movement or even holonomic movement (6 degrees of freedom). AUVs may also have an open frame design, but mostly, for reasons of hydrodynamic efficiency, a torpedo-like shape is used. The latter features a long, cigar-shaped hull with a single thruster at the rear end and rudders for changing direction. This design has low drag, but is less manoeuvrable than open frame designs. The vehicle can only change its attitude when it is moving forward. Without forward motion the rudders have no effect. Recently some groups tried to combine both concepts, e.g. CSIRO's *Starbug* vehicle [10], which has a double torpedo-shaped hull with three horizontal thrusters that allow full attitude control.

In a class of their own are *gliders* [40] [31][57]. These are typically similar to the torpedo-shaped designs, but instead of a thruster they feature horizontal hydrofoils and an active buoyancy device. Propulsion is achieved by changing the buoyancy and by using the hydro foils in a similar way as glider airplanes to glide forwards. After a period of gliding downwards, the buoyancy is changed to positive, and the submersible glides forwards back to the surface. Energy consumption is very low, and close to zero during the glides. Only the change in buoyancy and the data sampling equipment require energy. During surfacing GPS position readings are taken and the collected data is transmitted to the operators via a satellite link. Glider AUVs can carry out missions that last several weeks.

2.2 AUVs for swarming applications

There has not been a successful trial of swarming AUVs so far. Experiments have been successfully carried out with small formations of AUVs - i.e., groups of up

to 12 *SLOCUM* gliders have been deployed on a multi-AUV mission in Monterey Bay [12][4]. However, the AUVs could only communicate during repeated surfacing, but not during dives. Every AUV attempted to stay as precisely as possible on the preplanned trajectory until the next surfacing. During surfacing, collected data was transmitted back to the operators, a GPS position reading was taken and a new trajectory was planned.

While the principle of swarming can theoretically be implemented with almost any type of AUV, it is obvious that the large numbers of vehicles required pose severe constraints on handling requirements and cost. Currently even small commercially available AUVs weigh 300 kg or more and cost over $500,000. It is difficult for most research groups to finance one or two of these vehicles - a group of 20 or 30 is quite unrealistic for most budgets. Launching a 300 kg vehicle requires a crane, several people, a large boat and calm sea conditions. Efficient launching of 20 vehicles can be difficult, and it consumes significant amounts of time. Picking up the vehicles at the end of the mission poses similar problems and might involve the dangerous task of attaching a crane hook to the vehicles.

2.2.1 Performance requirements and vehicle characteristics

Performance criteria of AUVs are typically peak velocity, average cruise velocity, battery runtime, range, maximum operating depth, installed sensors and computational power. The size of a vehicle only matters for work class vehicles with manipulators, as they require a counter weight to balance heavy payloads in the manipulators. Essentially, all AUVs that are currently used are ”eyeball” vehicles - they do not have manipulation capabilities, but are able to carry a sensor payload and can be used for data sampling. The size of a vehicle is irrelevant for this task, as long as the other performance criteria are met. For swarming applications it is therefore desirable to reduce the size and weight as far as possible. A vehicle which can be launched by hand is preferable, which means that the maximum mass should not exceed 10 kg.

Currents in the open ocean can have velocities of over 2 metres per second (approximately 4 knots). Most submersibles are able to maintain a velocity of 2-3 knots, which allows them to cope with many, but not all currents. There is an obvious trade-off between velocity and energy requirements. The average cruise speed is usually chosen to maximise range, which typically translates into

roughly 1-2 knots. However, it is desirable that the vehicle can maintain higher peak velocities to fight currents.

Battery runtime and range are highly application specific and also cost driven. High density batteries typically cost an order of magnitude more than cheaper varieties with lower energy density. Obviously the battery runtime heavily depends on the use of propulsion. The two critical key factors are idle runtime (motors switched off) and cruise speed runtime, which maximises the range, but is not necessarily full speed. Actively propelled AUVs can achieve cruise runtimes of 10 hours or more. Idle runtime depends on the computing requirements - with energy optimised embedded processors, idle runtimes of several weeks are possible. Gliders form an exception by using buoyancy for propulsion - the cruise runtime is therefore almost as long as the idle runtime. The only energy required for propulsion is for pumping water into and out of the active buoyancy device. New approaches suggest to utilise the temperature gradient between surface and deep waters for this [57], which would cut energy requirements to a minimum, allowing mission times of weeks or months and ranges of thousands of kilometres.

The maximum depth also depends on the application. Commercial ROVs and AUVs are generally rated to either hundreds or thousands of metres. Typical ratings are 300 metres or 3000-6000 metres. The price tag changes accordingly by an order of magnitude. Static water pressure increases by 1 bar (roughly 1 atmosphere, or $1kg/cm^2$) per 10 metres. 100 metres depth corresponds to 10 bar pressure, which is comparable to household water mains pressure. Greater depths of thousands of metres require special materials, such as titanium, to obtain sufficient structural integrity; designing seals and connectors becomes a lot more challenging. For reasons concerning engineering and cost it is usually better to design a cheaper shallow water vehicle (up to 300 metres) and a more expensive deep sea version for special applications.

A minimum sensor suite comprises at least an inertial navigation system, depth sensor, compass and a surface communication system. Usually also included are sonar sensors - either simple, single-beam, forward or downward looking echolocation sonars; or additionally side-scanning sonars, or phased sonar arrays. Swarming robots are required to be able to locate neighbouring vehicles in the swarm. This can be achieved with an active sonar system [28][27] . A second requirement for swarming is underwater communication. Most available

underwater communication systems use acoustics, and are not able to cope with several transmitters. This thesis will propose alternatives in the next chapter. For positioning on the surface a GPS module can easily be integrated. Any additional sensors are application-specific. Examples are sensors for temperature, salinity, concentration of oxygen, methane or other gases, or cameras.

Computational requirements depend on the type of the installed sensors and their data rate. Mainly sonar and vision processing are computationally expensive. Sonar processing can often be kept simple, and real time vision processing is mostly not done due to bad visibility or general darkness. This makes it possible to use energy efficient embedded processors running at low speed. To accommodate bursts of computational load, a processor with variable speed can be used, or alternatively, the commonly available idle and standby functions can be activated while the processor is not used. The power consumption can be kept minimal this way. For low-level sonar or vision processing FPGAs can be tailored specifically for the task, which are typically much faster and more efficient than using a sequential processor.

Manoeuvrability is another important factor for swarming applications. Torpedo-like designs can only control attitude while moving forward. This can unnecessarily complicate the control of a swarm. Vehicles with holonomic control (or almost holonomic control) are able to react quickly to local changes in the swarm geometry and make control faster, easier and more reliable. However, this has to be carefully balanced against the increased hydrodynamic drag of vehicles with more degrees of freedom.

2.2.2 Small is beautiful

As mentioned before, small size is preferable for swarming applications. The main question is how small a submersible can be while still retaining its functionality. The size of an AUV is mainly determined by the size and mass of the sensors, motors, processors and the batteries. The mass in kilograms has to be equalled by volumetric displacement in litres to achieve neutral buoyancy. Sensors and processors can be miniaturised quite well with modern technology and have almost negligible volume and mass compared to the rest of the system. The main volume and mass is given by the batteries. Sizing of batteries depends on the expected range and battery runtime. The size of the vehicle determines the drag and the energy required for propulsion - this means that small vehicles

need smaller batteries. This is again counterbalanced by the standby power consumption of the processor and sensors.

There are more factors that play a role in determining a good size for the system, i.e. handling or launch and retrieval procedures. A submarine larger than 1 m and heavier than 30 kg requires a crane for launching and retrieval. If it is smaller, it can be launched by a single person, possibly with help of another person. A size around 50 cm and 5 kg makes handling with only one hand possible, making launching and retrieval very simple and efficient. Determining the best size becomes a difficult optimisation problem.

A practical approach is to make sensors and all circuitry (processor, amplifiers, etc.) as small and energy efficient as possible. The next step is to choose a battery that provides a sufficiently long idle runtime and expected cruise runtime, and to design the hull of the submersible around the volume and mass of circuitry and battery. Using this approach it appears feasible to build a submersible with a mass below 5 kg and an overall length of approximately 50 cm. The next sections describe a prototype of this class and a design study for a production model.

2.3 Serafina Mk I: Prototype development

A first prototype was designed and built in 2003 to answer the question of whether a small, cheap AUV can be built and what the performance can be. The original design was done by Alexander Bahr. The implementation and building of the vehicle was done by the author, as well as later extensions and changes.

2.3.1 Design

The initial goal was to build a submersible robot for less than $1000. This goal later had to be changed due to cost of external components such as the CPU module. The final design consists of a 50 cm pressure hull made from PVC with 5 DC thrusters, lead acid batteries, a MPC555 PowerPC module, inertial sensors, compass and pressure sensor. Figure 2.4 shows the layout of the key components. The cost is approximately $3000.

For good manoeuvrability the *Serafina I* has 5 degrees of freedom, given by 2 horizontal and 3 vertical fixed thrusters. This makes the vehicle almost

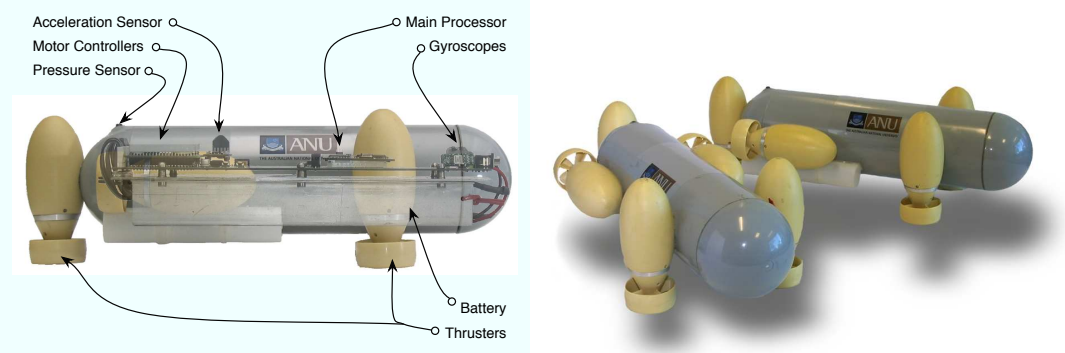

Figure 2.4: Serafina Mk I

holonomic, with the exception of sway (lateral motion). The three vertical thrusters are responsible for roll, pitch and linear vertical motion; the two horizontal thrusters provide linear horizontal motion and yaw. The overall shape of the hull has significantly less drag than an open frame design, which is very common for submersibles with 5 or 6 degrees of freedom. For simplicity all thrusters are identical and use cheap brushed DC motors. The propellers are coupled to the motors via a shaft running through a rubber o-ring for sealing. The bulky shape of the vertical thrusters increases drag in the forward direction; something that can be significantly improved by using thrusters with a smaller cross section.

Access to the interior of the hull is through the front bulkhead, which has a bayonet seal. To prevent water entering the hull, the entire air cavity is pressurised through a valve at the end of the vehicle after closing the bulkhead. Pressurisation to 500 mbar allows diving depths up to 5 metres, while the outside water pressure is always lower than the internal pressure. In the case of leaks, air exits the hull, resulting in a drop of differential pressure between inside and outside. This can be detected by the pressure sensor. The diving depth is always actively limited by maintaining a positive pressure differential between inside and outside. A pressure drop automatically leads to surfacing, preventing water from being forced into the hull.

Technical data for *Serafina I* (as tested)

Length	50 cm
Width	20 cm
Height	12 cm
Mass	4.5 kg
Displacement	4.5 litre
Buoyancy	Slightly positive
Static Trim	horizontal, slightly up-righting
Static Motor thrust (per thruster)	1 N
Maximum forward speed	1 m/s
Maximum roll rate	>300 deg/ sec
Maximum pitch rate	>200 deg / sec
Maximum yaw rate	> 50 deg / sec
Maximum operating depth	5 m
Battery type	Sealed lead acid, 12V, 4 Ahr
Cruise runtime	2 hours
Idle runtime	10 hours
Main CPU	MPC555 40 MHz PowerPC
Inertial sensors	MEMS gyroscopes and accelerometers (+/- 300 deg/sec, +/- 2 g)

2.3.2 Test results and experiences

The design and construction of the *Serafina I* prototype showed that 50 cm large submersibles are feasible. Despite the batteries using up most of the available interior space, there was still sufficient space for all the circuitry: an MPC555 PowerPC module with 40 MHz as the main processor, 3-axis accelerometer and gyroscopes, motor drivers, compass, a sonar distance sensor and a radio module for remote control. The complete circuitry with main CPU and all sensors consumes approximately 250 mA at 12 V when fully operational. The motors draw 1-2 A at full speed, resulting in a total energy consumption of less than

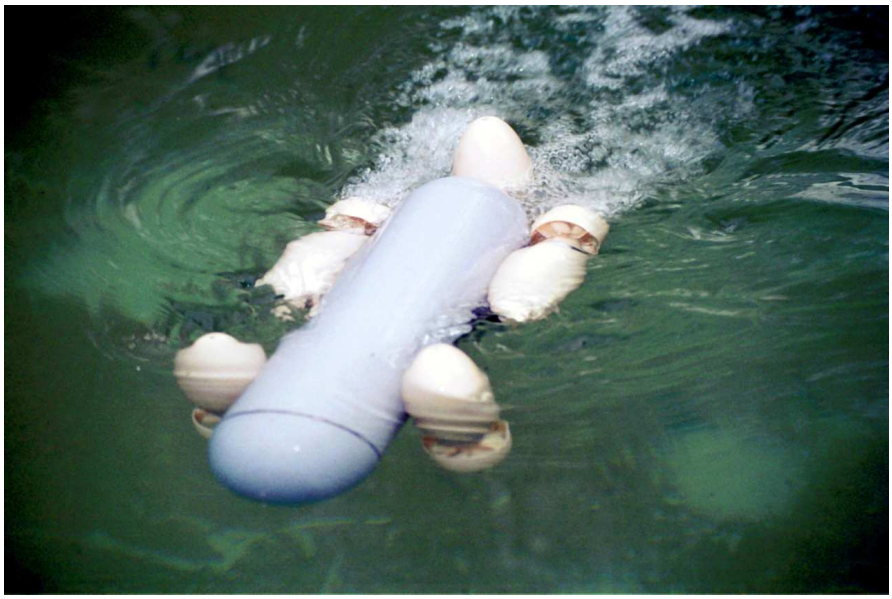

Figure 2.5: Serafina Mk I during a test run

40 W at full forward velocity. A runtime of 2 hours could be repeatedly achieved with a program of frequent motor use.

Attitude control was implemented using the sensor data from the accelerometers and gyroscopes. An inner velocity control loop maintains a given angular velocity in roll, pitch and yaw. This is the standard mode for remote control operation. A neutral joystick input results in the velocity being maintained at zero, with only slow drift. An outer control loop can be activated to lock and hold attitude. The control loops are cascaded - the attitude lock controller output is fed into the velocity input of the inner velocity controller.

Three different assisted remote controlled modes are available. The remote control unit has two joysticks with two degrees of motion each and two switches for mode selection. In all modes joystick position is proportional to velocity or angular rate. Mode 1 allows rotation of the vehicle around roll, pitch and yaw, as well as forward/reverse thrust for forward motion. Direct vertical motion is not possible in this mode, due to a lack of control joysticks on the remote controller. It is possible to activate attitude lock (mode 2). The attitude controller samples the current attitude at activation of mode 2 and maintains it. Possible control inputs

are forward/reverse and up/down (in vehicle coordinates), allowing linear motion while maintaining attitude. Mode 3 stabilises the vehicle in a horizontal position. Control inputs are forward/reverse, yaw and up/down. Mode 3 can be augmented by depth control. If the up/down joystick is neutral, the current depth is actively maintained by a control loop using depth information from the pressure sensor.

All three modes allow responsive and stable control and manoeuvring of the vehicle in any position. It is even possible to hold and stabilise the vehicle upside down and to do controlled reverse turns at constant radius. Control of the vehicle could easily be learned by beginners within minutes; Mode 3 was felt to be the easiest to master, as control is very similar to driving a car, with an additional degree of freedom which simply controls the depth. Mode 1 is more similar to controlling an airplane. Noteworthy is the fast response in roll and pitch - the maximum angular rates in both roll and pitch exceed 180 degrees per second. The maximum roll rate at full joystick deflection is one full revolution per second. Maximum forward speed is more than one metre per second (2 knots). Vertical motion and yaw are significantly slower due to hydrodynamic drag. It is significantly faster to rotate the vehicle into the desired direction and utilise the forward thrusters.

Additionally to the assisted remote controlled modes, a number of simple autonomous tests were implemented. Tests were formulated in terms of sequences of actions. The tests mainly aimed to identify vehicle parameters and to test navigational repeatability. The sonar sensor was not used at this stage. Tests showed that the Serafina could repeatedly follow a trajectory for approximately 1 minute, which involved changes in depth, sharp turns up to 90 degrees as well as pitch changes up to 45 degrees, and return to the start position with less than 50 cm error. Gyroscope drift was found to be approximately 2-3 degrees per minute without compass support.

2.3.3 Evaluation and possible improvements

The first prototypes provided a valuable proof of concept, that it is possible to build and operate an autonomous submersible in the 50 cm class. The maximum speed of 1 m/s is comparable to larger submersibles. A cruising endurance of approximately 2 hours was found satisfactory, considering that cheap lead acid batteries were used. The same battery volume using lithium polymer cells

would provide 4 times the capacity, which would increase endurance to a usable duration of over 8 hours.

It was found that the limited space inside the hull was sufficient for all the necessary electronics; i.e. the main processor, gyroscopes and accelerometers, compass, radio communication, motor controllers and batteries. Handling proved to be very easy - the vehicles could be launched and retrieved by hand, with no danger to operator or vehicle. The agility in all degrees of freedom of the *Serafina I* is very good, and superior to many larger vehicles. Roll and pitch angular rates can exceed 300 degrees per second (the maximum range of the gyroscope sensors).

The main problem with the initial design turned out to be the thrusters. The use of conventional DC motors, driving the propeller through a sealed shaft coupling, has hydrodynamic disadvantages in the 5-thruster configuration. The vertically mounted thrusters add significant drag in forward direction. Also, the dynamic seals of the drive shafts were prone to leakage after extended use.

2.4 Serafina Mk II: A design study

The main design goal for the successor submarine *Serafina Mk II* was to maintain the dimensions and agility of the first prototype, while making the design more reliable and more efficient. Reliability and hydrodynamic efficiency were mainly limited by the thrusters. To overcome these limitations, a flat rim driven brushless thruster was developed, which can be statically sealed (the only moving part is the propeller, which also forms the rotor of the brushless motor). The flat shape allows the thrusters to be integrated into slim hydrofoils, reducing the overall drag. A photo of the new hull design can be seen in figure 2.6. The overall length of the hull is 50 cm. The total mass of the vehicle will be approximately 5 kg, trimmed to neutral buoyancy.

Advances in microelectronics allow the main CPU to be a 400 MHz RISC processor with low power consumption. The inertial sensors were miniaturised further. The new 6-axis inertial measurement unit including a 2-axis magnetometer is only 5x5x1 cm^3. It is expected that future developments in microelectronics and battery chemistry will also improve performance while reducing the size even further.

Figure 2.6: Serafina Mk II

The outer hull seen in the photo in figure 2.6 is a wet hull, giving the vehicle a hydrodynamic shape and providing mounting points for the thrusters, the internal pressure hull and external sensors. The internal pressure hull is a commercially available cylindrical pressurised vessel, which can be chosen according to target depth rating and budget. A watertight connector provides power and data connections to the thrusters and sensors. The modularity of this design is a great advantage, as it allows the internal pressure hull to be exchanged.

The vehicle contains all sensory equipment required for swarm control and navigation. Additional payload sensors can be installed either inside the front dome of the hydrodynamic hull or mounted externally. A waterproof connector provides an interface and power supply for payload sensors. The measured data can be communicated to the main processor of the submersible and can be used as a parameter in the swarm control algorithm. This allows to implement swarm behaviour such as gradient descent, or isocline adaptation of gradient fields [22].

2.5 Summary

It was shown experimentally by testing a prototype that a submersible robot of 50 cm length and less than 5 kg total mass is capable of similar cruise speeds and endurance as larger submersibles. The available space is sufficient to house all vital electronics and sensory equipment required for swarming applications. The small size provides many advantages with regard to launching and handling vehicles. Small submersibles therefore are preferable over larger vehicles for swarming applications, for reasons of cost, usability and performance.

Chapter 3

Communication channels in water

The underwater environment severely limits communication channels. This chapter gives an overview of what the particular problems are for different modalities and how they affect the range and bandwidth. Optical channels, radio channels and a conductive channel are described in more detail together with experiments. At the end of this chapter, two possible solutions for wireless short range communication are presented.

3.1 Available channels and their limitations

Undersea cable is the most commonly used underwater communication link. The three most common wireless underwater communication channel modalities are radio frequency, optics and acoustics. These different modalities suffer different specific deterioration in water. Channel deterioration also depends on water quality - seawater is different from freshwater or pure water. This section gives an overview of various ways of transmitting information through water, while discussing the technology with regard to use for underwater vehicles.

3.1.1 Wired communication

Wires or cables can carry reliable high bandwidth communication links and are currently the most used solution to underwater communication. Cables consist of wires that either transmit electrical signals in electrical conductors (i.e. copper) or optical signals in optical fibres. Often a specific cable carries a combination of both optical and electrical signals. An additional advantage of cables is their capability of transporting electrical power. In the case of cables connecting remotely controlled vehicles with a mother ship, the cable can also be used as a safety rope to retrieve the vehicle. Virtually all currently used remotely operated vehicles communicate with the operator on the mother ship via a cable. A major problem with umbilical cables on remotely operated vehicles is the weight of a long cable and the associated hydrodynamic drag. Especially in deep sea missions with cable lengths of several kilometres, the drag of the cable becomes a major limitation of the vehicle's manoeuvrability.

Cables are also used as static communication infrastructure to connect continents. A large percentage of intercontinental telephony and data traffic is routed through subsea cables.

In the context of swarms of vehicles it is obvious that wired communication has severe drawbacks. With growing numbers of vehicles the number of cables easily becomes unmanageable, and there is a risk of entanglement. Connecting every vehicle of a swarm to the mother ship is highly unpractical. It is conceivable to use wired communication only between neighbouring vehicles in a group of robots. In this case wired robots would have to move in an almost rigid formation to avoid entanglement. The additional drag and complex control constraints are severe drawbacks, but could be compensated for by the high bandwidth, long range and superior reliability of wired communication.

3.1.2 Electromagnetic channels (radio frequency)

Wireless underwater communication is a challenging task. Most commonly used methods, which are well established for digital communication in air, do not work in water. Available radio modules such as Bluetooth or Wireless LAN (802.11) operate in the gigahertz range, around 2.4 GHz. The attenuation in water for high frequency radio, especially in electrically more conductive salt water, is extremely high. Assuming an average conductivity of seawater of 4 mhos/metre

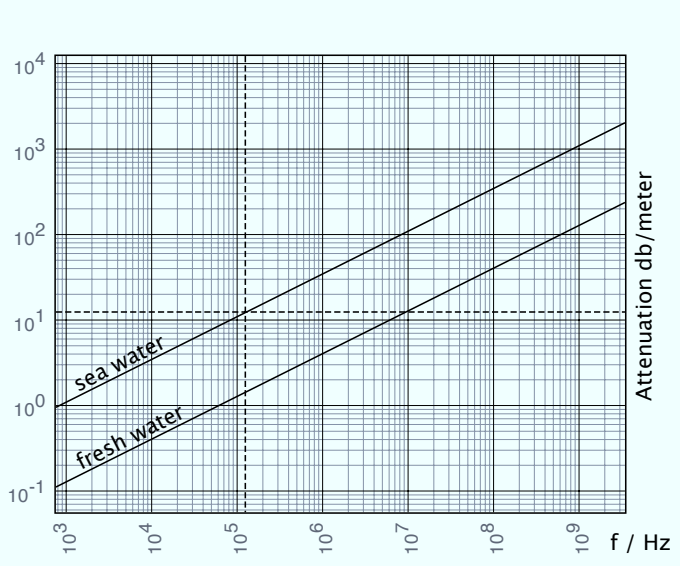

Figure 3.1: Approximate attenuation of electro-magnetic waves in freshwater and seawater

and 0.05 mhos/metre for fresh water (tap water), the attenuation for 2.4 GHz is around 1600 dB/metre in seawater and 190 dB/metre in freshwater. This is clearly not practical.

Plot 3.1 shows an approximate relationship between frequency and attenuation of radio waves in water. A more accurate model for a wider frequency range can be found in [51]. The attenuation of electromagnetic waves occurs multiplicatively per metre of water traveled, which means that the signal strength suffers exponential decay over the total distance from sender to receiver. This severely limits the range. It also means that transmitter power has to increase exponentially to extend the range. There are additional coupling losses on water boundaries at the antennas.

It can be seen that attenuation in plot 3.1 is far more manageable for low frequencies. However, low carrier frequencies put a natural limit on the available data bandwidth. This is additionally complicated by the difficulties of constructing efficient wide band antennas for low frequencies, especially if the available space is limited. As an example, for a carrier frequency of 100 kHz, the wavelength in vacuum is approximately 3 km and about a third of that in

water. For small underwater swarming vehicles external long trailing antennas are not feasible, and the available space for internal antennas is four orders of magnitude less than the wavelength. In general this means poor coupling of the antenna to the medium. Nevertheless, short range communication is still possible, as will be demonstrated later in this chapter. A suitable design for a given application can be defined as an optimisation problem over the frequency, attenuation and antenna efficiency. In reality there are further factors such as frequency dispersion, antenna directivity, interference, the availability of commercially available parts, cost, etc. which play a role. The optimisation of an actual radio link implementation is beyond the scope of this thesis, but a proof of concept prototype has been developed and will be discussed in following sections. A more detailed approach on the design of a low frequency under water radio link is provided in [52].

3.1.3 Optical channels

Optical communication offers high bandwidth, low noise channels and is commonly used in a range of applications. Most common are optical fibre links for large scale network infrastructure. Wireless links in air are usually implemented with focused laser beams to achieve high bandwidth and long ranges. The required power can be kept low if the beam divergence is low. In the ideal case of a perfectly parallel beam in vacuum the received light energy equals the energy emitted at the transmitter, which means that the required transmitter power can be kept constant independent of the link distance. In reality, laser beams have divergence caused by imperfect optics, diffraction and scattering. Light in media other than a vacuum is also subject to attenuation by absorption and scattering. A limitation of free-air laser communication links is that they require precise alignment of transmitter and receiver due to the small beam width. This makes it very challenging to use this technology on moving vehicles. Fast tracking pan-tilt heads can solve this task, but they mechanically complicate the design and also need precise position feedback of the opposite station. A different solution is to use optics to widen the beam, or to use omnidirectional light sources.

There are some existing publications on optical underwater communication. Bales and Chryssostomidis presented a high speed transceiver which achieves up to 10 Mbit/sec and has a long range of up to 20 metres in clear water [2]. This is achieved by using highly specialised and expensive hardware and powerful,

directed light transmitters. A recent publication [55] shows a compact, low-cost optical transceiver for underwater applications, with a range of 2.7 m, using the IrDA physical layer. Being 5 cm in diameter and 10 cm long, their device is too large for miniature submarines of the size of the *Serafina* system, and it has a speed of only 14.4 kbit/sec. Also, the narrow opening angle of their transmitter makes it difficult to achieve omnidirectional coverage despite the 22 LEDs used in the transmitter. It must be noted that these transceivers were designed for much larger submersibles.

Infrared communication according to the IrDA standard (Infrared Data Association, http://www.irda.org) is often used for short range communication and offers reasonable bitrates. Unfortunately water is not transparent for the infrared part of the spectrum, which means that standard IrDA does not work under water. However, the IrDA physical layer modulation can be used and adapted for underwater applications by replacing the infrared light emitting diodes (LEDs) with high power green or blue LEDs and also replacing the photodiode of a type which is sensitive in the visible part of the spectrum.

Solid state light sources have reached a level of maturity that makes it possible to use them for high speed communication. A prime example are light emitting diodes that offer switching speeds faster than 100 ns, allowing for bandwidths of several MHz. In the last few years, a new breed of high power LEDs emerged, capable of handling drive currents up to 1 A and delivering a luminous flux in excess of 100 lumen. In 2006 a dramatic increase in efficacy was achieved, yielding light emitters producing 100 lumen at 1 Watt electrical power (respectively 350 mW light output at 350 mA drive current for blue emitters) [8]. The achieved efficiency is close to one third, putting LEDs on a par with fluorescent and gas discharge lamps. The highest powered commercially available emitters are specified for up to 210 lumen of white light at 1000 mA drive current and can be pulsed to 1.8 A current. This opens up the possibility for optical communication links with omnidirectional coverage over short ranges. Beams can be collimated to 6 degree divergence or less, allowing links to bridge longer distance at lower energy, if the position of the receiver is known.

A proof of concept optical communication module is presented later in this chapter, based on IrDA physical layer modulation for simplicity, and employing efficient high power light emitting diodes.

3.1.4 Acoustic channels

Acoustic channels are currently the most common way of wireless underwater communication. Of all known wireless communication methods they achieve the longest range (up to several kilometres) at reasonable bandwidths up to 5-10 kbps. Over shorter ranges higher data bandwidths up to several hundred thousand bits per second are possible [25][23]. Underwater acoustic modems traditionally use frequency shift keying, but more recently other modulation schemes like quadrature or differential phase shift keying (QPSK and DPSK) became more popular.

Attenuation losses are generally fairly low for low carrier frequencies; i.e. the attenuation is only 0.3 db/m for a 30 kHz carrier [52][26]. This allows for relatively long range communication with acceptable power output. However, the underwater acoustic channel is highly non-linear and suffers from multi-path propagation, dispersion, scattering, reverberation, frequency fading and many other effects. The speed of sound in water depends on frequency, water pressure and density (temperature and salinity) [38]. Reverberation and multi-path propagation can be caused by reflections from obstacles. Due to the varying speed of sound refraction can occur, which leads to multi-path propagation even without the presence of obstacles. All these effects cause a discrepancy between the usable range and the distance that the transmitted signal propagates. This means that a transmitter can cause interference at distances far greater than the range at which the signal can be decoded. In the case of networks with many participants, this can be a severe limitation to the overall bandwidth and channel utilisation. The rather slow speed of sound additionally creates large propagation delays, which makes node synchronisation challenging. Generally acoustic communication does not scale well for large scale networks such as robotic swarms [13][52].

A possible solution for acoustic communication in swarms is to use very high carrier frequencies. The attenuation in water increases with frequency due to absorption losses (heat losses) [38]. Carrier frequencies around 1 MHz do not cause interference over large distances, as the signal is attenuated sufficiently. Over short distances channel distortions are not as severe, so it is easier to decode the signal successfully. For swarm communication an acoustic link is ideally limited by the received signal strength (if the signal can be detected, it can be decoded). The carrier frequency, receiver sensitivity and output power should be

chosen to match the range of usable communication and the range of detectable interference. As there is a large body of current research on acoustic underwater communication, this thesis does not address this approach. Note that many of the results of chapters 4 and 5 may apply as well to acoustic links with the described property of low interference beyond the usable range.

3.1.5 Electric fields

Acoustics and electromagnetic waves are not the only methods of wireless underwater communication. One other possibility is using electric fields and return current density, as presented in [46][56][36]. Both the transmitter and receiver consist of two electrodes exposed to water. The transmitter electrodes are connected by an insulated cable to a power amplifier; the receiver electrodes are connected to a high gain preamplifier. A modulated voltage is applied to the transmitter electrodes. Seawater is a conductor; hence the electric field between the two transmitter electrodes causes a return current to flow through water around the transmitter. According to the cited articles, the current spreads out around the transmitter to distances far greater than the distance between the transmitting electrodes. A sensitive receiver can measure this current at a distance by amplifying the voltage drop between two points, and it can pick up modulations of this current. The paper [46] reports on a system for scuba divers, in which the electrodes were attached to the ankle and chest pack of the diver. Ranges of up to 100 m in 30 m deep water could be achieved. The literature is very sparse on this topic - the only papers reporting on this effect are from the 1970s [46][56][36], namely the brief theoretical account given by Schultz in 1971, and the little-known paper by Momma and Tsuchiya, who experimentally investigated the phenomenon at the Japan Marine Science and Technology Center in 1976. The paper [56] by Walther merely mentions electric communication and cites the paper by Schultz; it can therefore be omitted from discussion. Notably, all publications only refer to return current communication in seawater. No results on freshwater are known. The reasons why this technology was abandoned are unknown. One might speculate that it could not compete with acoustic communication in terms of range and power requirements, and that at the time, the scalability problem of acoustic communication was not an issue.

There are currently no commercially available systems using electric fields for communication, and there are no recent publications on the effect. Doubts

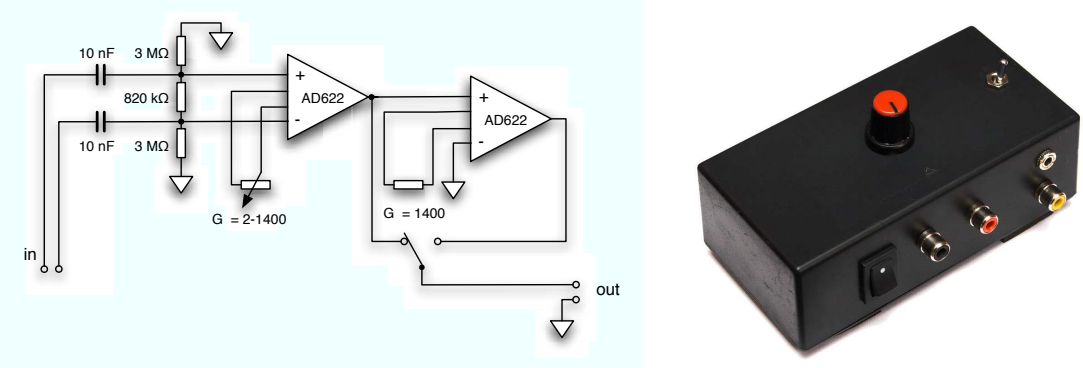

Figure 3.2: The two stage high gain instrumentation amplifier used in the initial return current experiment.

towards the reality of the phenomenon arose, as the reasons why this effect gets so little attention are unknown. It was suspected that there might be a general problem with this method of communication, which was never published.

As it could not be reliably determined from literature if this effect is in fact real, and as barely any data of the current densities measured in the field are available, a proof-of-concept experiment was conducted. A two stage high gain instrumentation amplifier was designed and built, based on two AD622 integrated precision instrumentation amplifiers by Analog Devices (fig. 3.2). The first stage has two high impedance differential inputs. The gain is adjustable via a potentiometer between factor 2 to 1400. The second stage has a fixed gain of 1400. The output can be switched between the intermediate output of the first stage or the final output of the second stage. Therefore the gain can be chosen from a range from 2 to over 1,000,000 (excluding the range from 1400 to 2800). At gains above 50,000 the amplifier saturates very easily due to sub-microvolt input voltages. The usable gain range of the amplifier is roughly 2 - 50,000. To avoid clipping of the sensitive amplifier, the sensing electrodes are decoupled using a 10nF capacitor in series. A 820 kΩ resistor connects the amplifier inputs to remove DC offsets. Shielded cable connects the two sensing electrodes to the inputs of the amplifier. The shield is connected to earth.

The transmitter consists of an audio amplifier and a signal source. The outputs of the audio amplifier are connected to two electrodes that are submerged in water. To exclude the possibility of unidentified signal paths, both the transmitter

and the receiver are battery powered and not connected to any other equipment, ground or external power supplies.

A first proof-of-concept experiment was conducted in a 15 cm x 15 cm x 5 cm water container. The container was filled with tap water, and 5 g of table salt (sodium chloride) were added. The transmitter electrodes were formed by a 3.5 mm audio jack plug connected to a battery powered audio player. The distance between the transmitting electrodes was approximately 3 mm. The receiving electrodes were spaced 5 cm apart and submerged on the opposite side of the container (15 cm away from the transmitter). A broadband audio signal (30 Hz - 15 kHz) was sent through the transmitter. The signal could be clearly received on the receiving side, with little distortion. A frequency sweep revealed fairly consistent amplitude for frequencies from 50 Hz to 50 kHz. Higher frequencies could not be reliably measured due to bandwidth limitations of the instrumentation amplifier. The signal could still be picked up after the receiving electrodes were removed from the water but kept in the vicinity of the container (within 20 cm). The signal strength dropped with increasing distance, both within water and out of the water. Outside the water the received signal was slightly weaker than within the water, and low frequencies suffered high attenuation. When the transmitter electrodes were removed from the water, the signal was lost. A possible hypothesis for this effect is that the current flowing through the water container around the sending electrodes formed a loop antenna, and the emitted electromagnetic signal could be picked up by the dipole that the electrodes formed while not submerged. While the electrodes were submerged the conducting salt water added significant resistive losses to the dipole, however, the signal strength and the low frequency response increased significantly. This is a strong indication that the signal received while submerged is not electromagnetic radiation, but is due to the voltage drop caused by the return current field. Further indications were that the signal strength dropped to zero when the line through the transmitting electrodes intersected perpendicularly in the center of the line between the receiving electrodes, and the phase is inverted on either side of the line through the transmitter.

The experiment was repeated in a larger container, 60 cm x 40 cm x 20 cm, filled with unmodified tap water. The transmitting electrodes were replaced by two separate electrodes. It was found that the received signal strength dropped with the distance between transmitter and receiver and rose with increasing distance

between the transmitting electrodes. The signal could still be picked up clearly over a distance of 50 cm, with 1 cm separation between the transmitter electrodes. In contrary to the few existing publications listed above, this experiment could show that the return current effect also occurs in freshwater. The effect was strong and convincing enough to justify a more precise study of the effect. The results are shown below in section 3.2.

3.1.6 Other channels

It is known that some species of fish use similar methods of electric conduction communication in their lateral line organs [5]. The electric eel emits electric pulses for hunting and sensing, and is also able to measure electric fields [24]. The lateral line organ also plays an important role in the swarming behaviour of some fish species, mainly for monitoring the speed and direction of travel of neighbouring fish [37] - however, in this case the lateral line organ more likely responds to pressure waves.

A communication method that is often overlooked is direct contact. While it is not a digital communication method expressed in bits per second, physical contact between members of a group can convey valuable information about the other member's position and velocity and possibly other state information.

If one defines communication as the process of moving information through physical space, then physical movement of a data storage unit (i.e. a robot) is also communication. In the concrete case of a robotic swarm, individual robots can contain large amounts (several gigabytes) of stored information. Long range communication links under water typically have a very low bandwidth; transmitting large volumes of data is not feasible. However, as the robot is mobile, it can simply move (or "communicate") this information to a new physical position by moving itself through space. There are high bandwidth short range links like optical communication or wired communication. Once the robot reaches the desired target region, the information can be transferred to other robots over short range links very quickly. The overall bandwidth of physical motion and short range links can easily outperform long range links in an underwater setting. There is a trade-off between the amount of information that has to be exchanged, the distance to be covered, the amount of energy available for the communication, the bandwidth of the available channels and other factors. In certain scenarios dedicated data courier robots are imaginable -

robots that move quickly between other robots and exchange information using short range high speed links.

Information can also be exchanged through the environment. This is called stigmergy; a classical example is ants leaving pheromones on the ground to signal their current behaviour, or the work of termites being guided by the state of the construction that they observe [19]. Stigmergy can also be used by multi-robot systems. Robots can actively modify their environment and also influence their environment by their presence (embodiment). Other robots can observe these changes and adjust their behaviour accordingly; this form of indirect communication can be used to control the behaviour of a multi-robot system [32]. Purely local communication via short range links or direct contact can also be regarded as (virtual) stigmergetic communication.

One last communication channel that should be mentioned is chemical communication. While not as relevant to robotics as the other modalities, mostly due to a lack of suitable sensors, it is an important channel in biological systems.

3.1.7 Summary

All communication channels mentioned above have particular advantages and disadvantages, and none of them fulfills all possible requirements of a robotic swarming system. Generally communication is very limited, and information is a valuable resource for swarm individuals. It is advisable to extract as much information from any received signal as possible. As an example, an acoustic signal may contain a digital message coded into the acoustic signal. Additional information conveyed by the exact same transmission is the direction from which it was received (revealing information about the sender's position), Doppler shift (revealing relative velocity), reverberation and echos (representing and possibly revealing information about nearby objects, the seafloor, etc.) and much more. It is not always possible to tap into this rich source of information. Additional hardware might be required, driving up the cost and complexity of a system. However, where some of this information can be extracted with acceptable effort, it should be attempted, even if the information is unreliable or imprecise. It is of more value to include data uncertainty and reliability into the representation of sensor information than to discard the data altogether.

Furthermore, it is useful to build systems that use more than only one particular modality or communication channel for redundancy and better coverage of

varying situations. Apart from the obvious benefit of having several communication channels to choose from, there are synergetic effects which vastly improve the richness of information available to a system. If radio and acoustic communication are combined, the difference in time of flight between the two signals can be used to estimate the link distance. This assumes that the two messages are sent out with a known temporal relationship (i.e. at the same time). If the time of flight information is combined with the content of the actual message and with the content of other messages received before or after, and if every node includes time-of-flight information into messages, the accumulated information can be used to triangulate the position of each node in the neighbourhood.

3.2 Measuring the return current field

The previous section introduced return current fields as a possible underwater communication channel. The preliminary experiments showed the existence of the effect and gave an indication of the properties and nature of the return current field. This section quantifies the effect by densely measuring the field strength and direction. The experiments concentrate on freshwater, as no experimental results could be found; the only publication giving experimental results is [36] by H. Momma and T. Tsuchiya and only reports on return current in seawater.

3.2.1 Method

The following setup has been chosen for the experiment. A transmitter consisting of two electrodes mounted at distance d_t is fixed submerged in water and moved to various positions. As before the transmitter electrodes are connected to the outputs of a power audio amplifier. The receiver comprises two pairs of electrodes in a plane, with the lines between the electrodes intersecting in the center at an angle of 90 degrees. The electrode pairs are connected to a high pass filter followed by the differential inputs of high gain instrumentation amplifiers. The measured signal is sampled by a 10 bit analog to digital converter and analysed by a microcontroller (figure 3.4). The arrangement of receiver electrodes makes it possible to measure the two orthogonal components of the return current field projected to the receiver plane. The two-dimensional field

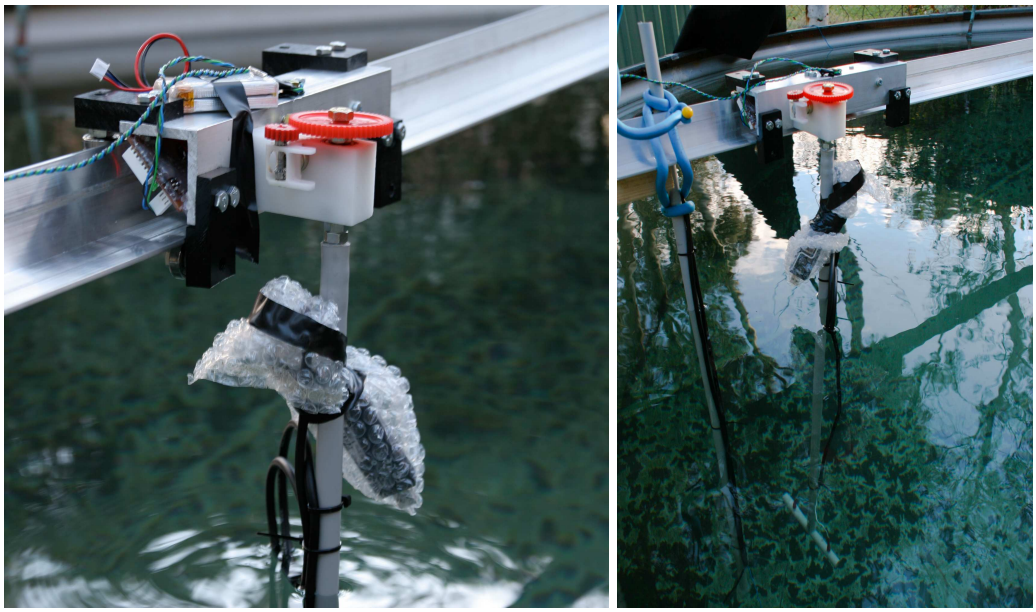

Figure 3.3: The gantry robot system that was built for measuring the return current field. The thin green cable is the serial communication link to the robot's motor controller. There is no electrical connection between the signal transmitter (wrapped in plastic for protection) and the rest of the gantry.

vector can easily be reconstructed; the direction of the vector indicates the current flow direction, the length indicates the field strength.

As the inputs are high-pass filtered and the transmitted signal is AC (alternating current), a RMS reading is calculated. The field direction is estimated from the dominating direction, as the two differential signals are assumed to be in phase (this assumption was established in previous experiments). To ensure accurate absolute readings, the receiver is carefully calibrated with known voltage sources.

A dense field measurement requires thousands of sample points. To make obtaining the data points possible, a robotic gantry system was designed and built. The receiver assembly is kept stationary, while the transmitter can be moved along a rail and rotated around its own axis. It is reasonable to assume that in the absence of nearby obstacles this method of measuring is equivalent to rotating the receiver around the centerpoint of the transmitter.

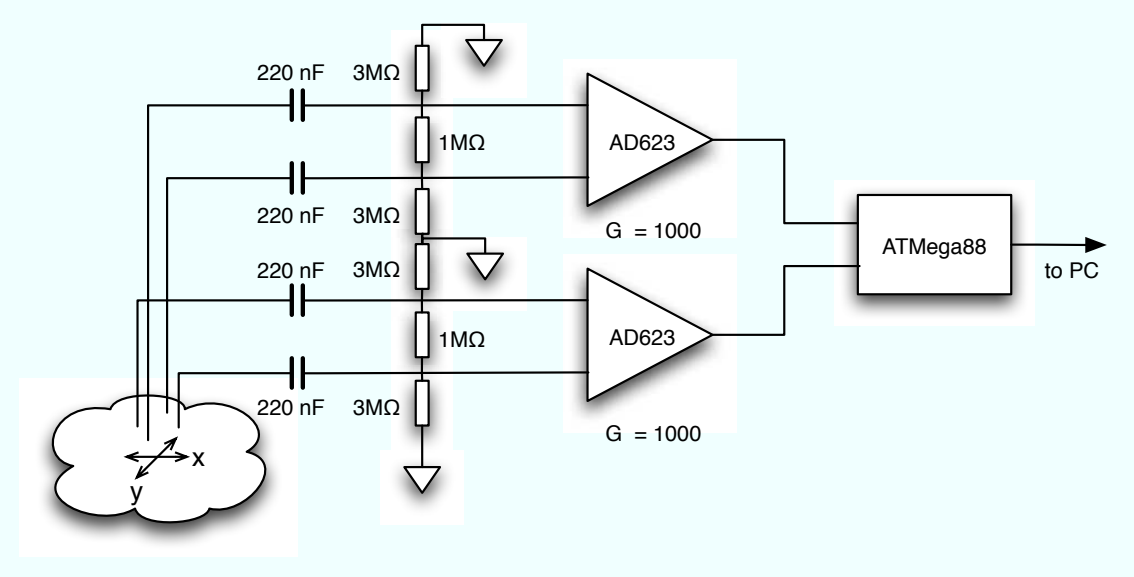

Figure 3.4: Schematic of the 2-channel return current amplifier

The robotic gantry has a 2 metre long aluminium rail. An actuated carriage glides along the rail, driven by a geared DC servo motor via a rubberised friction wheel. The motor encoder makes it possible to keep track of the carriage's position and velocity. A rotating joint with a second servo motor is mounted on the carriage. The rotating shaft of this joint carries the transmitter. For ease of use and reduced electrical interference, the carriage and rotation joint are battery powered and equipped with on board processing capabilities. Positioning commands are transmitted via a serial connection. The on board processor performs feedback control on the position along the rail and in the rotational axis, with bounded velocity and acceleration.

A host computer sends positioning commands to the gantry carriage and stores the corresponding field strength and direction as sampled by the receiver processor. The complete system can autonomously sample the entire field with adjustable resolution and high accuracy. The implementation of the gantry robot can be seen in figure 3.3.

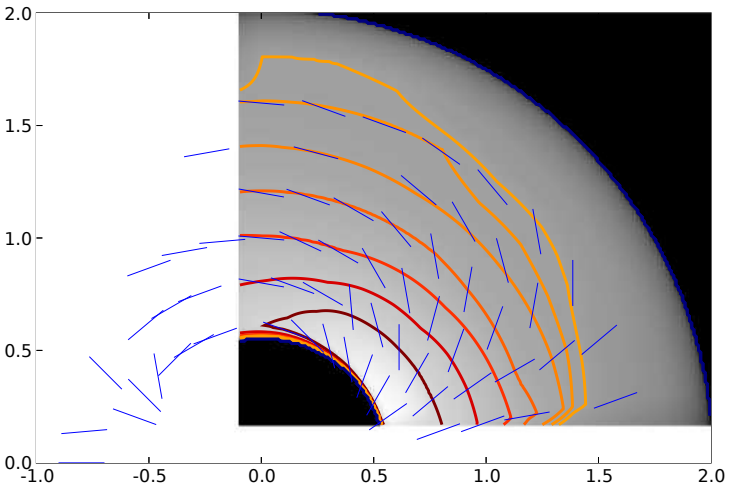

Figure 3.5: Manually measured return current field. Measurements were taken at the centers of the blue lines; the direction of the blue lines indicate the field direction. The grey background shows a linearly interpolated reconstruction of the field strength, the contour lines show an approximation of equal strength.

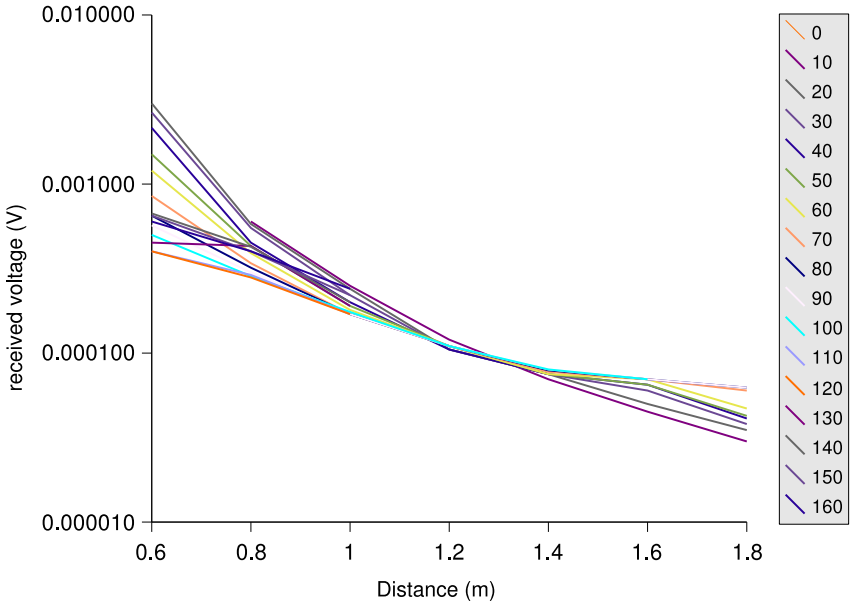

Figure 3.6: Voltage levels measured at different angles, in Volt

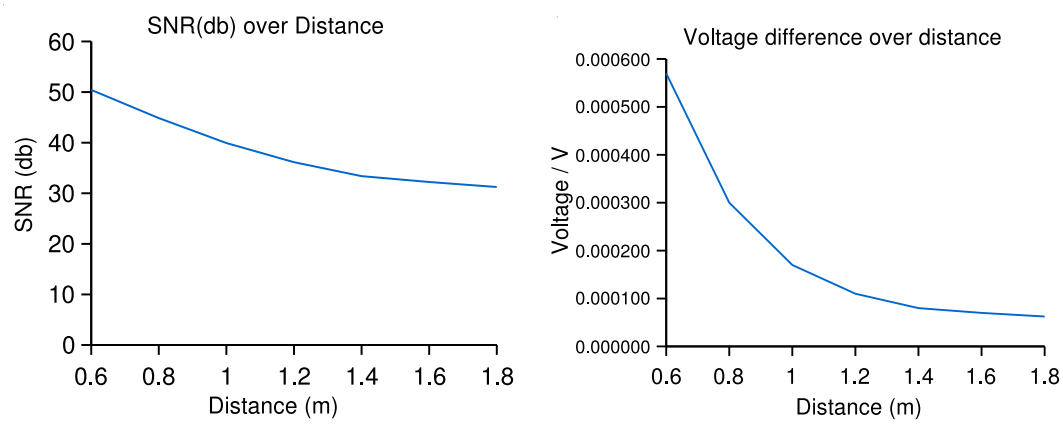

Figure 3.7: *Left:* Signal to noise ratio over distance for a 1 kHz sine wave with 2 V amplitude. Transmitter electrode distance was 10 cm. *Right:* Measured voltage over distance at 90 degrees angle. Noise was measured at 0.6 μV

3.2.2 Results

The first experiment was conducted manually prior to construction of the robotic gantry system.

Figure 3.5 shows the measured field. The transmitter electrodes were spaced 10 cm apart. A DC resistance of 11 kΩ was measured between the submerged transmitter electrodes in the test tank. The water temperature was 12 degrees Celsius. The transmitted test signal was a 1 kHz sine wave with 2 V peak amplitude. The receiver electrodes were separated by 15 cm. The receiver only had one channel; to obtain the angle of the field, the receiver assembly was rotated until the maximum field strength was achieved.

It can be seen that the return current field spreads out over distances which are significantly larger than the electrode distance. The direction of the field is as one would expect.

The received signals are very small, but nevertheless significant over the noise floor. The measured background noise of the test setup with the transmitter switched off was 0.6 μV. Figure 3.7 (left) shows the signal to noise ratio of the received signal over the noise floor. The plot on the right in figure 3.7 shows the received voltages of the same experiments for comparison. Both measurements were taken at 90 degrees angle.

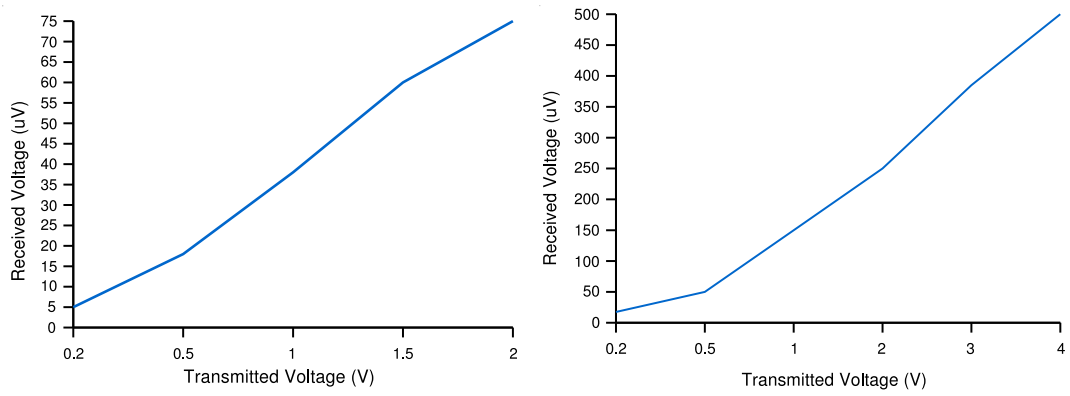

Figure 3.8: *Left:* Received voltage over transmitted voltage for 10 cm transmitter electrode spacing *Right:* Received voltage over transmitted voltage for 30 cm transmitter electrode spacing

The received voltage is directly proportional to the voltage across the transmitter, as can be seen in figure 3.8. Furthermore, the received voltage for 30 cm electrode distance is roughly three times larger than the received voltage for 10 cm transmitter electrode spacing. This suggests that the received voltage is also proportional to the transmitter electrode spacing. It was experimentally checked that the received voltage is approximately proportional to the receiver electrode distance (transmitter-receiver distance 1.2 m, measured for receiver electrode distances 0.1 m, 0.2 m, 0.3 m at 90 degrees). This approximation holds if the electrode distances are significantly smaller than the distance between transmitter and receiver (i.e. the field curvature between the receiver electrodes is small). The voltage drop between the receiver electrodes should be given by the local return current multiplied by the DC resistance across the electrodes. Therefore, the measured voltage is expected to be proportional to the receiver electrode distance for a given return current.

The relationship between distance and received voltage can be approximated with a $c \cdot 1/d^2$ model. Figure 3.9 shows measurements in relation to the adapted model. For the given electrode distances and drive voltages an approximate matching model is

$$V_r \approx \frac{2 \cdot 10^{-4} m^2}{d^2} V \tag{3.1}$$

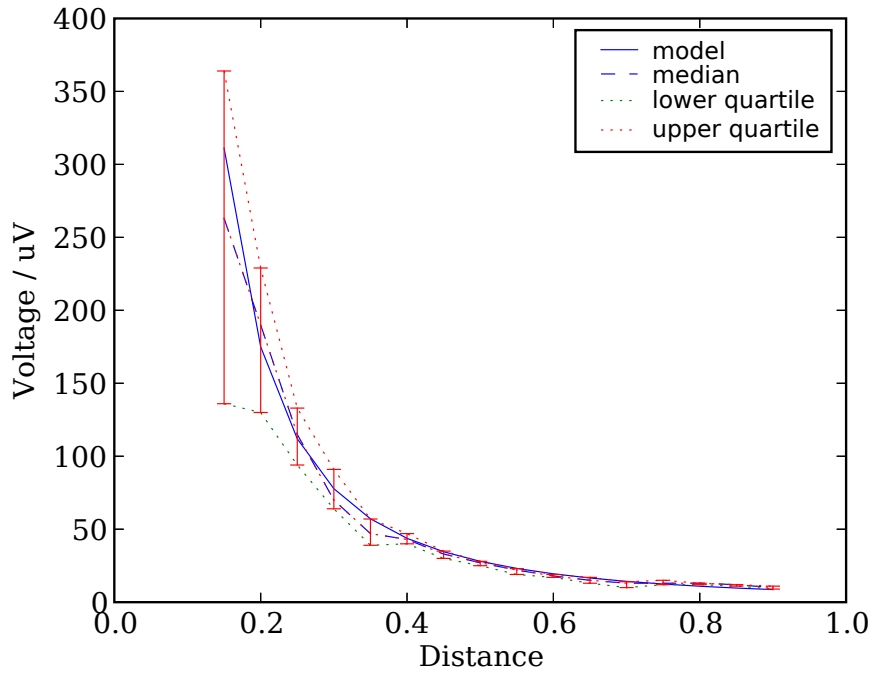

Figure 3.9: Comparison between measured voltages and a simple c/d^2 model.

where d is the distance between the centers of the transmitter and the receiver.

After the manual experiment it was realised that a higher resolution and better accuracy was required to fully capture the properties of the return current field. The previously described robotic gantry was set up in the test tank. In order to minimize interference with the limited size of the pool, it was decided to scale the experiment down by using a lower drive voltage and reduced electrode distances. Both transmitter and receiver electrodes had a spacing of 10 cm. The transmitted test signal was a 1 kHz sine wave with 200 mV peak amplitude. Great care was taken to avoid unwanted signal transmission other than the return current. The transmitter was mounted on the mobile gantry carriage with an insulated plastic tube. The signal was generated by a small battery-powered signal generator that had no electric connection to any other part of the setup. Shielded waterproof cable connected the signal generator to the transmitter electrodes.

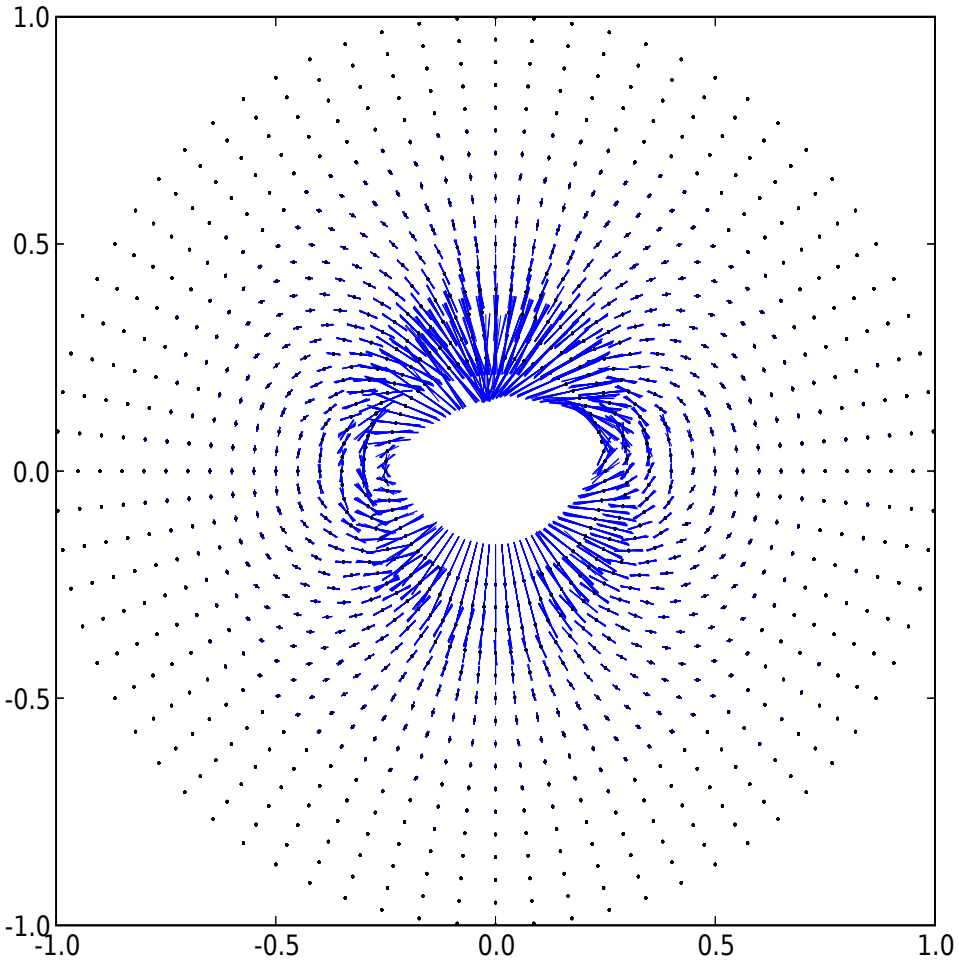

Figure 3.10: A dense measurement of the return current field. The blue lines visualize the RMS signal strength measured along the x and the y axis. The length of the vector is proportional to the measured RMS signal strength, the orientation of the vector shows the direction of the field.

The receiver was fixed at one end of the gantry rail. The spacing between the transmitter and receiver was varied from 0.2 m to 1 m. Longer distances were not required due to the low signal strength; no signal could be detected at distances beyond 1 m.

The gantry robot was programmed to take samples at distances from 0.2 m to 1 m at 5 cm increments and from 0 to 360 degrees in 5 degree increments. For each position the robot took ten measurements of the field within one second. The entire experiment was conducted fully automatically without human intervention. The obtained data can be seen in figure 3.10. Figure 3.11 gives a closer view of the field vectors. For better visibility the measurements for 20 cm and 25 cm have been removed.

The new measurements are consistent with the manually taken measurements from figure 3.5. It can be seen that in the near field up to 1 m the field is stronger along the axis of the transmitter electrodes. For larger distances the field strength is almost independent of the angle between the transmitter and receiver. The shape of the field follows the classical two-sphere shape that one would expect from an electric field which is emitted from two opposing point sources.

3.2.3 Discussion

The results of the experiment show that the return current phenomenon described in the literature [36, 46, 56] does indeed exist. The field strength was found to be approximately inversely proportional to the square of the distance between transmitter and receiver, and proportional to the voltage across the transmitter electrodes. A full study of the effect and a comparison to the results found in the described publications is beyond the scope of this thesis and shall be regarded as future work. Initial checks, however, indicate that the models described in the literature are quite accurate.

Return current appears to be a possible way of communication in swarms of small submersibles. If one assumes that the field properties scale accordingly to the distance and transmitter voltage relationships, it is possible to derive an approximate figure for a realistic communication range. A starting point is given by equation 3.2.2. The experiment from which this model was derived used 2 V across the transmitter. The electrode spacing was 10 cm for the transmitter and 15 cm for the receiver. If we incorporate the linear relationship between transmitted and received voltage, we get

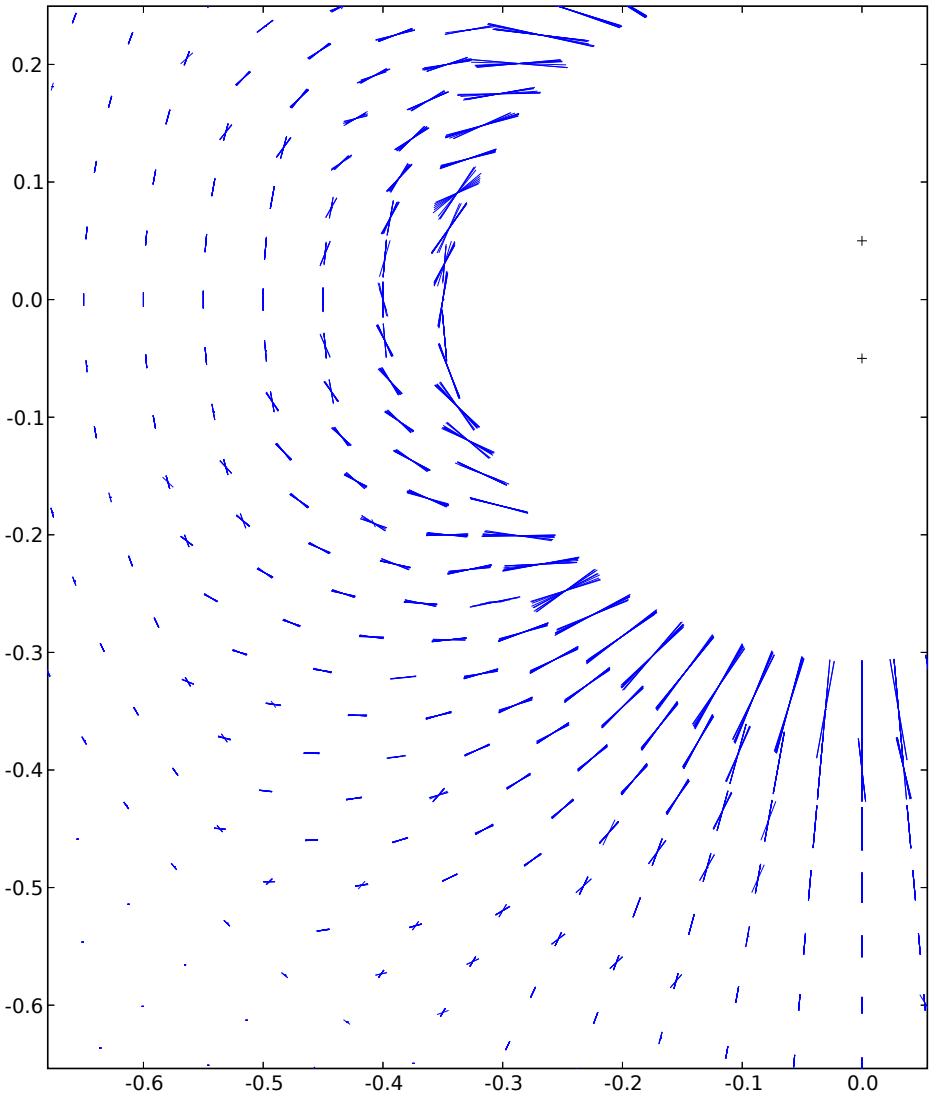

Figure 3.11: A closer view on the plot from figure 3.10. The markers in the center indicate the position of the transmitter electrodes.

$$V_r \approx \frac{2 \cdot 10^{-4} m^2}{2d^2} V_t \tag{3.2}$$

(V_r is the received voltage, V_t is the voltage across the transmitter). Adding the linear relationship between the received voltage and the distance between the transmitter and receiver electrodes leads to the following approximate model:

$$V_r \approx d_r d_t V_t \frac{2 \cdot 10^{-4}}{0.1 \cdot 0.15 \cdot 2d^2} \tag{3.3}$$

$$V_r \approx d_r d_t V_t \frac{6.667 \cdot 10^{-3}}{d^2} \tag{3.4}$$

Realistic numbers for a submersible robot such as the *Serafina* are $V_t = 10V$ and $d_t = d_r = 0.5m$. With the given receiver the noise floor is at 0.6 μV. This means that a received signal of $V_r = 10\mu V$ would give a sufficient signal to noise ratio of about 15 dB. Therefore, the maximum range would be

$$d \approx \sqrt{d_r d_t V_t \frac{6.667 \cdot 10^{-3}}{V_r}} \tag{3.5}$$

$$\approx 40\, m \tag{3.6}$$

These approximations are only meant to give a first indication of what range can be expected. Signals were successfully transmitted over distances of 20 m and more in trials with 5 V amplitude and $d_t = d_r = 0.5\, m$. A natural lake environment was found to be virtually free of noise. However, it was noticed that nearby street lamps caused a detectable 50 Hz hum in the water when they were switched on.

The required power to achieve this range would depend on the conductivity of the water. In the fresh water used in the above experiments a DC resistance of $11\, k\Omega$ was measured across the transmitter electrodes. A rough approximation for the required transmitter power with 10 cm spacing and 10 V drive (assuming only resistive losses) is therefore $P = U \cdot I = U^2/R \approx 9.1\, mW$ for DC voltage. If AC signals are used, the power requirements may vary due to capacitive effects. In seawater the resistance would be much lower, which increases the required power. However, the increased conductivity also reduces the voltage drop per current unit on the receiver electrodes; therefore, the power to range ratio would

be worse. Exact quantisation of the range and power requirements in various media would require more experiments in the future.

One concern regarding communication using electric currents is the effect on marine life and human operators. Some experiments regarding the physiological effects of electric current in water on humans and sharks are published in [49]. Critical levels for humans are reported to be 4-16 V/m. The voltages used for the shark barrier described in the publication are 10,000 - 30,000 V, and the electrode distance can be several hundred metres. Clearly the voltage levels and the field strength in this application are much higher than what is required for communication. If the transmitter voltage is chosen to be 10 V as described before, the field strength in very close proximity to the transmitter can exceed the critical level of 4-16 V/m, but only on the short distance between the transmitter electrodes. The voltage differential between any two points cannot exceed 10 V. At 1 m distance the field strength (using above approximate formula) is only 33.3 mV/m. It is unlikely that such a low voltage would have any noticable harmful effect on a human. It is also described in [49] that electrotaxis (which is a electrically induced physiological response) in sharks of a total length of 0.75 m requires field strengths in excess of 6 V/m. A fright response can be observed at lower field strengths. The paper does not report any lasting effects at the mentioned electric field strengths.

It has been shown that it is possible to transmit signals in fresh water over distances of tens of meters using electric return current fields. Previous work could show communication using return current in sea water. This thesis contributes by demonstrating return current communication in fresh water, and by publishing dense measurements of the direction and strength of the field. Sine waves up to 100 kHz were successfully transmitted over several meters. It is likely that the achievable bandwidth outperforms underwater radio communication. However, unfortunately the return current effect was only rediscovered by the author two months before submission of this thesis; therefore, the implementation of a return current communication module and further experiments will be subject of future work and cannot be included here.

A great advantage of a return current communcation system is its simplicity. The implementation requires only a few relatively cheap parts; the space requirements are also minimal. It is easy to use multiple sensing electrodes which make it possible to sense not only the strength but also the precise direction of the field.

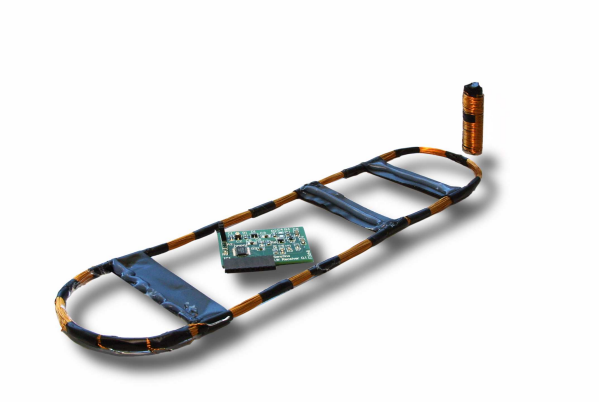

Figure 3.12: The long-wave transceiver module, together with the transmitter and receiver antennas

Once in place, such a communication system can also be used for other purposes, by taking advantage of the highly sensitive electric current sensors. An example is electrolocation as described in [50]. It might also be possible to use the sensors to detect electrical emissions of weakly electric fish for research in marine biology.

3.3 A digital long-wave radio module

It has been shown earlier in this chapter that short range communication using long-wave radio is feasible. In order to identify the capabilities of such a radio link, a proof of concept prototype was designed, built and tested. This section gives an overview of the design and measurements.

The most obvious design decision is the choice of carrier frequency. It should be high enough to offer sufficient data bandwidth, but low enough not to be affected too much by attenuation. Further factors are the availability of integrated circuits. For one-off prototypes or small production runs of less than several ten thousands of units it is not feasible to do custom designs of application-specific integrated circuits. A small low frequency receiver chip is available from Dynatron [11]. It is capable of operating at 8 different carrier frequencies from 8192 Hz to 122.88 kHz and offers data bit rates of 1024, 2048, 4096 or 8192 bit per second. The chip employs differential binary phase keying for digital modulation. It was decided to use this receiver at 122.88 kHz.

The modulation and generation of the carrier wave is done in software on a RISC microcontroller, using a number of hardware timers. The microcontroller runs at 9 MHz and offers 32 kilobytes of flash memory and 2 kilobytes of RAM. One of the hardware timers is used to generate the 122.88 kHz carrier wave. The carrier is modulated by a second output of the microcontroller. If the data modulation output is high, the carrier is inverted, otherwise it is left untouched. The microcontroller pre-calculates every data frame and then activates an interrupt driven routine. This routine programs a timer to switch the data modulation output pin to the correct polarity, exactly synchronised with the carrier. Every time the data modulation output pin changes its state, the phase of the carrier is inverted. The generated modulated carrier signal (a squarewave) is amplified and drives a tuned resonance circuit, which acts as the transmitting antenna. The antenna itself is a planar air coil (or a multiple loop inductive antenna) with 100 turns, of approximately 10 cm^2 area.

The receiver uses a second antenna, which is a ferrite core multiple loop antenna with 300 turns. To increase gain and improve narrow band sensitivity, the receiver antenna forms a tuned L-C oscillator together with an adjustable capacitor. The signal is preamplified in a two stage class-A preamplifier and decoded by the long-wave decoder chip. The long-wave decoder chip has a built in preamplifier with a gain of 2000. After preamplification, the decoder chip digitally samples the incoming signal and decodes it. The decoded data is passed on through an SPI interface to the same microcontroller that implements the transmitter. The complete modules are both transmitter and receiver.

The actual data communication is packet-based. A packet can be sent in one time slot and consists of multiple frames, which can be up to 16 bytes long. Frames are identified by an 11 bit frame header, which can be arbitrarily chosen. The receiver automatically rejects frames with invalid headers. The signal strength of the incoming signal can be monitored in two different ways. The receiver has a built-in comparator and the ability to count the number of peaks that exceed a preset threshold. This can be used to detect close proximity (less than 50 cm) of a transmitter. Secondly, the average amplitude as measured by the receiver's internal analog-to-digital converter can be retrieved to give a more accurate estimate for the received signal strength.

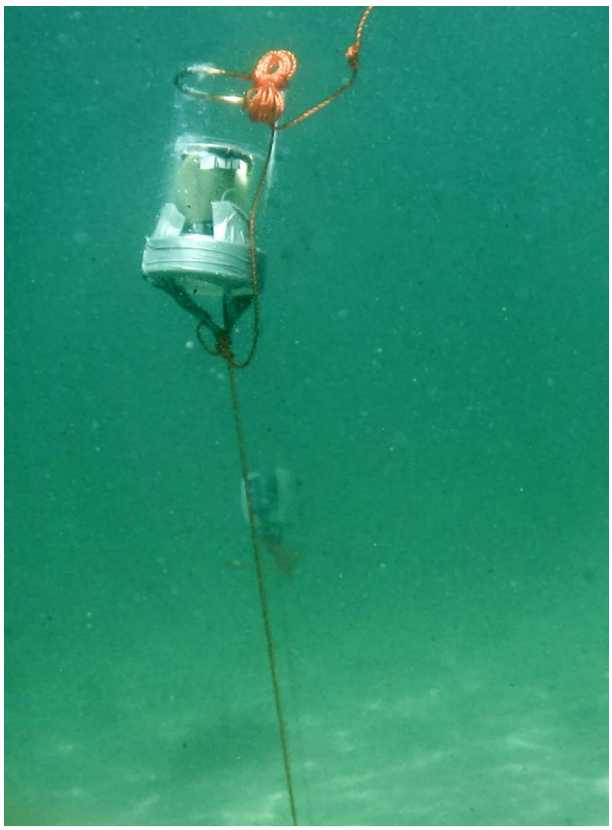

Figure 3.13: The long-wave transceiver module during a range measurement experiment in the Pacific. A second module can be seen in the background.

While not transmitting, the module consumes less than 20 mA current at 5 V. The output amplifier operates currently at 5 V output voltage amplitude, consuming 20 mA. This corresponds to 100 mW power.

3.3.1 Range

Obviously range measurements always depend on many parameters (output power, antenna efficiency and tuning, noise) and the environment. The following measurements are only examples to provide design guidelines to meet application requirements. The definition of range is the maximum distance between transmitter and receiver that provides less than 10% frame dropout. It should be noted that for this particular implementation the increase in frame drops is

Medium	Conditions	Bit rate	Range
Air	inside building (5 V)	4096 bit/s	4.15 m
Fresh Water	swimming pool, 1.5 m down (5V)	4096 bit/s	4 m
Salt Water (Pacific)	coastal water, 3 m down (5 V)	1024 bit/s	3.9 m
Air	inside building (5 V)	1024 bit/s	5.4 m
Air	inside building (14 V)	1024 bit/s	16.5 m
Air	outside, 1 m above ground (8 V)	1024 bit/s	12-14 m
Air	outside, 1 m above ground (12.5 V)	1024 bit/s	15-17 m

Figure 3.14: Range measurement results in various conditions, using drive voltages between 5 V and 14 V (in brackets)

very sharp - typically an increase in distance of less than 0.3 metres corresponds to a transition from virtually error-free reception to total loss of reception. That means that within the given range, reception can be assumed to be quite reliable. The following experiments were carried out with a 0.1 metre diameter circular transmitting antenna and a 10 mm diameter 50 mm length ferrite core receiver antenna. The drive voltage amplitude for the transmitter was either 5 V, 8 V, 12.5 V or 14 V. The transmitting antenna has a high gain resonance circuit. Properly tuned, the voltage across the transmission coil is 15-20 times the drive voltage (approximately 100 V for 5 V drive and 300 V for 14 V drive). Refer to figure 3.14 for the results.

It can be seen that the effect of water is minimal. Even strongly conductive seawater has only a small impact on the range. The actual range might seem short, but it lies well within our expectations. For the final application, the drive voltage will be increased to 20 V. A communication range of more than 10 metres is sufficient for a school of miniature submersibles which are only 50 cm long.

It has to be noted that the loop antennas used in the transceiver are polarisation sensitive. The receiver antennas were wound over the entire length of the ferrite core, which reduces directivity. It was found in the experiments that polarisation does have an effect and can reduce the range. The effect was found to be marginal if transmitting and receiving antennas are mounted along a vertical axis, and if the communication nodes were close to the same horizontal plane.

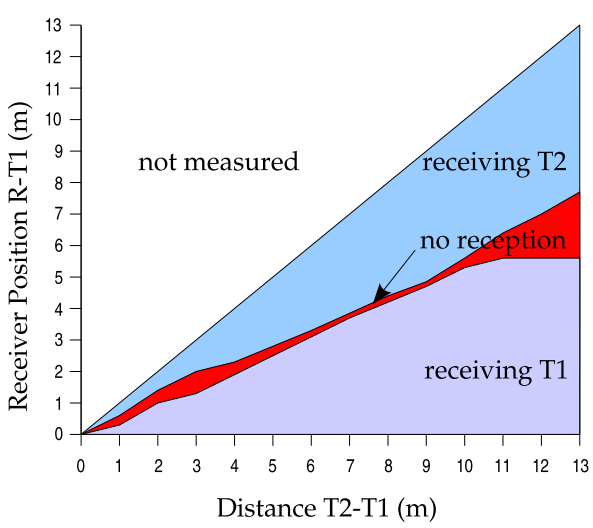

Figure 3.15: Reception zones and collision zone

3.3.2 Interference measurements

An important question is how transmitters interfere with each other, and how receivers experience message collisions. For this experiment two almost identical transmitters T1 and T2 were placed with varying distance from each other in one metre increments. Both transmitters have approximately 5 m range and are continuously sending out frames containing their own identifier. A receiver was placed on the straight line connecting the two transmitters. The receiver was then moved on this line to determine the boundaries of the region where the receiver reliably receives T1 or T2, or has no reliable reception. The results are shown in figure 3.15.

It may be surprising to see that the zone in which collisions occur is actually quite narrow. The theoretical network model which is commonly used in literature defines that a receiver receives a message if and only if exactly one transmitter within its range sends. Such a model therefore would have predicted collisions for the entire regions where the ranges overlap. In reality, the receiver can easily decode the respectively stronger signal of the closer transmitter. Reception is only distorted in a narrow zone in the middle between the transmitters. This behaviour can easily be explained by considering the data modulation for the long-wave radio modules used in the experiment. The transceivers employ a

phase modulation scheme to encode the binary data. The receiver locks on to the phase of the received signal and decodes the phase shifts (usually this is implemented in a phase locked loop). Since the receiver can only lock onto a single frequency it will always prefer the strongest signal and discard any other signals as noise. It can be expected that the same behaviour will be observed for other phase modulation and frequency modulation schemes, which employ a phase locked loop in the receiver.

In literature on wireless multi-hop networks it is often assumed that the network can be modelled as a graph. Nodes are expected to receive a transmission if and only if exactly one of their neighbours (i.e. nodes that share an edge with the node) transmits. For a class of phase or frequency modulated radio links, the assumed theoretical network model is therefore inaccurate in the assumed collision model.

A better model would be that a node receives the message with the strongest local signal strength. This shall be referred to as the radiometric collision model. For omnidirectional radio links, one can assume that the signal strength monotonically decreases for increasing link distance. This leads to a simplified geometric collision model - a node receives the message from the closest transmitter which is currently sending, if no equally close transmitter is sending at the same time. However, since there is often no direct way for measuring these distances, nodes are generally unaware of the geometric arrangements of the network. If nodes can reliably determine only the topological information, the more general graph-based collision model applies. If nodes can measure the local signal strength or the link distance, then the radiometric or geometric collision model can be used. It is described in 5.5 how the geometric model can be used to improve performance of omnicast communication.

3.4 A high power LED communication module

A prototype optical communication module was designed and built [45]. The module uses emitters with a wide opening angle to achieve omnidirectional coverage and simple, easily available components. The end of this section presents experiments and results for performance in air and water.

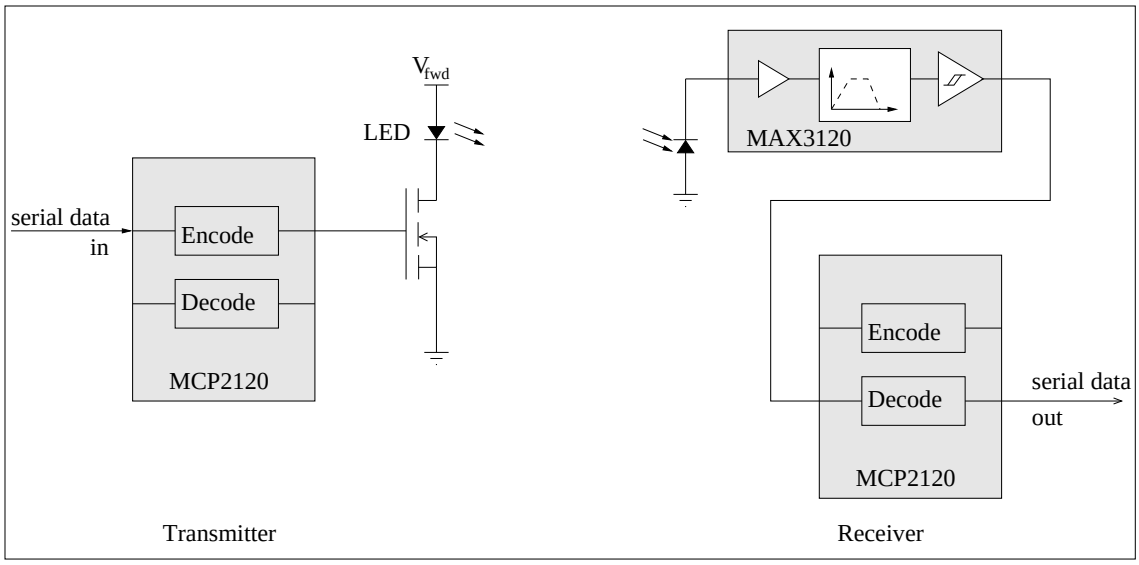

Figure 3.16: Block diagram of the transceiver

3.4.1 Choosing the best wavelength

Finding the optimal wavelength for underwater communication is a difficult task and depends on many factors. It is well known that light absorption in water increases towards the red and infrared part of the spectrum. Minimal absorption is usually achieved for blue light around 400-450 nm. This is only true for clear water, though - aquatic particles like chlorophyll, algae, or plankton have specific absorption patterns, which might lead to an absorption minimum at different wavelengths ([1]). This is known as Rayleigh scattering. Finally, the availability of high power LEDs, their luminance and also the sensitivity of the photodiode for different wavelengths play an important role. The best wavelength for the desired application depends strongly on the implementation and the environment and can only be found experimentally. Results for the current implementation and clear water are presented later on.

3.4.2 Light sources

The recent development in LED technology offers great light intensity, fast switching speed and small packages. The recent availability of 3 Watt LEDs

with superior luminous flux makes underwater optical communication possible. The Luxeon III Emitter by Lumileds ([33]) is a suitable choice. It is available in many different colours, particularly in the wavelengths 460 nm (blue), 490 nm (cyan) and 520 nm (green). The typical opening angle is 50 degrees off axis (or 100 degrees total) at 80 % relative brightness (compared to the brightness on the optical axis) and 60 degrees at 50 % relative brightness. At the maximum average forward current of 1000 mA and forward voltage of 3.9 V, they offer 80 lumen of luminous flux for green and cyan and 30 lumen for the blue emitter. This is approximately 20-50 times brighter than most other ultrabright LEDs. Combined with the wide opening angle, omnidirectional optical communication becomes feasible. New developments occurred after the experiments were performed. The highest brightness and highest efficacy is currently achieved by the X-Lamp XR-E by Cree [8] with up to 90 lumen/Watt and 210 lumen maximum flux at 1000mA. These LEDs can be pulsed to up to 1.8 A, allowing roughly 4 times the brightness of the Luxeon III.

3.4.3 Transceiver implementation

For the experiments presented in the following chapters, a prototype transceiver was developed. It consists of two separate units - a sending unit and a receiving unit. In the experiments, the sending unit and the receiving unit are separated, to test unidirectional communication. In the final implementation every communication partner will be equipped with both a sender and a receiver. The major design criteria are a range which is at least one metre under water and also low costs, small size and simplicity, since the optical transceiver will be used in swarms of small submersibles.

In order to simplify development and production, and also to profit from existing experience in proven technologies, the IrDA physical layer protocol was chosen and adapted to the particular needs of underwater communication. Using IrDA modulation has the advantage that highly optimised integrated circuits are readily available at very low prices. With up to 115 kbit/sec, the data bandwidth is absolutely sufficient for most underwater robotics purposes. It was decided to use 57600 bit/sec, as a compromise between reliability and speed. Using the same components, it is easily possible to reconfigure and increase the bandwidth up to 312.5 kbit/sec if required.

For greater flexibility in swarm applications the higher levels of the IrDA protocol, which have been designed mainly for point to point connections, were discarded. Instead, the implementation described here only offers a stream-based serial optical broadcast link, which can be interfaced to a UART compatible interface. On this low hardware level no link management or higher level error correction is done. Theoretically it is possible to operate in full duplex mode, but special care has to be taken to avoid reception of reflected light. The main changes to the IrDA physical layer specifications are the change of wavelength, optical characteristics, such as opening angle and light intensity, and the possibility of a full duplex mode. The implementation details are outlined in the following sections. Refer to figure 3.16 for a schematic diagram.

3.4.4 Transmitter

The transmitter accepts data over a UART compatible serial interface, encodes the data according to IrDA specifications and generates light pulses in the visible spectrum using high power LEDs. For encoding, a standard IrDA encoder/decoder chip is used (the MCP2120 by Microchip [35]). The output of the infrared encoder controls the MOSFET power stage, driving the high power LED. The Luxeon III emitter described earlier was used as a light emitter, in the wavelengths 460 nm (blue), 490 nm (cyan) and 520 nm (green). The power stage is designed in such a way that it is possible to switch and choose between these three colours. To simplify the underwater experiments, the power stage/LED unit is separate from the encoder unit and sits in a waterproof case.

The transparent domes of the LEDs are in direct contact with the surrounding water, to achieve optimal optical coupling. The emitters are unfocused, with the light emitting chip located in the focal point of the PMMA dome. This means that the light travels approximately perpendicularly through the boundary between dome and water, which minimizes losses due to reflections. This also ensures a similar spatial emission footprint in air and in water.

The power consumption of the transmitter mainly depends on the forward voltage on the LED. For a nominal forward voltage of 3.9 V (80 lumen luminous flux), the average DC current during broadcast is 100 mA (400 mW power).

3.4.5 Receiver

The receiver circuit mainly follows IrDA guidelines and uses standard components. To enable underwater operation using visible light, a special photo diode has to be used which is sensitive for wavelengths between 460 nm and 520 nm. Several different photo diodes were tested; the best results were achieved with the diode SLD-70 BG2A, which has a good tradeoff between speed and sensitivity. The current from the photo diode is amplified by a high gain transimpedance amplifier, followed by a bandpass filter and a trigger, to retrieve the digital modulated signal. The amplification and filtering is implemented using the integrated circuit MAX3120 ([34]). The filtered digital signal is then decoded by another IrDA encoder/decoder MCP2120. Even though one of these chips can already encode and decode IrDA signals, a separate chip is used for the receiver. The IrDA standard does not support full duplex data transmission, which means that the decoder is switched off while data is transmitted. This avoids a sending device receiving its own data, if light is reflected by nearby objects. For some applications outlined later on - especially the distance sensing presented later - full duplex mode is necessary. By using two separate chips for transmitter and receiver, this problem does not occur. Also, this makes it possible to employ multiple independent receivers on one submersible, i.e. for different directions, and only one or also several transmitters.

The receiver is placed inside a small box made from acrylic glass. The photo diode is aligned perpendicularly to the transparent floor of the box. There is room for improvement regarding the optical coupling to water when mounting the receiver to the submersibles. For the initial experiments this setup is sufficient.

3.5 Experiments for optical communication

Experiments were carried out to measure the maximum range and coverage of the optical digital link in air and in water. The range is defined as the maximum distance between transmitter (LED) and receiver (photo diode), for which an error rate of 0% at full transfer rate can be maintained over at least one second. Furthermore, experiments were carried out to investigate how transfer rate and error rate behave at distances greater than the range of the link.

3.5.1 Measurement of error and transfer rate

Since the transceiver interface is a byte-based serial UART interface, the error rate measurement is also byte-based. The following approach was chosen to identify transmission errors:

The sender, connected to the transmitter, sends out a byte stream at full transfer rate of 57600 bits per second. Due to bit overhead this corresponds to 6000 bytes per second. Bytes are sent out in ascending order, modulo 256. The receiver compares every received byte b_n with its predecessor b_{n-1}. If not $b_n \equiv b_{n-1} + 1 (\text{mod } 256)$, then an error is counted. The error rate is measured over intervals of one second, as the errors over the number of received bytes. The criterion used in this experiment is 0% error rate over an interval of one second, which means that 6000 consecutive bytes have been transmitted without any errors in one second. The transfer rate is measured as the number of received bytes per seconds.

There is a small possibility of false positives when exactly 256 bytes are not received and the next byte is then correctly decoded. It is theoretically also possible that two consecutive bytes are both wrongly decoded in a way that they exactly meet the requirement $b_n \equiv b_{n-1} + 1 (\text{mod } 256)$. These cases are very unlikely, and do not play a role if the error rate is 0% at full transfer rate. It must be noted that this measurement does not give representative figures for the bit error rate, but it allows to identify error free transmission of data.

3.5.2 Hardware setup

In order to avoid any unwanted interferences and hidden transmission channels, sender and receiver were physically separated. For transmission, a microcontroller board was connected to the optical transmitter, which generated the described byte stream. The microcontroller board and the transmitter were powered by a laboratory power supply, operated from mains. The receiver circuit was battery powered and connected to a laptop computer, which analysed the received bytes. The closest distance between sender and receiver unit was between the LED and the photo diode. The receiver unit was fixed. To measure the range, only the LED unit was moved.

For the experiments in air the LED and the photo diode were aligned horizontally and positioned well away from reflecting surfaces (> 1 m). The experiment was carried out in normal indoor lighting conditions, mainly fluorescent tubes.

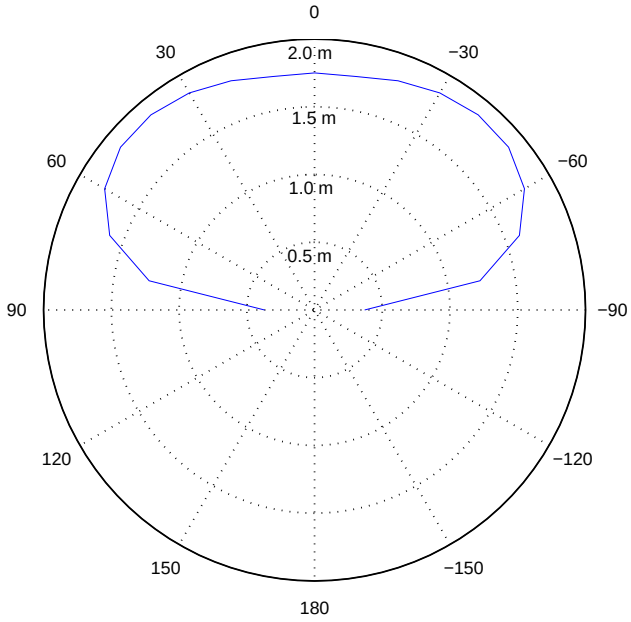

Figure 3.17: Angular coverage of the emitter (range for error-free transmission in air, cyan emitter, forward voltage 3.9 V)

The experiment in water was similar. The transmitter and receiver units were exactly identical. The experiment was carried out in a round pool with white walls, which is 1.5 metres deep and 5 metres in diameter, in clear water with no visible pollution. To avoid effects of the reflecting water surface, the receiver was half-submerged, but floating on the surface, with the photo diode pointing downwards, approximately tilted 15 degrees away from the vertical pool wall, with 50 cm clearance from the wall. The LED unit was submerged underneath the receiver, aligned with and facing the receiver. The experiments were carried out outside, at dusk, with slightly less environmental light than the experiments in air.

3.5.3 Performance in air

The range in this context means the maximum distance between transmitter and receiver that still allows for error-free transmission of data. To measure the range, the distance between transmitter and receiver was increased to the point where

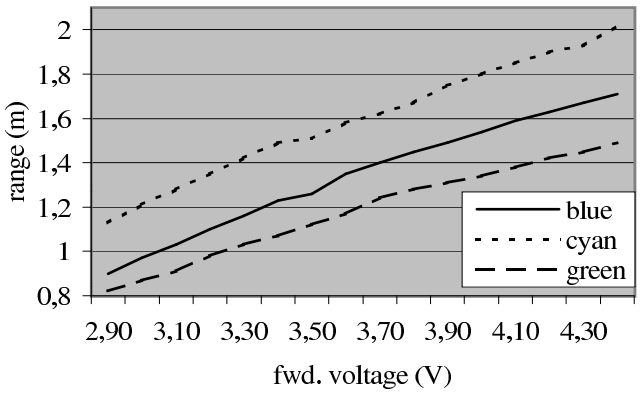

Figure 3.18: Range in air

the error rate becomes larger than 0% over a time span of one second. The measurements were done for three different wavelengths, 460 nm (blue), 490 nm (cyan) and 520 nm (green), and for different forward voltages in steps of 0.1 Volt. The forward voltage across the LED influences the luminous flux of the emitter. The results can be seen in figure 3.18.

As expected, the range increases approximately linearly (apart from an offset) with the forward voltage. This might be surprising at first, since the received light intensity decreases antiproportionally to the square of the increasing distance. It must be considered though, that with increasing forward voltage, the forward current through the emitter increases likewise. This means that the emitted power also increases proportionally to the square of the voltage, which explains the linear relationship. It is obvious that it can only be approximately linear within a range, since LEDs are non-linear devices, and also heat effects can play a role at higher powers. The LEDs are only specified up to a forward voltage of 3.9 V, at a current of 1000 mA, for which the maximum luminous flux of 80 lumen is achieved (refer to the datasheet for details). The maximum forward voltage in the experiments was 4.5 V, which is clearly over the manufacturer's specification. The emitters still worked stably and did not overheat, since they were only driven by pulses with a short duty cycle, but a reduced lifetime might be the result.

As can be easily seen, the different wavelengths yield different ranges. The best range was achieved with the cyan emitter (2.02 m), followed by blue (1.71 m) and

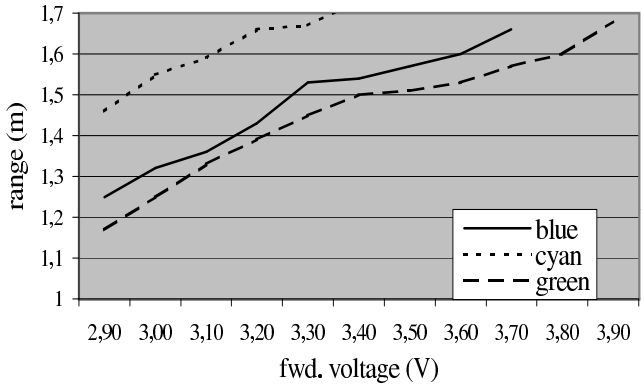

Figure 3.19: Range in water

green (1.49 m). The differences come from the wavelength-dependent sensitivity of the photo diode and also slightly different efficiencies of the emitters.

Figure 3.17 reveals that the range is almost uniform within a cone of 120 degrees opening angle (measured for 3.9 V forward voltage). The opening angle can easily be changed to meet the application by using lenses. The good uniformity allows for a communication link which is fairly direction independent.

3.5.4 Performance in water

The same experiment as described before was conducted in water, in the pool setup described above. Surprisingly the range increased, even though there should be absorption and coupling losses in water. Figure 3.19 shows the results. Unfortunately it was not possible to measure ranges greater than 1.7 metres, due to the limited depth of the pool.

The increased range can easily be explained by the small size of the pool environment. Light from the emitter is reflected by the bright pool walls, increasing the light intensity at the receiver. The LED is bright enough to visibly illuminate the whole pool. The high frequency noise level disturbing the receiver is expected to be lower in the outdoor pool environment, since there is no artificial light source such as fluorescent lamps. Also, water shields and reduces electrical interference at the highly sensitive receiver circuit.

The important result is that clear water attenuation in fresh water does not have a big impact on the range of optical communication up to a range of 1.7

metres when using visible light in the blue and green range of the spectrum. Considering the huge impacts of water on HF radio, a range in the same order of magnitude is a good result. The relative difference in ranges of the three different wavelengths is still approximately the same. This also indicates that the attenuation in clear freshwater is low, so that the changing attenuation of different wavelengths does not have a big impact. While more precise measurements are not possible with our current equipment, the figures suggest an attenuation of less than 1 dB/metre. For purposes within the scope of the submarine project, more precise results are not required.

3.5.5 Possible extension of the usable range

The described setup assumed a wide angle beam pattern, which provides omnidirectional coverage with multiple transmitters and receivers. A common problem of omnidirectional wireless links is that most of the transmitted energy is wasted, as only a very small part of the energy can be detected by the receiver. Collimated beams can significantly increase the communication range in clear water and can improve efficiency. The LED transmitter described earlier can be fitted with a lens that creates a beam with 10 degrees divergence. The bearing to the receiver node has to be known when using narrow beams. The transmitter has to be able to track the receiver with the optical beam. A wide angle camera can be used to detect the bearing to other nodes. The bright emitters carried by each node make them relatively easy to detect. A pan-tilt unit carrying the transmitter (or a mirror assembly) can be used to adjust the beam direction and track the receiver.

Once beam tracking is established, one can furthermore add optics to the photo receptor to reduce the opening angle of acceptance and to optically increase the receptor area. Making the photo receptor on the receiver more direction-dependent improves the signal-to-noise ratio, i.e. the ratio between the modulated signal and the ambient light.

Both methods have the advantage that they do not require more transmitter energy, while the range can be greatly improved. An obvious disadvantage is the added mechanical and technical complexity. Especially on small underwater robots, water proof pan-tilt units are prohibitively complex and unreliable. It is however possible to rigidly mount transmitter-receiver units with narrow opening angles on the submersible and to change the attitude of the craft to

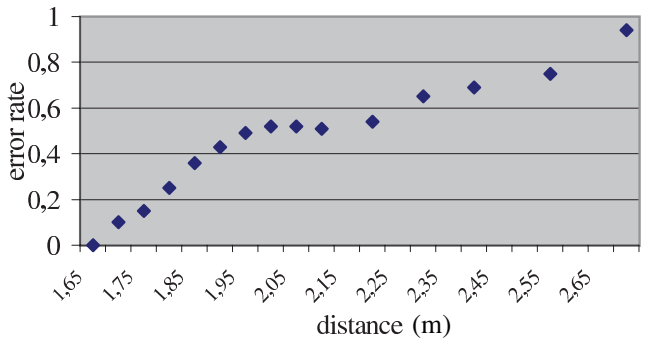

Figure 3.20: Byte error rate for increasing distance.

perform beam tracking. In this case the communication benefits have to be balanced against the additional complexity in the vehicle and swarm control.

3.5.6 Behaviour of the transfer rate for large distances

As a last experiment the behaviour of the transfer rate and error for distances greater than the range was investigated. The results are shown in figures 3.21 and 3.20 For this experiment the cyan emitter was used with a forward voltage of 3.9 V. A range of 1.8 metres is expected. The region of interest for the behaviour of transfer rate and error rate is here above the range, from 1.8 metres to 2.7 metres. The transfer rate was measured and averaged over one second periods. The distance was increased continuously at a rate of one centimetre per second.

It can be seen how the transfer rate is very accurately linked to the distance. As the plot in figure 3.21 reveals, there is only a very small variation. It might appear counter intuitive at first that a digitally decoded, synchronized bytestream shows such a predictable and almost linear decrease in the transfer rate over such a large range of distances, especially if considering that, at 2.7 metres, where the transfer rate almost becomes zero, the received light intensity is only 44% of the intensity at 1.8 metres.

This result suggests that the IrDA physical link layer approximates the additive white Gaussian noise channel model (Binary Input AWGN). This means that noise is mainly generated on the receiver side, i.e. by the preamplifier and filter stage commonly used in IrDA applications. It is reasonable to assume that the

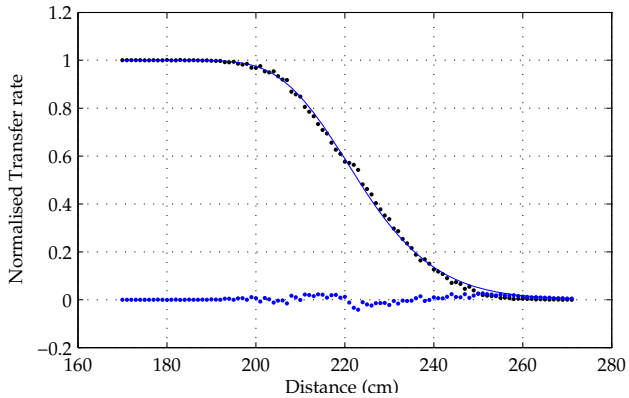

Figure 3.21: Transfer rate for increasing distance. Black dots represent the measured transfer rate, the blue curve is the fitted parametrised model, the blue dots are the error, i.e. the difference between model and data.

optical transmission of light in air is almost noise-free. Furthermore, the relatively short range does not allow for strong interference, scattering or reflection effects.

The reason why the transfer rate is decreasing can be explained as a side effect of the serial communication protocol. The start of a new byte is identified by a start bit. If the start bit cannot be received, the IrDA decoder rejects the byte, which results in a decrease of the transfer rate. Therefore the transfer rate is an indirect measurement of the bit detection error.

According to the AWGN channel model, the received analog signal is

$$Y(t) = X(t) + W(t) \tag{3.7}$$

where $X(t)$ is the transmitted signal and $W(t)$ is white Gaussian noise. This formula can be extended to include the transmission distance, assuming the transmitter is a uniform point source. To include ambient light into the model, the average light power P_{avg} received by the photo diode has to be included as well. The parameter α is for normalisation and has to be identified experimentally.

$$Y(t) = \frac{\alpha}{d^2} X(t) + P_{avg} + W(t) \tag{3.8}$$

After preamplification the signal goes through an adaptive comparator and trigger stage, which adapts to the average power level. The average power level

can be measured by low-pass filtering. By comparing the incoming signal to this adaptive average, the digital pulse signal is recovered. This is a statistical process. The trigger accepts a pulse if the received signal is greater than the average power level, plus a hysteresis term. So the decoded digital signal is

$$Y_d(t) = \begin{cases} 1 & \text{if } Y(t) > (P_{avg} + h) \\ 0 & \text{otherwise} \end{cases} \tag{3.9}$$

Therefore, the probability for receiving and decoding a pulse is

$$\begin{aligned} P(Y_d = 1|X) &= P(Y - (P_{avg} + h) > 0) && (3.10)\\ &= P\left(\frac{\alpha}{d^2}X + W - h > 0\right) && (3.11)\\ &= P\left(-\frac{\alpha}{d^2}X + h < W\right) && (3.12)\\ &= Q_\sigma\left(-\frac{\alpha}{d^2}X - h\right) && (3.13)\\ &=: Q_\sigma(D(X)) && (3.14)\\ \text{with } Q_\sigma(u) &:= \int_u^\infty \frac{1}{\sigma\sqrt{2\pi}} e^{-z^2/(2\sigma^2)} dz \end{aligned}$$

This probability is basically a function of the signal to noise ratio. The signal power is proportional to the inverse square of the distance. The noise level is specified by σ in Q_σ. The detection probability $Q_\sigma(D(X))$ can be rewritten to obtain parameters which are easier to calibrate. We can assume unit variance and rescale the function D(X):

$$Q_1(D'(X)) = Q_1\left(-\bar{\alpha}\left(\frac{1}{d^2} - \frac{1}{d_0^2}\right)X\right) \tag{3.15}$$

Here, the distance d_0 expresses the distance where the transfer rate is only 50%, and $\bar{\alpha}$ is a scaling factor, expressing the luminous flux of the LED and the sensitivity of the detector relative to the noise level. This function approximates the observed data astonishingly well. The fitted curve with $\bar{\alpha} = 400000$ and $d_0 = 223$ cm can be seen in figure 3.21. The absolute error is below 0.02, which is close to the measurement resolution.

3.5.7 Distance sensing

Although the main use of the optical communication system is data transfer between submersible swarm members, the observed behaviour of ranges and

the transfer rate can also be used for distance sensing. This can be done without any changes to the hardware and can be useful to identify the swarm geometry, and to track communication partners. The following considerations assume that the receiver is within the uniform part of the angular coverage (within 60 degrees off axis). This restriction can be checked using a camera sensor, or can also be overcome by using several transmitters, to achieve overall uniform coverage. There are two general ways for measuring the distance between transmitter and receiver; one for short ranges and one for longer distances beyond the communication range.

3.5.8 Short range distance measurements

If the receiver is within the 0% error rate range of the transmitter, the transfer rate and error rate do not reveal the signal strength, and therefore cannot be used for distance measurements. In this case it is possible to change the forward voltage at the transmitter, in order to identify the 0% error rate threshold.

Normally, the transmitter (or master) sends at the maximum nominal forward voltage (3.9 V). When a receiver identifies and decodes the signal, this receiver can calculate transfer rate and verify the 0% error condition. If this is fulfilled, the receiver sends back a request for short-range distance sensing. The master now slowly lowers the forward voltage, and, while doing so, always sends out the currently measured forward voltage. This goes on as long as the receiver can decode the signal without errors. When the range boundary is reached for a specific forward voltage this process can stop, and the receiver can calculate the range for the last correctly received forward voltage, using the range plot shown in figures 3.18 and 3.19 (This is a simple, linear function). The measured distance can be sent back at full power to the master, which then can abort the transmission.

It is also possible to implement this as a non-interactive process. In this case, every unit would send out a frame, where the forward voltage is swept down continuously, while it is again sending out the current voltage. Every receiver within range can then determine the error rate drop-off point individually, which is proportional to the distance. The short-range method has the advantage that the identity of the receiver is known. This makes it easier to generate a geometric swarm model.

3.5.9 Long range distance measurements

If transmitter and receiver are so far apart that an error-free transmission is not possible even at full power, it is still possible to measure the distance. Assuming the sender is transmitting an evenly distributed bit pattern at full speed, all the receiver has to do is to measure the current transfer rate, which stands in a close relationship with the distance (figure 3.21). Using a parametrised, calibrated function, or a look-up table, the distance can be retrieved.

Since error free communication is not possible over this long range, the identity of the sender is not known. This problem can easily be solved if the optical communication system uses a time-sliced, collision-free sender arbitration protocol as described later in chapter 5. In this case, the sending schedule is known throughout the network, and it is easy to look up the current sender in the scheduling information, even if the transmission cannot be decoded correctly.

3.5.10 Limitations

Both approaches assume a homogeneous environment without any obstacles close to receiver, transmitter or the line of sight. Reflections can disturb the measurements. Obviously, both curves for range and transfer rate over distance have to be experimentally measured for the given hardware setup, in order to calibrate the measurements. The calibration results can be stored as parametrised functions, or also as look-up tables. With proper calibration, the expected accuracy is better than 2 cm for both measurement methods.

The distance estimations can be affected by the water quality. For many swarm applications, this is not critical. Assuming homogeneous visibility for the whole swarm, all distances are still relatively correct, up to a scaling factor. The scaling factor can be estimated using other sensors, or by combining dead-reckoning with distance measurements.

3.5.11 Redesign of the optical communication module

The initial prototype used in the experiments was redesigned and manufactured (see figure 3.22). The transmitter and receiver design are mostly identical to the previous version described earlier. The size of the module was significantly reduced by using surface mounted devices.

Figure 3.22: The final optical communication module. The upper board carries the controller and modulation components, the lower board is the programmable switching power supply. The LED and photo diode can be seen at the top, but may be mounted elsewhere, i.e. on the external hull.

A programmable voltage source generates the drive voltage for the LEDs. The voltage can be adjusted in software. The voltage source was implemented as a synchronous switch mode step down converter. The power stage is controlled by an *Atmel ATTiny26L* microcontroller, which senses the output voltage and generates the pulse width modulated signals that are driving the power stage. A software PID controller loop running at 20 kHz maintains a virtually constant output voltage. This design adds flexibility as the drive voltage and therefore the transmitted power can be adjusted instantly to adapt to varying application conditions.

The optical transceiver is modular; the drive voltage source can be detached from the transceiver board. The LED carrier board features a capacitor capable of delivering energy for a sharp pulse and a high speed power MOSFET for switching the LED. Local switching reduces electromagnetic interference and improves the responsiveness. The photo diode carrier board locally preamplifies the received signal and sends a TTL compatible signal to the main module. The complete module was successfully tested.

Method	Return current	Long wave	Optical
Range	10-20 m	5-10 m	2-3 m
Power requ.	$\approx$ 10mW	$\approx$ 100mW	$\approx$ 400mW
Bandwidth	N/A	8192bit/sec	56700bit/sec
Coverage	omnidirectional	omnidirectional	120 deg. cone
Sensing capability	direction, distance	proximity	distance
approximate cost	$20-30	$100	$100
available media	water	air, water	air, clear water
Strengths	cheap, high speed	reliable	high speed
Weaknesses	external electrodes	slow, large antenna	needs clear water, short range

Figure 3.23: Comparison of underwater communication methods as investigated in the previous sections

3.6 Summary

This chapter introduced the advantages and disadvantages of particular underwater communication channels: an optical channel, a radio channel and a return current density channel. The return current field was measured with high resolution in several experiments. For the first time it has been shown that return current communication is possible in freshwater. In-depth investigation of electric communication using return current is subject to future work. The broadband characteristics, long range, low cost and low power consumption makes return current communication a promising candidate for underwater communication in swarms. As proof-of-concept devices, a long-wave radio transceiver and an optical LED communication module were introduced. It was shown experimentally that communication over up to 10 metres is feasible using long-wave radio, with sufficient data rates of up to 8192 bit per second. Interference between multiple transmitters has been measured and described. The observed interference model will be used in chapter 5. The optical link in an omnidirectional form provides higher bit rates, but only over 2 metres range. It is of great advantage to use a hybrid solution on practical systems - a longer-range, slower radio link, in combination with a short-range, high-speed optical link. The low cost and complexity of the communication methods presented here make it

possible to implement all three modalities even on very small submersibles such as the *Serafina* presented in the previous chapter.

Chapter 4

Communication in groups of robots

Group robotics require a change of paradigms in network theory. The first part of this chapter outlines the current paradigms and compares them to the requirements of group robotics. The second part introduces a new mode of communication, which is currently not implemented in existing networks, and provides theoretical findings.

4.1 Introduction

Communication networks are present everywhere - the internet, TV and radio broadcasting, mobile phone networks, wireless networks and many more. Most of these networks rely on a fixed infrastructure, and the channel access and network protocols are tailored to the specific requirements. Also, these networks are often structured according to some hierarchy, and are mostly static.

Communication in classical networks often assumes a specified sender-recipient relation. Packets usually have a defined and intended destination, and most connections are point-to-point. However, in groups of identical communicating

robots, the requirements to the underlying network are different and require a change of paradigms.

For reasons of robustness and scalability, it is reasonable to remove hierarchies - all robots are identical and run identical software (apart from a unique identifier). Adding or removing (losing) robots should not affect the functionality of the group. It therefore makes little sense to explicitly specify a dedicated recipient for any message. Any robot which is able to receive a message can potentially make use of the information. Who the recipients of a message are is rather determined by their location relative to the sender of the message, or other criteria that temporarily distinguish them from other nodes. An underlying communication network should therefore not attempt to artificially distinguish between different nodes.

It is also not practical to expect a fixed infrastructure, since this would complicate deployment and significantly increase complexity in any realistic application. A group of robots is dynamic and might change the network topology at any time. Network protocols therefore have to be able to cope with changing topologies and have to be able to reconfigure themselves efficiently.

Finally, the way information should be dispersed throughout the network in a group of robots is different from how it is implemented in classical networks. Robots operating in groups require continuous real time updates from all surrounding robots with low and predictable latency. In contrast, in many classical networks communication is sporadic and mainly requires a high bandwidth.

4.1.1 Modes of communication

Generally there are four modes of communication - one-to-one, one-to-many, many-to-one and many-to-many. While the first three modes are often and commonly referred to in network theory (unicast, broadcast/multicast and convergecast), interest in the last mode many-to-many seems to be rather new in the context of communication networks.

A very important problem, especially in swarm control and formation control, is the exchange of certain parameters throughout the entire network. This might be control commands from the operator, which would correspond to a broadcast. However, in most autonomous missions there is no contact possible between the operator and the swarm. More importantly the problem is the exchange of

parameters from every member of the swarm to all the other members. This includes, but is not limited to, finding a consensus on the direction and speed of the swarm, i.e. to find the slowest member in order not to lose anyone, estimating the swarm density, the center, size or shape of the swarm, or calculating gradients and extrema on external sensory data such as temperature, brightness, pressure, salinity, just to name a few examples.

This requires efficient communication from everyone to everyone. As a continuation of the communication modes "one-to-one" (unicast), "one-to-many" (multicast), "one-to-everyone" (broadcast) and "everyone-to-one" (convergecast), the term "omnicast" for "everyone-to-everyone" is proposed. A formal definition follows below. This mode is rarely discussed in literature and is sometimes referred to as *global gossiping*. Literature on swarm and formation control [20][21] often assumes the availability of a real time many-to-many or all-to-all communication link. A recent paper [47] confirms that all-to-all communication is required for the collective motion control algorithm proposed there.

4.1.2 Current network technology

Communication networks are ubiquitous, and are employed for a large variety of applications. The properties of communication networks are highly application specific. Networks can be categorized by the communication medium (wired versus wireless), the data flow (e.g. broadcast-oriented, point-to-point, local area, wide area), the organisation (centralised or decentralised), topology (star, bus, cells, etc.) or the application (telephony, television, computer, weather, etc.).

A common model that underlies most networks is called the Open Systems Interconnection Basic Reference model (OSI model)[60]. It defines 7 network abstraction layers (Physical, Data link, Network, Transport, Session, Presentation and Application layer), and also contains a set of protocols. While almost all data network implementations follow the OSI model, some implementations unify several layers into one protocol (i.e. the TCP/IP model only implements 5 layers) Unifying layers reduces overhead, but also limits extendability and modularity. The main advantage of layered models is that layers have well-defined interfaces; implementations of layers can therefore be freely interchanged and mixed. An example is the AppleTalk suite [48], a straight implementation of the 7-layer OSI model. While the higher network layers remained practically unchanged, the physical layer, data link layer and network layer were changed

when newer technology became available. Available choices for the physical and data link layer are LocalTalk and LLAP, Ethernet and ELAP, TokenRing and others. Another implementation (AURP) replaces the three lowest layers with IP, and maps AppleTalk through UDP packages.

Disadvantages of a strictly layered approach is that an implementation might not be able to provide the best possible service to the application, because some information might not be available on the layer on which it is needed. Of course this information can be passed through well-defined interface layers, but this again reduces exchangeability of implementations. How many layers are implemented and what the exact interfaces are has to be decided for each particular application.

4.1.3 Sensor networks

As mentioned in the introduction, a sensor network consists of spatially distributed nodes that are equipped with one or more sensors and communicate to cooperatively gather spatio-temporal data about an environment. Sensor nodes are initially placed, either in a predefined grid or randomly. The nodes do not have actuators to provide mobility (this sets them apart from swarms of mobile robots) and are mostly stationary after initial deployment. An exception are drifters, which are moved by external forces such as wind or currents.

Typical communication scenarios for sensor networks are broadcast (the base station programming the nodes, or uploading new parameters), convergecast (many or all sensors reporting back to the base station) and local gossip (sensor nodes communicating locally with their direct neighbours). The requirements for a sensor network communication infrastructure are therefore similar to robotic swarms, but differ in the mobility of nodes. Algorithms designed for sensor networks are usually not designed for quickly changing network topologies. Another difference which is being caused by node mobility is optimisation of energy consumption. For a sensor node, a large part of the stored energy is used for communication; for this reason many publications concentrate on minimizing energy usage in data transmission and avoiding unnecessary transmissions [54]. In mobile robotic swarms most of the energy is spent on propulsion, which means that the portion of energy spent on communication is less significant. Furthermore, robotic vehicles in groups require quick and continuous updates

about the state of surrounding vehicles. While energy optimisation still plays a role, the focus is more on optimal channel utilisation and real time performance.

4.2 Network model

A common way to model communication networks is using a graph [9]. Let $G = (V, E)$ be a graph describing a network with $n = |V|$ nodes. Vertices $v \in V$ represent communication nodes, containing a transmitter, a receiver and a processing unit. A directed edge $e \in E \subset V \times V; e = (u, v); u, v \in V$ expresses that node u can (in principle) send data to node v. For completeness this includes reflective edges $(u, u) \in E \; \forall u \in V$, even though a node is not assumed to be able to send and receive at the same time. A node $v \in V$ receives a message if and only if there exists exactly one node $u \in V$, so that u is transmitting and $(u, v) \in E$. If there exists more than one node with these properties, a collision occurs at v. In case of a collision, node v can not decode any of the transmitted messages and can in general not detect the collision, meaning it can not distinguish between a collision and noise. Data integrity can still be guaranteed, since v can use higher-level protocols, such as cyclic redundancy check, to verify messages. Noise and distorted transmissions will then be ignored.

More specifically, it is often assumed for simplicity that the network graph is symmetrical and connected. This assumption can be met if all nodes are identical and therefore have an identical range.

4.3 Omnicast communication

4.3.1 Definitions

Definition 4.3.1 (Omnicast) *Let $G = (V, E)$ be a graph describing a communication network with $n = |V|$ nodes. In the start state, every node $u \in V$ has a set $I_u(t_0)$ of information tokens, which contains exactly one unique token B_u of information. During the communication phase, a node v updates its set $I_v(t + 1) = I_v(t) \cup I_w(t)$, if and only if it successfully receives a message from $w \in V$ in time step t (refer to the network model for message exchange), and $I_v(t + 1) = I_v(t)$ otherwise. The end state is reached, when all nodes have the full set with all tokens, $I_u(t) = \{B_v | v \in V\}$ for all $u \in V$.*

Having defined the task to solve, we can now define an optimality problem:

Definition 4.3.2 (The Optimal Omnicast Problem)
Find a schedule $S_G = (T_1, ..., T_t), T_i \subset V$ for $i = 1...t$, with T_i being the set of sending nodes in time step i, such that S_G solves the omnicast on the network graph G, and t is minimal.

It is important to understand that omnicast is not necessarily equivalent to multiple concurrent broadcasts, or a convergecast followed by a broadcast. A key difference between classical broadcast and omnicast in multi-hop networks is that in omnicast it is not necessarily assumed that all messages are retransmitted to everyone in their original form. Nodes are allowed to collect information and reformulate a new message that contains all the crucial new information. For theoretical analysis as described later it is assumed that nodes posess a unique information token, which is passed on to everyone else. This token is only a metaphor for information they posess - it is primarily of interest how long it takes until every node can *in principle* have all the information available anywhere in the swarm. This does not mean that all nodes received every bit of information. A simple example is a distributed maximum calculation; only a single value has to be transmitted in every message (the current local maximum of local data and all received data). Once the omnicast is completed, every node has the correct maximum.

4.3.2 Upper and lower bounds

Let us first find a lower and upper bound for the optimal omnicast. A lower bound shall be defined as a lower bound on the worst-case number of time steps for the best algorithm, working on arbitrary connected graphs. It is not the minimal number of time steps it will at least take for all graphs - but for each algorithm, there is a worst case in which it can not be solved faster than the lower bound. A lower bound is usually specified as a function on properties of the graph, such as the number of nodes n, the diameter D, etc. This definition is in accordance with the literature. Formally, let $\mathcal{A} = \{A : \mathcal{G} \mapsto \mathcal{S} |$ A solves omnicast $\}$ be the class of all algorithms (or functions) A, that solve the omnicast problem and that map from a subclass $\mathcal{G}$ of all connected graphs to the class of all schedules

$\mathcal{S}$. Let $|S|$ denote the number of time steps t of that schedule. Then, a function $L : \mathcal{G} \mapsto \mathbb{N}$ is called *lower bound*, if it fulfills the following condition:

$$\forall A \in \mathcal{A} : \exists\, G \in \mathcal{G} : |A(G)| \geqslant L(G) \tag{4.1}$$

The *absolute lower bound* $L_a : \mathcal{G} \mapsto \mathbb{N}$ is the minimum number of time steps any algorithm needs for any given graph:

$$\forall A \in \mathcal{A} : \forall\, G \in \mathcal{G} : |A(G)| \geqslant L_a(G) \tag{4.2}$$

An *upper bound* $U : \mathcal{G} \mapsto \mathbb{N}$ is an upper bound for the worst case number of time steps for the best algorithm and is defined likewise with this condition

$$\exists A \in \mathcal{A} : \forall\, G \in \mathcal{G} : |A(G)| \leqslant U(G) \tag{4.3}$$

Obviously, since omnicast solves broadcast, it can not be faster than an optimal broadcast. That means that a lower bound for broadcast is also always a lower bound for omnicast. [6]

Theorem 4.3.3 *Let G be a connected graph with n nodes. $L(G) = n$ is a lower bound for omnicast on the network modelled by G.*

Proof. Consider the class of all fully connected graphs. To solve omnicast in this class, each node has to send a message with its token of information at least once. If more than one node sends per time step, no node can receive any information, therefore only exactly one node can send per time step. In this case, after n timestep every node transmitted exactly once, the omnicast is solved, and every node has every token of information. It follows that a lower bound for omnicast in general is $L(G) = |V| = n$. □

Theorem 4.3.4 *Let G be a connected graph with diameter D_g. Then $L_a(G) = D_G$ is an absolute lower bound for omnicast on the network modelled by G.*

Proof. Consider the shortest path in G with maximum length D_G. Obviously, information has to be exchanged from the start to the end of this path. A particular token of information can only proceed by one node per time step on that path. The token from the start node of the path hence needs at least D_G time step to reach the end node. Omnicast can not be solved faster than this on any graph. □

Theorem 4.3.5 *Let G be a connected graph with n nodes. Then $U(G) = (n^2 - n)$ is an upper bound for omnicast on the network modelled by G.*

Proof. It is sufficient to show the existence of an algorithm that solves omnicast in all cases in not more than $(n^2 - n)$ time steps. Consider an optimal schedule, which can be found by exhaustive search. Assume each node's information state I is described by an n-dimensional bit vector, that describes which tokens of information it owns. In the beginning, each node's vector has exactly one bit set, its own bit. In the end state, every node's vector has every bit set. This means that altogether $(n^2 - n)$ bits have to be set by communicating tokens. In every time step, at least one bit will be set in the whole network, otherwise this time step would be redundant, and the schedule would not be optimal. It follows that an optimal schedule has a maximum of $(n^2 - n)$ time steps, which is therefore an upper bound for omnicast. □

Theorem 4.3.6 *If G is a graph with n nodes and G is a member of the subclass of Hamiltonian graphs, then $U(G) = (2n - 3)$ is an upper bound for omnicast on the network modelled by G.*

Proof. If G' is a Hamiltonian graph, there exists a cycle $C = (1, \ldots, n)$ which visits each node exactly once. Proceed on that cycle, one node each time step, with the currently visited node being the only sender. After n time steps, the cycle is completed and node n, node 1 and node $n - 1$ now own all tokens of information. Proceed again on the same cycle. At time step $(2n - 3)$, node $n - 3$ sends and provides node $n - 2$ with all tokens. All nodes have now received the full set information. □

A linear upper bound can also be given for arbitrary connected graphs. The idea for the following proof for theorem 4.3.8 has been provided by John Hallam. The author would like to express thanks for the kind contribution. The proof is included here for reasons of completeness.

Lemma 4.3.7 *Every undirected finite connected graph can be disassembled, one node at a time, without disconnection.*

Proof. A graph either contains cycles, or it does not. In the former case, remove an edge from a cycle. This does not disconnect the graph. Repeat until the graph does not contain any cycles. A graph without cycles is a tree. Removing a leaf from a tree does not disconnect it. Repeat removing leafs from the tree, until it is

empty. The order of nodes being removed from the tree can also be applied to the original graph, disassembling it without disconnection. □

Theorem 4.3.8 *Let G be a connected graph with n nodes. Then $U(G) = (2n - 2)$ is an upper bound for omnicast on the network modelled by G.*

Proof. For induction, assume there is a solution for omnicast with $2(k - 1) - 2$ steps, for any connected graph with $k - 1$ nodes.

Take a graph G of size k and remove a node without disconnection. The resulting graph G' has an omnicast solution with at most 2k-4 steps. Add two steps to this to obtain a solution for G:

Step 1:	the removed node sends.
Step $2 \ldots 2k - 3$:	apply the solution for G'.
Step $2k - 2$:	any node connected to the removed node sends.

This is an omnicast solution of length $2k - 2$.

Base: The trivial graph of size 1 requires 0 steps. □

The proof for linear complexity of omnicast can also be extended to the more general case of directed graphs.

Theorem 4.3.9 *Let G be a directed, strongly connected graph (there is a path from each vertex to every other vertex) with n nodes, such that an omnicast solution exists. Omnicast can be solved on G in at most $2n - 2$ steps.*

Proof. It is possible to split the omnicast problem into a convergecast followed by a broadcast. For simplicity we assume that only one node sends at a time. To obtain the schedule for the convergecast, select a target node. Invert all edges and perform a breadth-first search, starting from the target node. The nodes send in the reverse order in which they were discovered by the breadth-first search (not including the target node). Similarly, to obtain the schedule for the broadcast, perform a breadth-first search starting from the target node, with the original edge orientation. The nodes send in the order in which they were discovered (including and starting with the target node).

The convergecast requires at most $n - 1$ steps (the target node does not have to send). The broadcast requires at most $n - 1$ steps (the nodes discovered in the last round of the breadth-first search do not have to send - this is at least one node).

After the convergecast finishes, the target node has all information. After the broadcast, every node has all the information. Therefore, omnicast can be solved in at most $2n - 2$ steps. □

It is obvious that there are better solutions. The broadcast will usually require much less than $n - 1$ steps, depending on the number of nodes discovered in the last round. Additionally, by applying concurrent schedules, omnicast converges much faster in most cases.

4.4 Solutions for special classes of graphs

It has already been shown in theorem 4.3.3 that an optimal solution for fully connected graphs requires exactly n time steps. It is obvious that any complete enumeration of all nodes in G corresponds to an optimal solution.

It is possible to extend this to all graphs of radius 1. In this case, there exists a node (the center), which is connected to all other nodes. It is always possible to construct a schedule with n time steps, by simply letting all nodes except the center node send one after another, and let the center send as the last node. The center will have accumulated all information by then, and a single transmission from the center reaches all other nodes, completing the task. These solutions are optimal among the solutions without collisions, but are not necessarily optimal among all solutions. This can be illustrated by a counter example.

Imagine a graph $G = (V, E)$ with radius 1, for which it is possible to partition its nodes into three disjunct, non-empty subsets V_l, V_r, $C \subset V$, such that $|C| = 1$ and the subgraphs $(V_l \cup C, E)$ and $(V_l \cup C, E)$ are fully connected, and $\nexists u \in V_l, v \in V_r : (u, v) \in E$. This class of graphs shall be called *butterfly graphs* (Figure 4.1 shows an example with 7 nodes). It is now possible to independently and concurrently solve omnicast on the subgraphs (V_l, E) and (V_r, E). Since V_l and V_r are fully connected, this takes $\max\{|V_l|, |V_r|\}$ time steps. The center node in C may have experienced a collision in every time step and may therefore not have received any information. If we make sure that the last node sending in V_r and respectively V_l may send exclusively, only one extra time step is required. In this scenario, the center node will have received all information from V_l and from V_r, which it now can transmit to all other nodes in V. The overall number of time steps required for networks modelled by butterfly graphs is therefore $(\max\{|V_l|, |V_r|\} + 2) \leq n$.

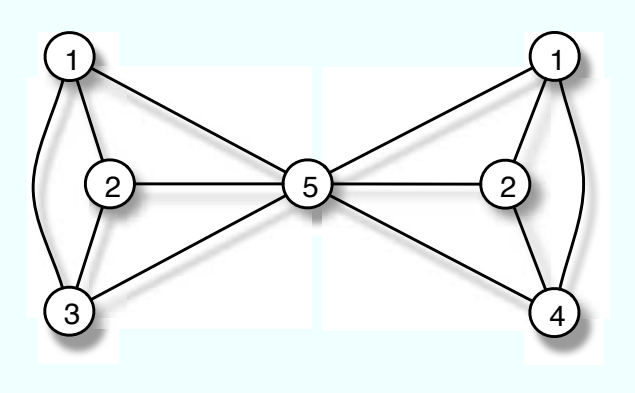

Figure 4.1: A butterfly graph with 7 nodes (numbers next to nodes are transmission time steps)

This shows that omnicast can be solved in less than n time steps for butterfly graphs if $|V_l| > 1$ and $|V_r| > 1$. On the other hand, if collisions are not permitted, no two nodes can send at the same time, or else the center node would experience a collision. Since every node has to send at least once, a collision free solution for omnicast requires at least n time steps in butterfly graphs. It follows that avoiding collisions can yield suboptimal solutions.

Another class of graphs is symmetric graphs of diameter $n-1$, that is, graphs which are a single line (figure 4.2). We can construct a solution with exactly n time steps for all graphs of this class. Assume an undirected graph $G = (V, E)$ with $V = \{v_1, v_2, \ldots, v_n\}$, and $E = \{(v_1, v_2), (v_2, v_3), \ldots, (v_{n-1}, v_n)\}$. A solution for a schedule S_G is of the following form: $S = (T_1, ..., T_n)$, with

$$T_i = \begin{cases} \{V_i, V_{n-i+1}\} & \text{for} \quad i = 1 \ldots (\lfloor n/2 \rfloor - 1) \\ \{V_{\lfloor n/2 \rfloor}\} & \text{for} \quad i = \lfloor n/2 \rfloor \\ \{V_{\lceil n/2 \rceil + 1}\} & \text{for} \quad i = \lfloor n/2 \rfloor + 1 \\ \{V_{\lceil n/2 \rceil}\} & \text{for} \quad i = \lfloor n/2 \rfloor + 2 \\ \{V_{i-1}, V_{n-i+2}\} & \text{for} \quad i = (\lfloor n/2 \rfloor + 3) \ldots n \end{cases} \tag{4.4}$$

This applies to graphs with an either odd or even number of nodes. Nevertheless, only for graphs with an even number of nodes, this solution is collision free. For graphs with an odd number of nodes, the center node will experience a collision in time step $(\lfloor n/2 \rfloor + 3)$, and it shall be noted that in this case there

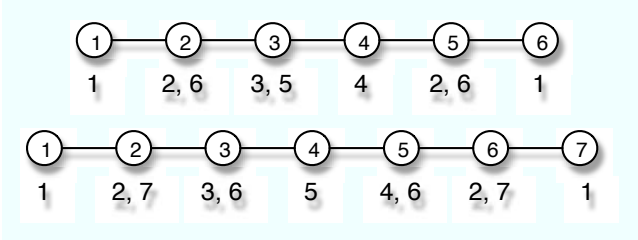

Figure 4.2: Optimal schedules for line graphs (numbers next to nodes are transmission time steps)

is no collision free solution with only n time steps. It can be shown that this solution is optimal for line graphs. Transporting information from the left end to the right end of the line graph requires $n - 1$ time steps. The same applies for transporting information from the right end to the left end. Both processes can be performed in parallel. However, at some point the information token from the left-most node and the right-most node have to swap sides. This requires at least one extra time step, otherwise the occuring collision will avoid information exchange. Therefore the shortest solution for line graphs needs at least n steps; it follows that the optimal solution for line graphs has n time steps.

4.4.1 Full search results

For analysing optimal schedules on small graphs, an exhaustive search on all schedules was implemented. The run time of the search algorithm is exponential, so it is only possible to analyse small graphs up to 7 nodes. The search algorithm itself is a hybrid between breadth-first and depth-first. It starts as breadth-first, until the available memory is exhausted and continues depth-first with limited search depth.

The search algorithm iterates on the time steps of the schedule. For every time step, all possible sets of senders are evaluated. For each possible set of senders, the new information state vector for the network is calculated and added to the input search space for the next time step. This continues until the end state vector with all bits set is found. In case of the depth first search, the maximum search depth is limited. A first guess for the depth limit is the number of nodes, which is further refined by calculating an optimal collision free solution. This converges

much faster than a full search, since the number of possible sets of senders is heavily restricted. In the second stage, a full search is performed. The algorithm starts with a breadth-first search, which runs slightly faster. When it hits the memory limit, it then switches to depth-first search, based on the last output set of the breadth first search. Once a solution is found, it finishes calculation for the current time step, to collect all optimal solutions. In case of the depth first search, the search depth is reduced to the number of time steps of the best solution found so far.

The simulation program gives the choice between a search on collision free solutions only and on all solutions. The set of sets of senders is pre-calculated, based on a collision analysis on the input graph. Furthermore, a greedy version of the algorithm has been implemented. Here, in every time step, only states with the largest Hamming norm (the number of bits set) of their information state vector are kept for the next time step.

The described search algorithm is exponential in the number of nodes, and in the number of time steps, which is an unknown function of the number of nodes. Run times are therefore extremely long. A full search takes approximately 5 seconds for 5 nodes, 3 minutes for 6 nodes and 3 days for 7 nodes. The exact times are not relevant, and so the numbers given here are only to indicate the order of magnitude for run times to be expected on a single-CPU instalment at clock rates of about 2 GHz. Collision free solutions can be found much faster - approximately 10-15 seconds for 7 and 8 nodes and 1-2 hours for 9 nodes. Obviously the run time heavily depends on the number of time steps and therefore also on the complexity of the network. However, the greedy algorithm can perform analyses of networks up to 25 nodes in reasonable time.

The greedy algorithm delivers only sub-optimal solutions. Interestingly it usually performs better if the search space is restricted to collision free schedules.

In most cases the optimal solution contains collisions. For a butterfly network with 7 nodes, shown in figure 4.1, a collision free solution requires 7 time steps; the optimal solution with 2 collisions only requires 5 time steps. This can be easily explained by the fact that even though one node suffered a collision, several other nodes did not, and could still successfully decode the message. The information gain outweighs the disadvantage of a collision.

Furthermore, up to now no small graph could be found for which the optimal omnicast would require more than n time steps. All connected graphs of up to 6

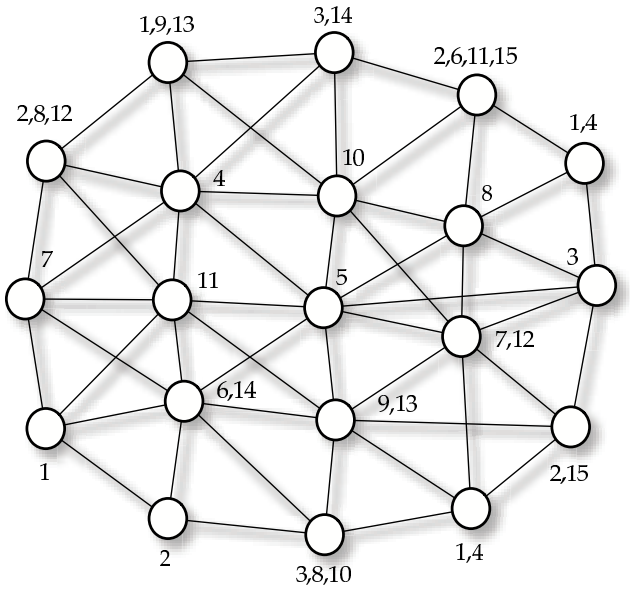

Figure 4.3: A solution with only 15 time steps can be found for this 20-node network.

nodes have been exhaustively searched. All optimal solutions had n or $n-1$ time steps (n being the number of nodes). This suggests that a better upper bound for omnicast could be n - unfortunately there is no proof for this hypothesis yet.

The 20-node graph with diameter 4 shown in figure 4.3 has a solution of omnicast with only 15 time steps. This solution has been found with the greedy algorithm, searching only collision free schedules. There is most likely a better solution, which is also expected to contain collisions. Unfortunately, there is no feasible way to find optimal solutions for a graph of this size. Even though this solution is suboptimal, it further strengthens the hypothesis that n is an upper bound for omnicast. The solution also gives valuable hints how to construct a local heuristic algorithm - it can be observed that very often neighbouring nodes send consecutively. This has the advantage that a sending node can further distribute information, which it just received. Or, to put it another way, every token of information moves further every time step without delay. A similar argument applies to Hamiltonian graphs, where it has been shown that omnicast can be solved in linear time. This seems to be a useful starting point for a distributed local TDMA scheduling algorithm. If existing TDMA scheduling algorithms

can be modified in a way, so that they try to assign successive time slots to neighbouring nodes, they should show improved performance with regard to the omnicast problem and also the speed of information dissemination.

4.4.2 A geometrically derived upper bound

The considerations so far assumed that the network is described by an arbitrary connected graph. In the case of multi-hop radio networks, these network graphs typically have an embedding in three dimensional Cartesian space. It is therefore possible to make further assumptions in order to arrive at a better upper bound for some types of networks.

An approximated upper bound can be derived geometrically, to gain more insight into the performance of networks embedded in 2-D or 3-D space. Let $G = (V, E)$ be the network graph of a network, and P the longest shortest path in G (diameter $d = |P|$). Select every fourth node on P: $K := P_2, P_6, P_{10}, ...$, so that their two-hop neighbourhoods $k \in K \mapsto C(k) := \{n \in V : |n, k| \leq 2\}$ are overlapping. There are $O(d(G)/4)$ nodes in K. Each cell $C(k)$ can achieve complete information exchange among all nodes in $C(k)$ in less than 2γ time steps, with γ being the size of the largest 2-hop neighbourhood (every node in a 2-hop neighbourhood sends twice in a dense schedule). To transmit all information along P this has to be repeated $d/4$ times (all other paths are shorter, therefore omnicast is complete). The upper bound is therefore $U = O(\frac{1}{2}\gamma \cdot d)$ time steps.

Assume a homogenous network of circular shape with n nodes in a plane with equal node density, network area A, and r being the communication range. The graph degree can be approximated by $\Delta' = O\left(n\frac{r^2}{A}\right)$. The network diameter is approximately $d' = O\left(\sqrt{A}/r\right)$. The size of a 2-hop neighbourhood is approximately $\gamma' = O\left(n\frac{4r^2}{A}\right)$. It follows that the geometrically approximated upper bound is $U' = \gamma' \cdot d' = O\left(n\frac{r}{\sqrt{A}}\right)$. The 3-D case is very similar and leads to an upper bound of $O\left(n\frac{r^2}{\sqrt[3]{A}}\right)$. This geometric approximation is only valid for large n, and if r is greater than the maximum distance between nodes. Obviously the connectivity of the network suffers with small values for r, and the network can become disconnected if the value is too small. What can be seen is that, at least for sufficiently large networks and carefully chosen communication range, the upper bound is proportional to the communication range r for a given network size. It

is obvious that this geometric approximation does not apply for small networks. Simulation results presented later give actual numbers for various network sizes, and provide support for the presented approximation.

The upper bound U' suggests that r can be chosen for a given network area and density, so that the upper bound for omnicast is smaller than n. In fact, solutions can be found that require less than n time steps for networks with small degree. However, fully connected networks have a lower bound of n steps. Additionally, fully connected graphs suffer from poor local update rates - the frequency at which nodes can send is exactly $1/n$. Networks with lower connectivity allow for parallel communication in different parts of the network; the per node transmission frequency is proportional to $1/d(G)$. Fast local update rates are important for swarm control.

Generally this means that for a given swarm size, links with a range smaller than the swarm diameter are a significant advantage. A similar result was presented in [13]. In a swarm with evenly spaced robots the communication links optimally connect each node only to its direct geometric neighbours. However, in real swarms it can be necessary to increase the range for better redundancy.

4.5 Summary

The many-to-many communication mode *omnicast* has been introduced. It was shown that omnicast can be solved in linear time. Furthermore, special cases have been presented where the number of time steps required to achieve global information exchange is equal to the number of nodes in the network. Exhaustive searches revealed that there are network graphs for which an even shorter solution can be found.

An intuitive geometric approximation suggests that omnicast can be solved faster for sparsely connected networks, or in other words, in networks where the wireless link range is short compared to the geometric size of the network. For large numbers of nodes the upper bound for omnicast is approximately proportional to the range of the communication links, if the range is sufficiently larger than the distance between nodes. It follows that short communication ranges are not necessarily a disadvantage for communication in large groups, but rather an advantage.

Chapter 5

Ad hoc networking

A common problem with wireless networks is that the network topology is not known and can dynamically change - especially in robotic swarms where nodes are mobile. Medium access therefore has to be adaptive and robust to changes. Furthermore, at start up of a multi-hop radio network, there is no prior communication infrastructure. This poses a bootstrap problem, as information has to be exchanged in order to identify and distribute the current network topology, the number and identity of participating nodes and parameters for the medium access algorithm. Solving these problems is commonly called *Ad hoc networking*, meaning that a network configures and maintains itself automatically as nodes are added, moved or removed, without initial knowledge of the network topology. This chapter gives an overview of ad hoc networking methods and proposes an ad hoc time division multiple access (TDMA) algorithm suitable for robotic swarms.

5.1 Medium access with multiple transmitters

When there is more than one transmitter accessing the same medium, the problem of interference occurs. Transmitters have to access the medium in

a way that avoids interference with other transmissions as much as possible. The problem is common to both wired and wireless networks. This thesis focuses on wireless networks. There are several possibilities to access a medium avoiding interference - transmissions can be separated by time, space, frequency, polarisation, medium, modulation or other means.

Separation by time is achieved by scheduling transmissions so that no two transmissions happen in overlapping time intervals. This method is employed in Time Division Multiple Access protocols and, in its non-deterministic form, in ALOHA or CSMA/CA type protocols.

Separation in space can be achieved by placing the transmitters far enough apart from each other so that they are outside the other transmitter's range (the signal strength is below the noise level). Another method is to limit the wave propagation of the transmission to a smaller volume, i.e. beam forming, directional antennas, or using lasers for optical communication. In the case of electromagnetic waves, transmissions can travel through the same volume without interfering. Beams can be sent out so that every receiver is only within one beam. Alternatively, receivers can employ directional filtering, only accepting signals from a particular direction. Examples are laser links, parabolic antennas as used for satellite TV reception or microwave radio links and phased arrays and MIMO channels.

Frequency separation assigns different carrier frequencies to each transmitter, so that their emitted bands do not overlap. As the available frequency bands are limited, and carrier frequencies have to be separated by at least the data bandwidth, this only works for a relatively small number of transmitters. Frequency separation techniques are mostly combined with separation in time or space. A commonly known example of pure frequency separation is CB radio, which typically implements 40 channels. A related technique is separation by polarisation, which is used in satellite TV reception.

A fairly recent method is separation by modulation, also called Code Division Multiple Access (CDMA), which is a form of spread spectrum radio communication. Each transmitter signals over the full range of the assigned frequency band, by phase-modulating the carrier with a pseudo-random sequence. If this sequence is known to the receiver, it can reverse the modulation and decode the signal. The codes that are applied by different transmitters are orthogonal; this makes the modulated signal robust against interference. Spread spectrum

techniques require high bandwidth and high frequency carriers compared to the bandwidth of the data being communicated. Spread spectrum is usually implemented in the GHz range.

Most practical implementations use a combination of these techniques, i.e. a combination of frequency division (FDMA), spread spectrum and time division. An example is WiFi according to standard 802.11b, which uses a combination of Direct Sequence Spread Spectrum (DSSS) and CSMA/CA (Carrier sense multiple access with collision avoidance).

Chapter 3 introduced digital long-wave communication and optical pulse modulated communication for wireless short range underwater communication. The particular problems with long-wave radio communication are the limited bandwidth due to the low carrier frequency and the difficulty of designing efficient small antennas. Additionally, water as a medium for radio waves has effects on the signal quality, such as frequency-dependent dispersion and attenuation [51] [52]. It is therefore desirable to use a narrow bandwidth to reduce dispersion and signal alterations and to be able to use tuned high gain antennas. The particular system with antenna presented in this thesis has a narrow bandwidth. Spread spectrum techniques require wide band radio and cannot be easily implemented on such a long-wave radio system. While it would be possible to use FDMA methods by splitting the frequency band into narrow sub-bands, this would limit the data bandwidth available per channel. It has been decided to instead use only one channel and share it in the time domain.

Optical channels as introduced in chapter 3 generally do not allow direct frequency modulation of the carrier signal. This excludes the use of phase or frequency modulation and also spread spectrum modulation. Typically pulse modulation or amplitude modulation are used. It is possible to use light sources of different wavelengths and optical bandpass filters to construct multiple channels. However, this only allows for a small and fixed number of channels, far less than the expected number of vehicles. Additionally, different wavelengths of light again are subject to frequency dependent attenuation and scattering. With multiple frequency transmitters and receivers the technical complexity and space requirements are multiplied. The problem of channel access is therefore only shifted, but not solved. It is far simpler to use only a single high data rate transmitter and receiver per node and implement time sharing of the channel.

Optical channels also allow beam forming by using optics or lasers. This offers an easy way of space multiplexing the channel. In the case of mobile nodes, the beams have to track the receiver, which requires precise position information and mechanical devices to steer the beam. This usually adds too much complexity for practical small scale underwater robots, as the unit has to be water and pressure proof, and the optical coupling of transmitter and receiver to water (the medium) is crucial. A possibility is a number of beam transmitters fixed to the vehicle, whereas the whole vehicle is moved in such a way that the beam tracks the receiver. This method would offer superior range and energy efficiency over omnidirectional optical communication, but the control of such a system requires complex swarm control algorithms and is beyond the scope of this thesis. A swarm that dedicates a number of nodes to form a wide area optical backbone by moving and arranging vehicles with fixed optical beams is proposed as future work.

5.2 Channel access protocols

Due to the severe bandwidth limitations and real time requirements in underwater swarms, Time Division Multiple Access (TDMA) is a good choice. TDMA scheduling algorithms are known in literature, which are mostly tailored for sensor networks or applications with sporadic communication [15, 17, 54, 59, 18, 30, 16, 14]. In [30] a TDMA algorithm specifically for sensor networks is introduced that also allows for tuning to either broadcast, convergecast or local gossip. It is assumed that nodes are arranged in a rectangular or hexagonal grid and that sensors know their location on the grid. This limits the algorithm to static sensor networks with known placement of nodes. [18] gives a TDMA slot assignment algorithm suitable for ad hoc networks (the node location is not known), which produces collision free schedules. The authors assume omnidirectional, bidirectional communication and sparse node distribution. The algorithm selects leaders that perform a distance-two colouring and dictate the schedule to non-leader nodes. The paper [15] presents a gossiping algorithm, which effectively is a TDMA scheduling algorithm optimised for a single round of omnicast (or global gossip). The authors do not assume knowledge of the network topology, but it is assumed that the total number of nodes is known, the network is strongly connected, and nodes can transmit all their current

knowledge in a single message. The algorithm implements a strategy called *collate and broadcast*. It is not designed for continuous repeated omnicast, and it only optimises for global information exchange while not considering maximal use of bandwidth in local neighbourhoods. Swarming requires fast updates locally and globally; this algorithm is therefore not expected to perform well in a swarming application.

The problem with comparing different approaches and algorithms from different communities is that the underlying assumptions and the intended applications are often different. It has to be noted that single shot ad hoc algorithms for broadcast and omnicast [14][15][29][59] usually do not reuse information about the network topology for the next round and are therefore principally suboptimal for continuous communication compared to continuous TDMA scheduling algorithms. Technically broadcast and gossiping algorithms mostly are TDMA slot assignment algorithms, but usually all knowledge about the network is discarded after each round. Some papers on broadcast capacity assume that each node wants to broadcast a verbatim message, and therefore messages have to be resent in unmodified form by other nodes [53][29]. This is different from the assumption in some other papers that nodes can transmit all their current knowledge in a single message [15][42][43][52]. Often the conclusions seem contradictory; [53] states that the broadcast capacity of arbitrary ad hoc networks is bounded by $O(1/n)$, which is the capacity of fully connected networks, while [44] and this thesis show that collision free omnicast is fastest in sparsely connected networks. Although seemingly contradictory, both are true, as the underlying assumptions differ in the point of information content of single messages. A similar example is [3], which identifies an exponential gap between determinism and randomisation for broadcast algorithms. The paper assumes that the network topology is unknown at the beginning of the broadcast. However, in the case of a continuously running TDMA scheduling algorithm, knowledge about the (local) topology can be accumulated over several rounds, hence the result from [3] does not apply here.

A distributed TDMA scheduling algorithm adjusted to omnicast communication in swarms is presented in [42]. It assumes a strongly connected network and that nodes can transmit all current knowledge in a single message. Simulation results for continuous omnicast operation are presented. A variant of this algorithm is presented later in this thesis.

Experiments with the phase-modulated long-wave radio modules revealed that the graph-theoretical network model commonly assumed in literature is too conservative (4.4.2 and [43]). In fact, if two or more transmitters send within the range of a receiver the receiver will only observe a collision if the closest nodes have very similar incoming signal strength. Otherwise the receiver will reliably receive the message with the highest signal strength. This chapter presents two TDMA scheduling algorithms. The first algorithm is based on the graph based collision model. The second algorithm uses a geometric collision model. It is shown how this can be used to speed up both local and global information exchange by virtually decreasing the local degree of the connection graph as seen by the scheduling algorithm. This technique assumes that the signal strength can be measured by the receiver.

5.3 Distributed Ad hoc Omnicast Scheduling

This section describes the DAOS algorithm. The assumed underlying collision model is the graph network model - a node receives a message if and only if exactly one of its neighbours in the graph sends. The algorithm converges towards a solution that is collision free in the sense of this collision model. Measurements of the signal strength or the link distance are not required, which makes this algorithm quite versatile. The algorithm is symmetrically distributed - each node runs identical code. Nodes are assumed to have a unique identification number (ID). Furthermore, the algorithm is able to bootstrap and can adapt to a dynamically changing network topology.

In order to avoid collisions, it has to be guaranteed that no node will have more than one transmitting neighbour at any time. This is equivalent to that a maximum of one node may transmit within any 2-hop neighbourhood at any given time. This can be described as a graph colouring problem. If $G = (V, E)$ is the network graph, let $G^2 = (V, E^2)$ be the graph with an edge between any nodes $n, k \in V$ with $|n, k| \leq 2$. A collision free schedule at time t is a colouring of G^2 with 2 colours (transmitting, not transmitting), so that nodes marked as transmitting are not neighboured in G^2.

In a distributed setup the knowledge of nodes is limited to their neighbourhood. Nodes can increase knowledge by communication, but the communication

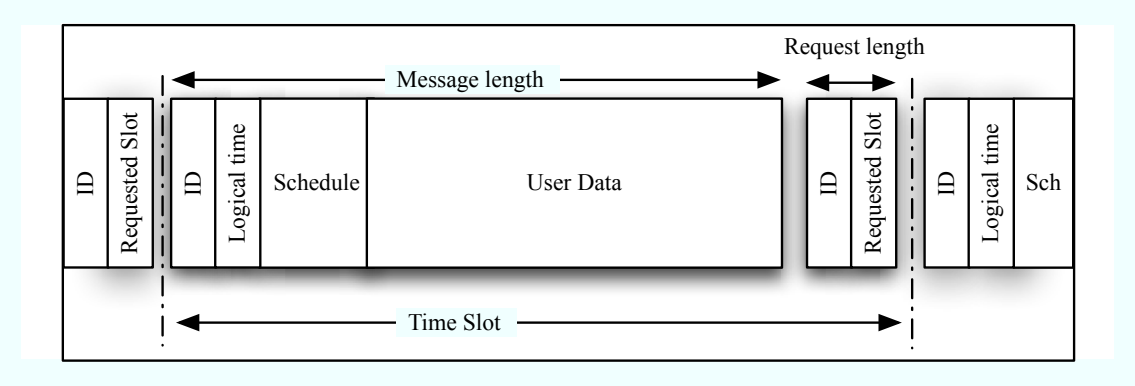

Figure 5.1: Format of the data packet and the time slot

overhead rises quickly for larger groups. If every node knows when each of its neighbours sends, it can detect potential collisions of its neighbours. There are two problems: If a node detects a collision, it has to inform the colliding nodes. This might not be possible if the collision impairs communication. Second, nodes can only know about their neighbours if they successfully receive a message from them. In the case of two neighboured nodes causing a collision, the only nodes that can detect and resolve the collision are the nodes that are affected by it. In other words, these nodes will not receive any of the messages, and therefore no node will be able to detect the collision.

If every node knows the schedule slots of its 2-hop neighbourhood, they can determine if they can send themselves. A node can only send in a particular slot if no other node in their 2-hop neighbourhood sends in the same slot. This simplifies the problem, as this is a local decision.

5.3.1 The basics

DAOS is a distributed algorithm for TDMA scheduling in wireless multi-hop networks. It is computationally inexpensive and can be implemented on a small microcontroller. It is designed for continuous traffic and quick local and global information dissemination.

Communication nodes maintain a logical clock $L(t)$, an integer clock which increments at the beginning of each time slot. The logical clocks are synchronised

during the start up of the network by monitoring incoming messages. It can be assumed that the logical clock $L(t)$ is equal and in sync up to sufficient accuracy on all participating nodes.

To allow for dense packing of schedules on one hand, but reconfigurability on the other hand, time slots are divided into packet slots separated by short 2-byte request slots. Figure 5.1 shows the packet and time slot structure. Each packet contains the unique identifier of the sender, the sender's logical clock and the current local schedule of the sender. A request packet contains the sender's ID and the number of the requested schedule slot.

The following data structures are maintained by each node:

- The visible neighbourhood N. This is a database of all nodes that are within range of the node. The node database is maintained according to received messages. It is assumed that a node whose message has been received is within range. Nodes are removed from the list and the schedule if no messages from them were received over a certain amount of time. With every node in the list the node's local schedule is stored as received in the last message of this node.

- The local schedule s_i. Nodes try to establish local collision free schedules. The schedule of each node consists of a number of schedule slots. A slot can be either empty ('e'), blocked ('b'), used by a node present in the visible neighbourhood ($j \in N$), or marked for own use ('o'). The local schedule is recalculated with every received message from the most recent schedules received from all nodes in visible neighbourhood.

A node can be in one of three states: *Listening*, *Requesting* and *Running*. The default is *Listening*. The node remains in the *Listening* state for a random number of time slots, while updating its local schedule and visible neighbourhood list from received messages. If the visible neighbourhood list is still empty after the time-out (no messages received), the node adds itself to the first slot of its local schedule and transmits a message containing this schedule. Otherwise the node sends out a request for the first empty slot in the current schedule and returns to *Listening* for a random time.

5.3.2 The DAOS algorithm

All nodes keep track of their neighbourhood (all nodes from which they recently received a message). Nodes also save the most recent schedules they received from their neighbours. A neighboured node is marked as *established* if a complete message with schedule has been received from this node and as *not established* if only a request has been received up to now (respectively if the last message from this node was a request).

If a message from a neighboured node cannot be received for an extended period of time (at least two schedule rounds), the node is removed from the database of neighboured nodes. Removal of a node from the database also invokes the removal of that node from all locally stored schedules (the locally stored copies of schedules received from other neighboured nodes).

The algorithm consists of two tasks, the transmitter task and the receiver task. The two tasks communicate indirectly through the node database, which has to be implemented as a multi-tasking save entity (i.e. a protected object).

Receiver task

The receiver task takes care of incoming messages. It only becomes active if a message is received and can therefore also be implemented as an interrupt handler. This is especially useful for implementations on microcontrollers, on which a distributed programming language might not be available. This is the description of the receiver task in pseudocode:

```
loop:
   Wait_For_New_Message;
   Message:=Retrieve_Message;
   Synchronise_Clock(Message.Logical_Clock);

   If type of Message is 'Request' then
      If Message.Sender is in Node_Database,
         remove Message.Sender from Node_Database;
      end if;
      Clear requested slot from all schedules in Node_Database;
      Create new entry for Node_Database:
        Node.ID := Message.Sender;
        Node.Schedule := Empty_Schedule;
        Mark requested slot in Node.Schedule as used by Message.Sender
        Node.Established := false;
      Store new Entry 'Node' in Node_Database.
   else if type of Message is "Message" then
```

```
      update Node_Database with
       (Message.Sender, Message.Schedule, Established := True;)
   end if;
end loop;
```

Transmitter

The transmitter task is more complex than the receiver task. This is a periodic task which becomes active at every beginning of a time slot or a request slot.

```
loop:
   Wait_For_Next_Time_Slot;
   Local_Schedule := Recalculate_Local_Schedule(Own_Slot);
   Current_Time_Slot := Calculate_Active_Time_Slot;
   If This_Node is in Local_Schedule, then
      State:=Run;
   else if not State=Listen then
      State:=Listen;
      Set Listen_Time to random number of time slots
   end if;

   Case State is
   Run:
      If Local_Schedule(Current_Time_Slot) = This_Node then
         If better slot with lower index available then
            with Probability of 33\%:
               Prepare_Request for better slot;
               State := Request;
         end if;
         with Probability of 95\%
            Transmit_Message(Local_Schedule);
      end if;
   Listen:
      If Node_Database is empty, then initiate:
         Own_Slot := first empty schedule slot;
         Local_Schedule(Own_Slot) := This_Node;
         Transmit_Message(Local_Schedule);
      else
         If Listen_Time=0, then
            Prepare Request for first empty slot in local schedule;
            State:=Request;
            Set Listen_Time to random number of time slots
         else
            Decrement Listen_Time
         end if;
   end case;

   Wait_For_Next_Request_Slot;
   if State = Request then
      Clear this node from local Node_Database
```

```
       and all locally stored schedules;
      Send out Request for prepared slot;
      Set Own_Slot to Requested slot
   end if;
end loop;
```

The two crucial functions called in the transmitter task are *Recalculate Local Schedule* and *Calculate Active Time Slot*. The nodes in the local node database are denoted $n_j \in N$; each node in the local database has a locally stored schedule s_j (the most recently received schedule from each neighbour) with schedule slots $s_{i,j}$. Schedules of nodes from which no schedule has been received yet are empty, i.e. all slots of that schedule are marked as empty slots. The calculation of the local schedule is described here by the function $s_i(k)$:

$$s_i(k) := \begin{cases} b: & \exists n_j \in N : s_{i,j} \neq e \wedge s_{i,j} \neq b \wedge s_{i,j} \notin N \\ j: & j = \min\{A_i\} \\ o: & i = k \wedge \exists n_j \in N : s_{i,j} = k \wedge \forall n_j \in N : (s_{i,j} = k \vee s_{i,j} = e) \\ e: & \text{otherwise} \end{cases} \tag{5.1}$$

with $A_i := \{k : n_k \in N \wedge s_{i,k} = k\}$

It can be seen from the definition of s_i that collisions between competing nodes are resolved by favouring the node with the lowest index. This is an invariant which can be equally computed by all affected nodes. The ”own” slot (the slot that a particular node requested last) is only assigned if at least one node in the 1-hop neighbourhood confirms this and if the slot is otherwise unused.

The active time slot is calculated recursively from the current logical time t and the current local schedule $s = s_1..s_l$. The schedule length l is assumed to be a power of 2.

$$\begin{aligned} \alpha : (N, S, N) &\mapsto N \\ (L(t), s_t, l) &\mapsto \alpha(L(t), s_t, l) := \begin{cases} 1 + (L(t) \bmod l) & | \quad s_{1+(L(t) \bmod l)} \neq e \\ \alpha(L(t), s_t, l/2) & | \quad s_{1+(L(t) \bmod l)} = e \\ 1 & | \quad l \leq 1 \end{cases} \end{aligned} \tag{5.2}$$

The initial call of the recursive function uses the maximal schedule length (must be a power of 2) as a parameter. The described mapping function has the

advantage that it does not return empty time slots if the first slot of the schedule is filled, thus increasing utilisation especially in the case of sparse schedules. Alternatively other time slot mapping functions can be used.

5.4 Properties of DAOS

Theorem 5.4.1 (DAOS converges to a collision free solution) *Assume the schedule length l is unbounded and the network topology is static. After a node initiates by starting to transmit its own schedule, surrounding nodes receive that schedule, select an empty slot and send out a request at a randomly chosen request slot. If two nodes occupy the same slot, all nodes that receive their schedule will assign this slot to the node with the lower ID. The node with the higher ID will receive that schedule from one of its neighbours and will give up the slot according to the definition of the algorithm. The node then reapplies for a new, free slot. As long as there are free slots in the schedule, every node will eventually receive a slot in the schedule.*

Every node locally constructs a schedule in which every time slot can only be occupied by one node within their 2-hop neighbourhood. A node can only use a slot if there is no competing node within a 2-hop neighbourhood. The slot is therefore collision free in the local view of the node. Schedules are continuously redistributed. If a collision is detected, it is resolved, and the resolution is redistributed within one schedule round. There is a possibility that an offending node can not receive an updated schedule due to the collision it causes or the collision caused by a neighbour. For that case, nodes statistically omit 5% of their transmissions. Over time, the probability of the node receiving the update approaches 1.

If a node detects an empty schedule slot with a smaller index than its own, it will request that slot. If the request does not cause collisions for any of its neighbours, the slot is assigned to that node.

Once all collisions are resolved, and all nodes have obtained a schedule slot, and for all nodes' schedule slots there is no (locally perceived) empty slot with a lower index, the schedules remain stable. Therefore, DAOS converges to a collision free, stable solution in a static network.

Theorem 5.4.2 (DAOS produces dense schedules) *A non-empty slot is either marked with a node identifier or marked as blocked. If a slot is blocked, there is at least one node within a 2-hop neighbourhood which is assigned to this slot. As long as the first schedule*

slot is used, the time slot mapping function 5.2 never returns an empty slot. Additionally, nodes apply for empty slots with a lower index. DAOS converges towards schedules that are densely packed for low indices. The first schedule slot will therefore be filled with a node. Therefore, after convergence, in any 2-hop neighbourhood there is always one transmitting node (except statistical omission).

5.4.1 Complexity of DAOS

The scheduling function is defined for schedule slots and therefore has to be repeated l times (for l being the schedule length). The complexity per slot is dominated by the all quantor and existence quantors, which iterate over the local neighbourhood. The local neighbourhood is bounded by the maximum graph degree. The schedule slot mapping function 5.2 has a maximum complexity of $O(\log l)$. The upper bound for the computational complexity is therefore $O(l \cdot \Delta)$. As the graph degree has to be bounded by the schedule length for correct function of DAOS, one can assume a constant upper bound for the computational complexity, which enables the design of a real time implementation.

5.5 Pruned Distributed Ad hoc Omnicast Scheduling

The algorithm described in the last section assumes a graph topological collision model (a node receives a message if and only if exactly one of its neighbours transmits). However, some real radio links (notably frequency modulated or phase modulated channels) have different characteristics (3.3.2, [43]). Typically, a node receives the message from a transmitting node if the ratio of signal strength of that node over noise and interfering messages in the local vicinity is greater than a certain threshold. Due to the strong attenuation of radio waves over distance (especially for omnidirectional links), there is only a small region where a collision occurs (the receiving node cannot decode any of the received messages).

A problem of the DAOS algorithm described in the previous section is the increasing schedule length for networks with high connectivity (high graph degree). Longer schedules have a negative impact on the performance. Round trip times grow locally and globally . An additional problem is that longer schedules take up large parts of the transmitted messages, wasting valuable

bandwidth. If the geometric/radiometric collision model is applied, it is possible to make use of the fact that collisions actually occur less frequently than the graph topological model suggests. In principal it is possible to pack schedules more densely by re-issuing slots taken by distant nodes to closer nodes. Messages sent by closer nodes will "overwrite" the messages sent by the more distant nodes sending in the same time slot. The perceived connectivity is quantitatively similar to a network of the same geometric layout, but with reduced transmission ranges. This technique is often referred to as *spatial reuse* [54].

The following algorithm implements spatial reuse on top of the DAOS algorithm. It requires the following assumptions:

- Symmetric links: If (in undisturbed conditions) node A can receive messages from node B, then node B can also receive messages from node A.
- Monotonically decaying signal strength over distance.
- A node receives whichever message sent out in its local neighbourhood which is received with the strongest signal (a small "collision zone" is acceptable, where no message is received if the n strongest messages have comparable signal strength. The collision zone is assumed to be small compared to the maximum range or the distance between nodes).
- The signal strength can be measured (directly or indirectly).

The modified algorithm is largely identical to the original DAOS algorithm. The two differences are in the collision resolution in the scheduling function 5.1 and in the slot request mechanism during the *listen* state. The new scheduling function is defined here:

$$s_i(k) := \begin{cases} b: & \exists n_j \in N' : s_{i,j} \neq e \wedge s_{i,j} \neq b \wedge s_{i,j} \notin N' \\ j: & j \in A_i \wedge \forall k \in A_i \setminus j : \sigma(j) > \sigma(k) \\ o: & \exists n_j \in N' : s_{i,j} = k \\ e: & \text{otherwise} \end{cases} \tag{5.3}$$

with $N' := \{n_j \in N : \exists m : s_m(k) = j\}$, $A_i := \{m : n_m \in N \wedge s_{i,m} = m\}$, and where $\sigma(j)$ denotes the signal strength of the last message received from node $n_j \in N$.

The main differences are that collisions are now resolved based on signal strength instead of node index. Since every node has different signal strengths for their neighbours, this decision is not identical any more on different nodes. In DAOS

the decision which node occupies a slot is coherent within a 2-hop neighbourhood around that node. In PDAOS the decision is localised. Nodes geometrically close to a node n_k will assign the slot to n_k, while other nodes within range or within a 2-hop neighbourhood may assign the same slot to other nodes that are received more strongly. This effectively shrinks the virtual neighbourhood of nodes in dense areas.

Derived modifications take into account that only nodes which appear in the final local schedule are used to mark blocked slots. Also, a slot is only marked 'o' (for own use) if at least one neighbour that appears in the local schedule confirms this slot.

The second difference is in the request mechanism. As before, nodes apply for empty slots within the schedule. As an extension, if there are no available empty slots, a node may apply for a blocked slot (with a preference for blocked slots at the end of the schedule). If no blocked slots are available, a node requests the slot occupied by the node with the lowest locally measured signal strength.

5.6 Properties of PDAOS

The PDAOS algorithm behaves in a similar way as the DAOS algorithm for low density networks, for which the schedule length is sufficient to accommodate for all 2-hop neighbourhoods in the network. Differences occur when the density increases over the limit given by the schedule length. The DAOS algorithm is not able to cope with this situation and converges to a stable solution that excludes some nodes. The PDAOS algorithm allows these nodes to apply for a slot which is used by a node with low signal strength (as perceived locally by the applying node), i.e. a node which is far away. PDAOS does not converge to a collision free solution in the sense of the graph theoretical model. However, in the geometric collision model, the occurrence of collisions is reduced by scheduling nodes with large spatial separation in the same slot. The amount of interference is kept low, and it is ensured that all nodes in a close proximity are able to receive the message.

It has to be noted that the behaviour of PDAOS for low density networks is similar but not identical to DAOS due to the different rules of collision resolution. DAOS uses node identifiers to resolve collisions; this is a spatial invariant for

all participating nodes. PDAOS uses locally measured signal strengths, which varies for different nodes. PDAOS is therefore less strict regarding collision resolution. Nodes only lose a schedule slot if all of their neighbours prefer a different node because it is closer. While it is difficult to formalise the exact behaviour, it is intuitively clear that schedules are limited to geometric cells which scale according to the local density. The geometric size of a cell is linked to the network density via the bounded schedule length - a cell cannot contain more nodes than the maximum schedule length.

By excluding nodes which are further away, the resulting schedule for a dense network is similar to the schedule of the same network, but with reduced communication range. Messages from distant nodes are "overwritten" by messages from closer nodes, which are scheduled in the same slot. As long as the closer node sends, it is as if the more distant node cannot be received at all. The behaviour is not exactly identical - if a slot is not filled locally, the message of the more distant node will be received, which would not be the case in a network with reduced communication range.

5.7 Discussion

This section compares the performance of DAOS and PDAOS with regard to omnicast round trip time, reconfiguration times and start up. Results are given for theoretical considerations and from a real time simulation.

5.7.1 Upper bounds

A theoretical analysis of the omnicast problem (4.3.2, [41]) revealed an upper bound of $2n - 2$ for networks with n nodes. Other upper bounds have been presented in [17] [15]. An upper bound for the DAOS algorithm can be given as $O(\frac{\gamma d}{2})$, where γ is the maximum size of any 2-hop neighbourhood ($\gamma := \Delta G^2$):

The diameter of a 2-hop neighbourhood is at most four. In a connected graph it is possible to find a chain of overlapping 2-hop neighbourhood subgraphs that minimally cover the diameter of the graph. This requires $d/4$ subgraphs. DAOS only considers nodes in a local schedule that are contained in the 2-hop neighbourhood of a node. It follows that completely executing a schedule once

takes $O(\gamma)$ time slots. In the worst case, after a local schedule has been executed twice, all nodes within this 2-hop neighbourhood have locally solved omnicast, including the nodes that overlap with neighbouring subgraphs. This process has to be repeated at most $d(G)/4$ times to spread all information along the diameter. Since the diameter is the longest shortest path across the graph, this implies that omnicast can be solved in at most $O(\frac{\gamma d}{2})$ time steps. This assumes that the maximum schedule length of DAOS can accommodate at least γ nodes. If this is the case, γ can be substituted by the maximum schedule length l.

Both γ and l are linked to the graph degree Δ. In the general case, an upper bound for γ is Δ^2. However, in network graphs that are derived from a two- or three-dimensional embedding, the size of a 2-hop neighbourhood is typically proportional to the graph degree. As has been shown in 4.4.2, for 2-dimensional homogeneous large networks, the graph degree can be approximated as $\Delta' = O\left(n\frac{r^2}{A}\right)$. Similarly, the size of a 2-hop neighbourhood is $\Delta' = O\left(n\frac{(2r)^2}{A}\right) = 4\Delta'$, which is typically lower than Δ^2. For convergence of DAOS, the schedule length l has to be chosen accordingly $l > 4\Delta$ for a given network. It is obvious that short schedules are preferable. This means that DAOS works better for networks with a low degree. If the network becomes denser, a short schedule length might lead to nodes being excluded from the schedule. The algorithm still reaches a stable state, and excluded nodes still receive updates, but cannot send any updates themselves.

PDAOS is able to deal with 2-hop neighbourhoods which are larger than the schedule length, by virtually reducing the neighbourhood according to signal strength. It is therefore advisable to choose the schedule length as short as possible, while still preserving good connectivity for the given swarm configuration. PDAOS requires the schedule length to be long enough to accommodate all nodes in the closest proximity, so that these nodes maintain good connectivity with the rest of the graph. In the case of homogeneous 3-D configurations, nodes typically have 12 neighbours in close proximity, or 6 neighbours for 2-D configurations. The schedule length does not have to accommodate nodes that are 2 hops away, as blocked slots can be rescheduled to closer nodes. A schedule length of 16 slots is therefore sufficient for most homogeneous swarm configurations. A short schedule length will enforce pruning of schedules, which effectively lowers the upper bound for omnicast. A further advantage of PDAOS is that it continuously updates the schedules

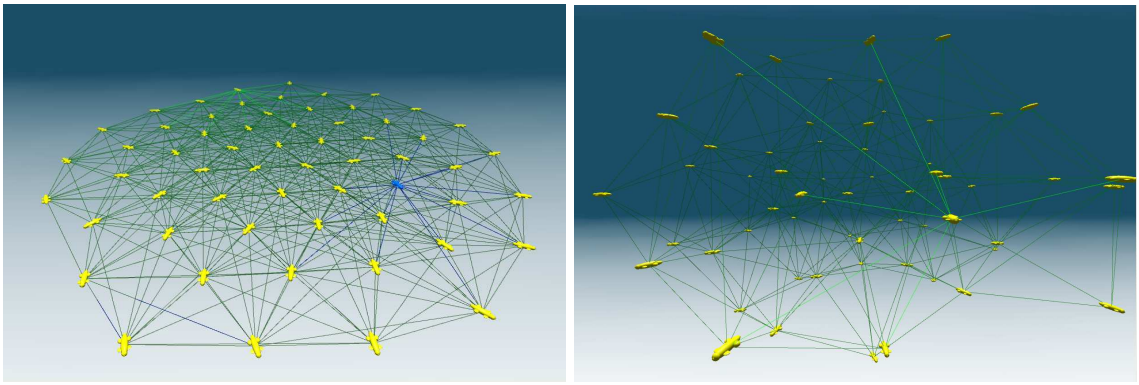

Figure 5.2: Simulation of a swarm of submarines in 2D configuration (left) and 3D configuration (right). The lines between submarines indicate that they are within communication range.

according to the currently received signal strength of a node's neighbours. If a node moves throughout the network, the schedules are continuously updated, and only rarely does a node lose its slot and has to reapply. The next section goes into more detail of the actual performance of both algorithms.

5.7.2 Swarm simulation

The distributed TDMA scheduling algorithm has been implemented both in a simulation environment and on a hardware system. Until now the hardware system could only be tested with up to four nodes, which could indicate the principal similarity between simulation and reality, but does not allow in-depth analysis of the scheduling at larger scales. For small networks with size 4, the algorithm behaved identically in simulation and reality for all possible configurations.

A simulation environment has been implemented in Ada 2005 which provides a 3D arena for simulated submersibles. The simulation runs in real time and is multi-threaded. Each submarine consists of several tasks and encapsulated data structures. Access to simulation parameters and other submarines is only possible over simulated sensory equipment. The simulation implements encapsulation, i.e. the access routines to simulated hardware are identical to the access routines to the actual, real hardware. This means that the tested algorithms

can be directly ported to hardware without any changes to the implementation, and the algorithms should not experience any principal differences between simulation and real world. However, no simulation can fully emulate the real world, and there will always be subtle differences.

The submersibles are subject to a simplified force-based dynamics simulation. They can perform distance and bearing measurements on all their neighbours within their sensing range (set to 4 metres for the following experiments) at an update rate of 2 Hz. This allows the submarine to apply a primitive swarming rule using simulated springs with a positive neutral length. The swarm will quickly converge towards a configuration with approximately equal distances between neighboured submarines, which leads to a triangular arrangement, if submarines are restricted to a plane, or a tetrahedral configuration in the 3D case (Figure 5.2). It has to be noted that submarines try to equalise the distance between *all* their neighbours within their sensing range, which means that the minimum distance (and the average distance to closest neighbours) can be smaller than the length of the virtual spring. While not critical for the following evaluation of the communication system, this has to be kept in mind when considering the connection between measured average distance of submarines within sensing range and the average degree (or number of neighbours) of the network.

To simulate the communication system, the range and collision model has been implemented to closely match the behaviour found in the experiments (figure 3.15). Transmission durations are simulated in real time, and message delivery is performed by the simulation arena according to the modelled link distance and competing messages, to detect possible collisions. While not being a physical simulation of wave propagation, it closely reproduces the results in message delivery found in physical experiments with previously designed and tested 122 kHz digital long-wave radio modules [43].

The following experiments were done running the dynamics simulation, the simple swarming rules and the distributed scheduling algorithms (DAOS and PDAOS). The swarm is initialised with a virtual spring length of 1.5 metres. The sensing and communication range is set to 4 metres. After start up, the submarines quickly arrange themselves in a round disk-shaped (2D) or ball shaped (3D) configuration, equally spaced in a triangular configuration. Due to the ratio of average distance between subs and the maximum communication

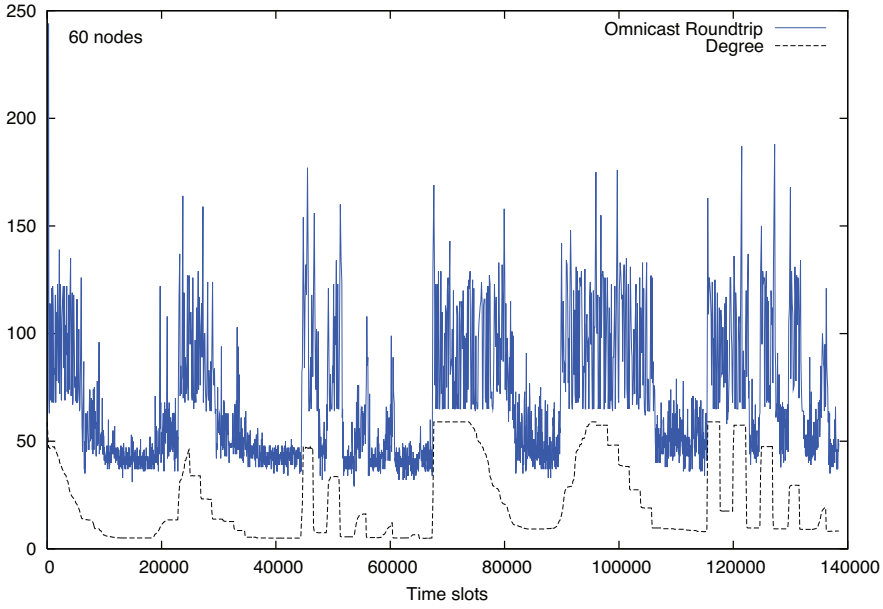

Figure 5.3: Full simulation run of DAOS with 60 nodes.

range this results in a fully connected network for all presented networks up to 60 nodes.

The communication network immediately starts building up local schedules, and after less than one minute all submarines are included in the schedule and exchange information. After the initial stabilisation phase the length of the virtual spring applied in the swarming rules is dynamically changed in various patterns. This causes the swarm to expand or shrink slowly and the network to become less or more densely connected. The scheduling algorithm and the swarming behaviour are never stopped for the duration of the experiment. During this process the following parameters are measured:

- Average distance. This is the average distance between a submarine and all submarines within its sensing range, averaged over all submarines. This value is closely linked to the virtual spring length.
- Average degree. This is the number of visible neighbours (submarines within communication range), averaged over all submarines.

- Omnicast Performance. This number indicates the required number of time slots to achieve *Omnicast* (full information exchange between all nodes). In contrast to the static case described in the definition of *Omnicast*, this number is calculated in a distributed way as follows. Every submarine maintains a list of counters $V = v_1, ..., v_n$ for all submarines in the swarm, which is sent out with every message. With every received message (containing the external counter list $x_1...x_n$), every local entry v_i is updated by $v_i(t + 1) = \max(v_i(t), x_i)$. If all entries $v_1...v_n$ are equal, the local own counter is incremented by 1. The time between two counter increments reflects the average duration of omnicast in the current schedule, since every node has to receive every other node's update to be able to increment its own counter.

A sample run can be seen in figure 5.3. The plot shows the real time performance of DAOS (red) for a dynamically changing swarm of varying network graph degree (green). The round trip time is the measured duration between omnicast counter increments. The simulation run starts with a 2-D configuration and then switches to a 3-D configuration and repeats the changes in density (also refer to the images in figure 5.2). The change to 3-D occurs at approximately t=68000.

5.7.3 Results of simulated scheduling

As expected the average degree of the network is closely linked to the average distance, as figure 5.4 shows. The plot in figure 5.5 shows the omnicast time (the time it takes to perform a full exchange of all local information) over the average distance between submarines. Outliers that are significantly higher than the majority of plot points are due to a reorganisation of the schedule, as the network changes. This usually involves nodes temporarily falling out of the schedule, which results in all other nodes not being able to increment their counter. In a real scenario submarines would not be affected by lacking updates from only a few units. Units not represented in the schedule are still able to listen to all ongoing information, which means they will not be lost. As can be seen in figure 5.3, these spikes due to reorganisation are typically lower than 200 time slots for 60 nodes.

Up to a distance of approximately 2.3 metres the performance is quite stable at around 100 time steps. This is not surprising, considering that the average degree is still high enough for most nodes having a 2-hop neighbourhood which contains

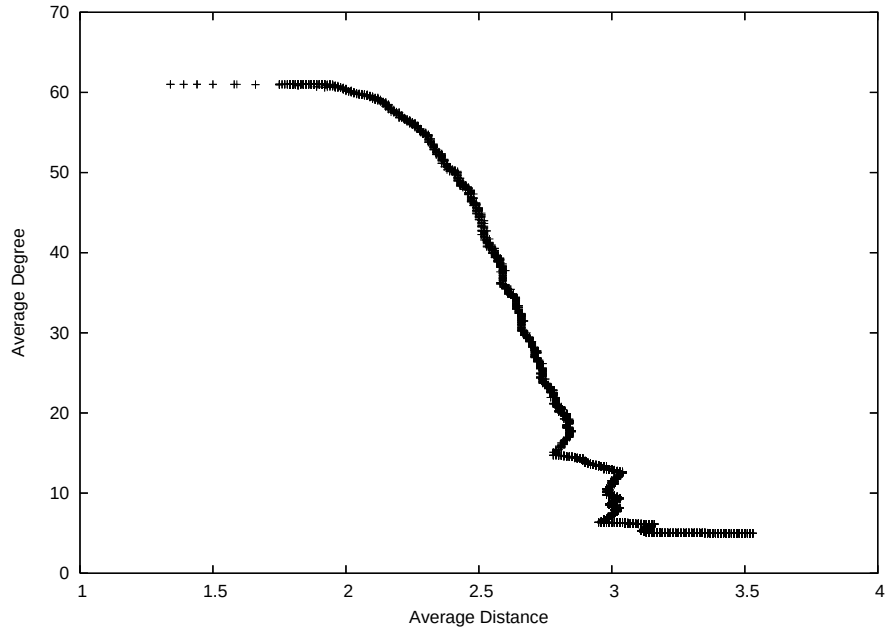

Figure 5.4: Relationship between average distance and average degree for 62 nodes restricted to a plane

all other nodes. This means that there is no concurrent communication, and the schedule is basically identical to the schedule of a fully connected network. The reason why the omnicast time is higher than the number of nodes lies in the stochastic omission of transmissions as described earlier. Omitted messages slightly delay the global information exchange by up to the duration of one schedule round. For a fully connected network a schedule round is as many time steps as there are nodes in the network, rounded up to the next power of two (in this case 64 time slots).

As the average distance increases, the network connectivity goes down. This allows for more concurrent communication, since 2-hop neighbourhoods no longer overlap. The omnicast time drops to less than half, with an average of 45 time steps, and reaches the best performance as the average distance approaches the maximum sensing range. It might appear counter-intuitive that the best performance is reached when the swarm reaches its maximum size, but there is a logical explanation as indicated before.

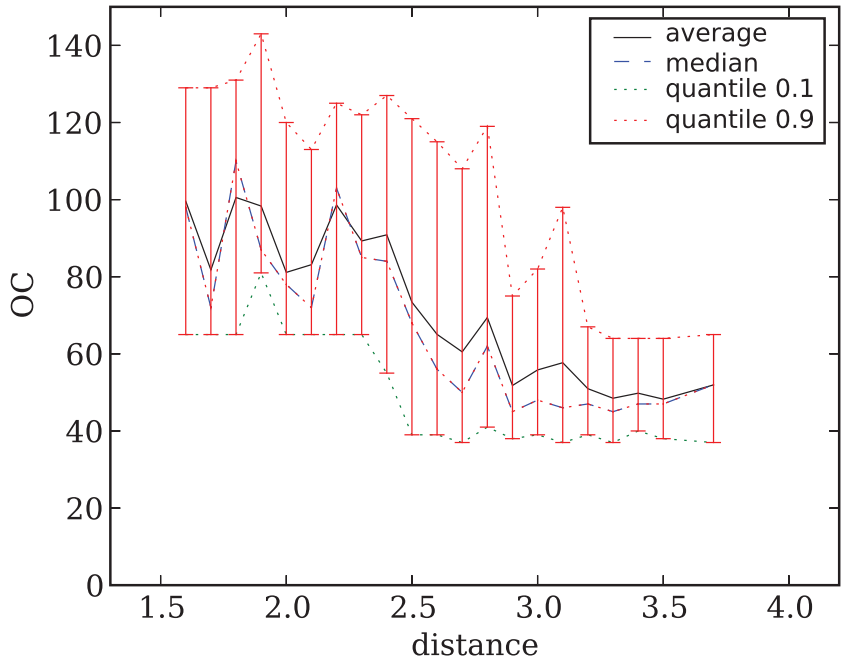

Figure 5.5: Average distance versus omnicast performance for 60 nodes in a plane

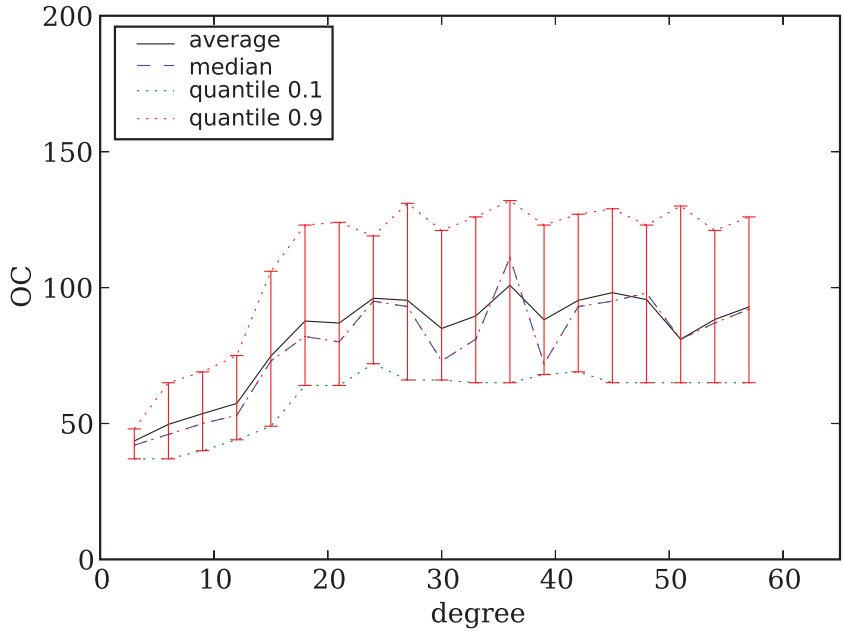

Figure 5.6: Average degree versus omnicast performance for 60 nodes in 2D and 3D combined (DAOS)

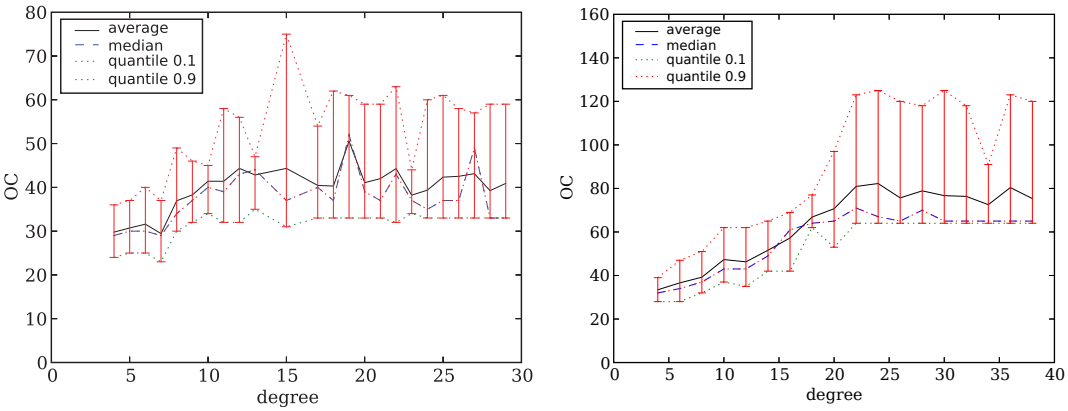

Figure 5.7: Average degree versus omnicast performance for 30 nodes (left) and 40 nodes (right) for DAOS

Figure 5.6 shows the same experiment plotted against the average degree of the network. The left side of this plot roughly corresponds to the right-hand side of the previous plot. The best performance is reached for an average degree of 5. This describes the swarm being spread out to almost the maximum communication range, with center nodes having 6 neighbours, and edge nodes having down to 3 neighbours. A low degree allows for concurrent communication in several parts of the swarm at any time. The diameter of the network graph in this configuration is 7. Even though not visible in the plots of the global performance, it is obvious that the local schedule lengths are much lower for low degrees (the upper bound is the size of the 2-hop neighbourhood). This results in quicker updates for nodes from their direct neighbours, which is important for swarm control.

Figure 5.7 shows similar results for networks with 40 and 30 nodes respectively. In all cases the maximum performance is reached for the largest possible expansion of the network without disconnection.

Both theory and simulation results indicate that a low degree is preferable over networks of diameter of 2 or less, mainly because those networks inhibit concurrent communication and inherently do not scale well. Inversely, this means that for a given geometric swarm size, the communication links should be just long enough to reach the nearest neighbours of each swarm member, but not

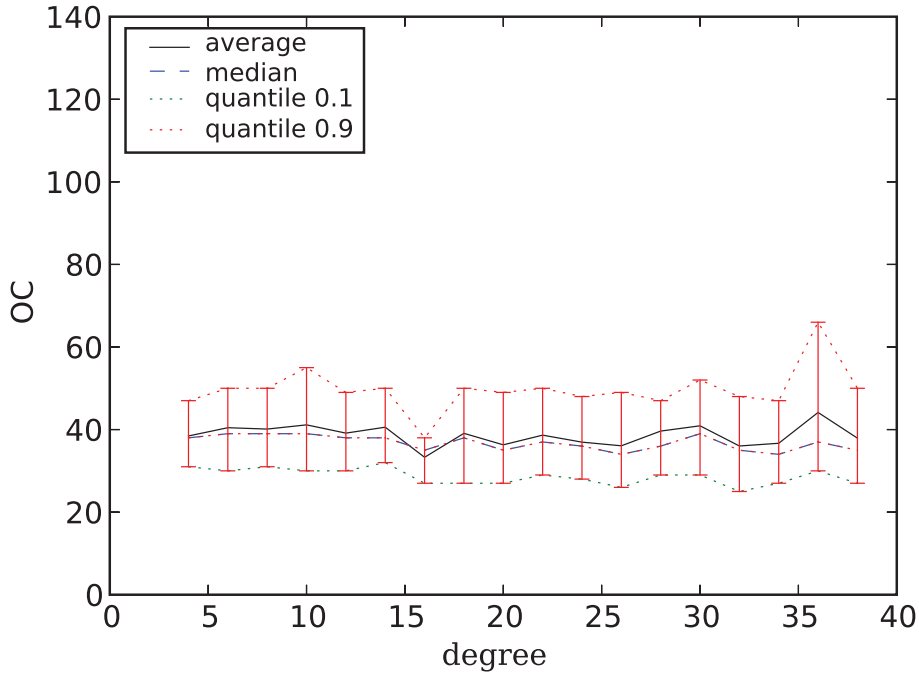

Figure 5.8: Average degree versus omnicast performance for 40 nodes, PDAOS algorithm

longer. Short range links are better with regard to local update frequency, global information exchange and, of course, also with respect to power consumption.

As expected, the collision-avoiding DAOS algorithm performs well for low graph degrees, but loses performance for high degrees (figure 5.7 and also figure 5.9). This is due to the fact that for high connectivity only one node can send per time slot, or otherwise messages would collide. The PDAOS algorithm performs equally well over the full range of network densities, in both 2-D and 3-D (figure 5.8). The average performance is around 40 time steps, which coincides with the number of nodes. There is very little variation in the performance, as the 10% and 90% quantiles indicate. It should be noted that the PDAOS algorithm achieved this with a schedule length of only 16, while the DAOS algorithm required 64 slots to be able to fit all nodes into the schedule during periods of high connectivity. This means that the message size overhead in PDAOS can be greatly reduced.

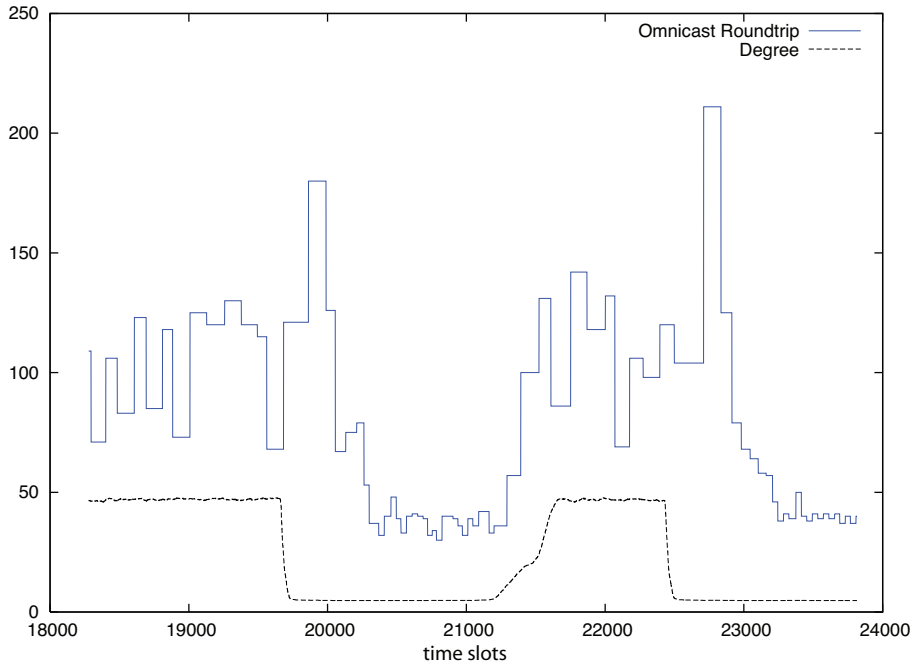

Figure 5.9: Step response in a simulation with 60 Nodes (DAOS)

5.7.4 Dynamic response

One of the most important aspects of a scheduling algorithm for swarming is the responsiveness to dynamic changes in the network. The communication system must not break down when the swarm changes its configuration, or else the coherence can not be maintained reliably.

Both algorithms have been subjected to dynamic changes in the network topology and geometry. One test involves the step response of the omnicast performance following a sharp change in density.

The DAOS algorithm responds so quickly that the performance can be maintained during the adaptation of the schedules to the new topology (figure 5.9). In the case of the network thinning out, there is a short period of reduced performance. This is because the schedule of the dense network is still active after the transition (i.e. one node at a time sends), but the network now has a worse connectivity. After a brief period of time the change is detected and the schedule adapted, leading to a much shorter round trip time. The first adaptation is then followed by further optimisations.

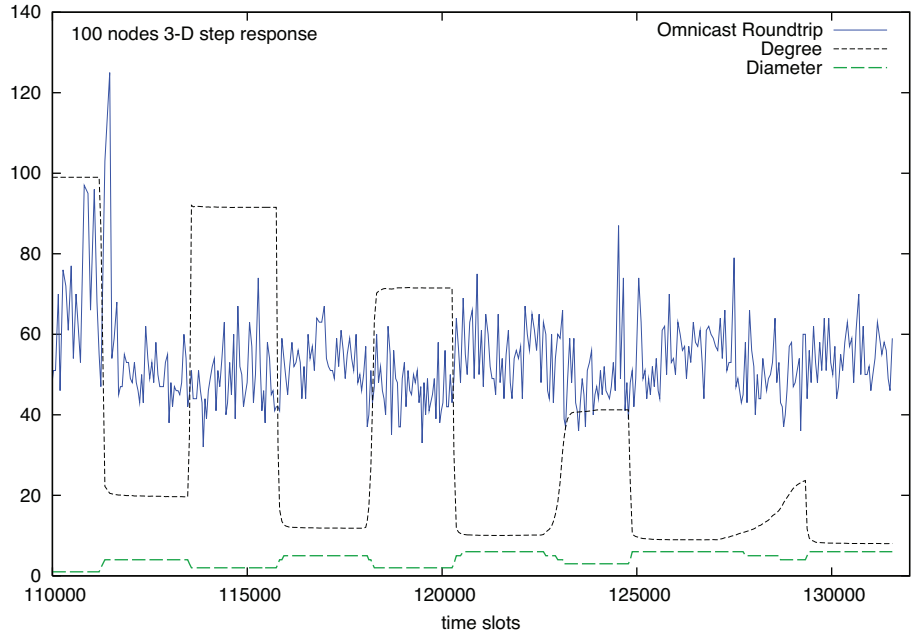

Figure 5.10: Step response in a simulation with 100 Nodes in a 3D configuration (PDAOS, schedule length 32 slots)

In the opposite case of the network becoming denser, the change can be detected more quickly, because nodes now receive more messages from new neighbours containing their schedules. Changes can be immediately made, and the performance immediately reaches the best possible performance for the new configuration. The variation in performance for fully connected networks can be explained by the stochastic omission of transmissions by nodes. This is necessary to detect messages that might be hidden by own transmissions.

Extended step response experiments have been carried out with the PDAOS algorithm (figure 5.10) As can be seen, the performance of PDAOS is largely unaffected by dramatic changes in the network topology. The average performance is around 50-60 time steps, which is below the number of nodes in the network. This result is not surprising, since the PDAOS algorithm packs schedules more densely in the case of high network degree, which means that the schedules resemble the schedules achieved during periods of low degree. The geometry of the swarm mainly changes the scale, but only changes slowly the neighbourhood

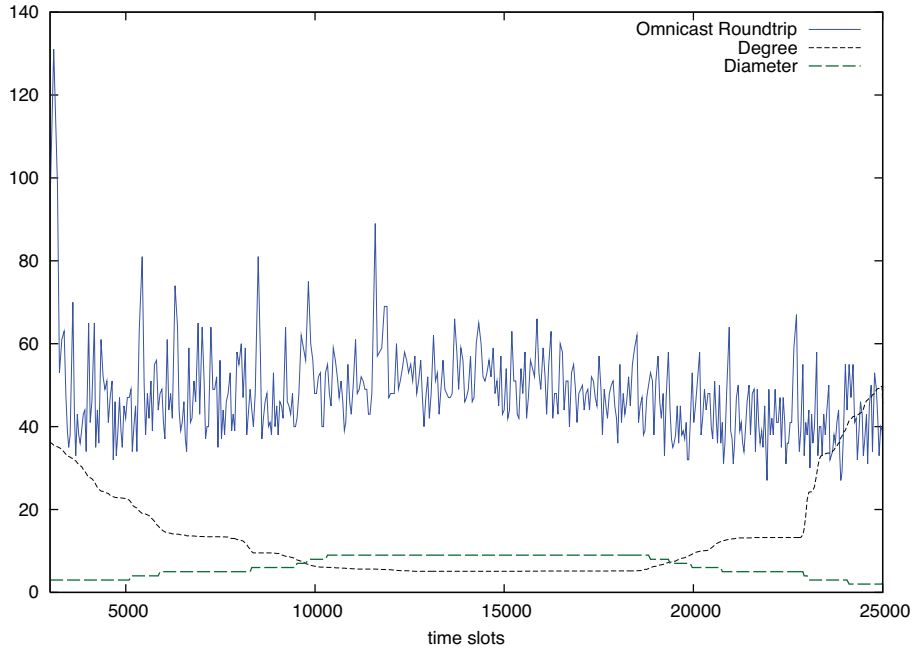

Figure 5.11: Omnicast performance for 64 Nodes in a 2D configuration with slowly varying density (PDAOS, schedule length 16 slots)

relations, so that the already established schedule only requires minor changes. There is a slight effect that the performance profits from lower graph diameters - the omnicast round trip time is slightly reduced for graphs of high degree and low diameter.

A final experiment was conducted where the graph density was changed slowly. The results are shown in figure 5.11. Again, the performance of PDAOS is mostly unaffected by the changes in the network. It is also again visible that the performance is slightly better for low diameters.

The most severe change to the schedule of each node occurs during start up, when no previous schedule exists. The start up is therefore the worst-case test scenario for adaptability. Figure 5.12 shows the behaviour during the first 1000 time slots for the DAOS algorithm. During initialisation the number of collisions is high while the schedules are established. The first omnicast round trip (global information exchange) has been completed after less than 100 time slots. After 150 time slots the performance is already at its optimum and remains stable. The

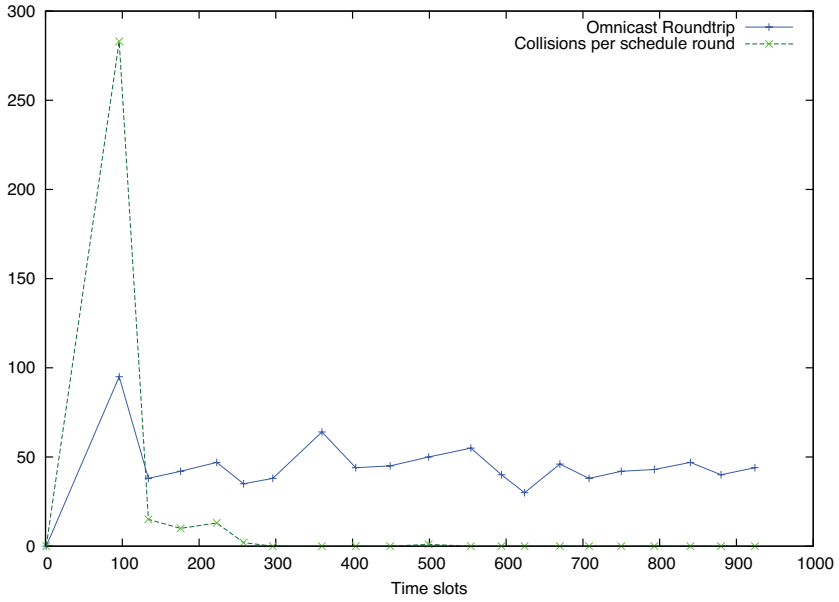

Figure 5.12: Start up of a network with 45 nodes (DAOS).

last collisions are resolved after less than 300 time slots. At a typical sending frequency of 10 Hz the communication system has stabilised after less than 30 seconds.

The dynamic response of the DAOS algorithm together with occurring collisions can be seen in detail in figure 5.13. The network changes its connectivity first slowly and then rapidly from a network graph degree of approximately 44 (fully connected) to a degree of 6 (fully spread out). The change in degree is shown by the black curve. The blue curve shows the measured omnicast round trip time, measured by the distributed counter as described earlier. Occurring collisions are counted per schedule round and plotted in green. A collision is counted for each node that has two or more transmitting nodes within its range. Collision frequency can quickly rise in dense networks, because many nodes are affected if two nodes transmit at the same time. In networks of low degree only few nodes are affected, hence the number of collisions is typically lower.

After a short delay the scheduling algorithm detects the change in network topology and reschedules. The rescheduling happens quickly enough so that the effect on communication performance is small. The step from high degree to

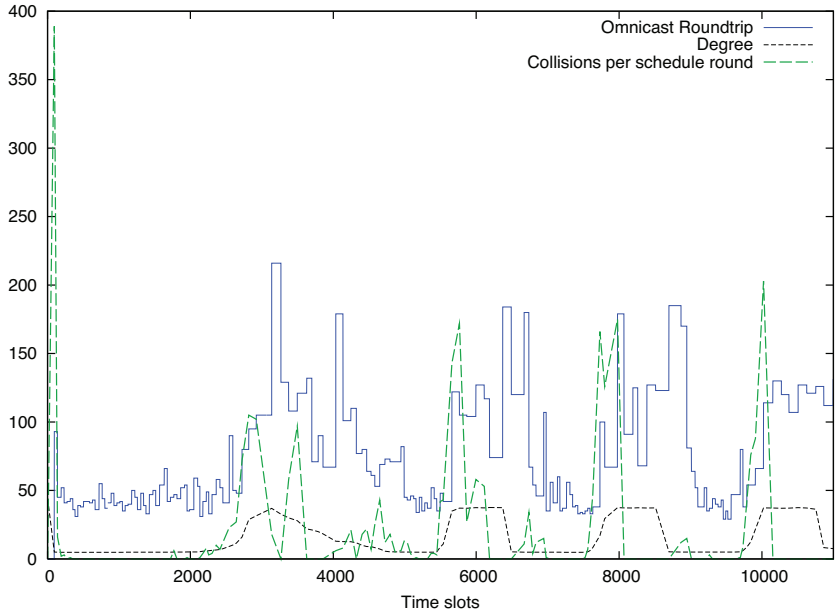

Figure 5.13: Start up and step response to a change in connectivity for 45 nodes (DAOS)

low degree causes the round trip time to double for one schedule round. This is because the schedule for high connectivity (only one node per time slot) is still active, but the lower connectivity inhibits message exchange. Subsequently the schedules get optimised over the next 5-10 schedule rounds, gradually improving performance. The initial delay of about 2-3 schedule rounds is due to the time-outs for local removal of disappearing nodes. Nodes are removed from a local schedule if they are not received for more than 2 schedule rounds. During rescheduling a few collisions may occur if several nodes apply for the same time slot. Collisions are quickly resolved. Once a stable state is reached there are no further collisions.

The change from low degree to high degree is more easily detected due to the occurring collisions in the denser network. Nodes that were previously transmitting in the same schedule slot are now closer and within each other's range. Once these nodes detect a collision in their received schedules, they immediately apply for a free slot and are quickly rescheduled. The scheduling

algorithm adapts to the new topology very rapidly and reaches a new collision free stable state.

5.8 Experiments

Experiments were carried out in hardware, using four digital long-wave radio modules as described in chapter 3. The DAOS algorithm was implemented to run on the *Atmel ATMega32* processor on the long-wave module. The PDAOS algorithm could not be implemented due to the lack of signal strength sensing capabilities on the existing hardware. This is planned for future work.

The long-wave modules offer frames of up to 16 bytes. A packet was defined as a number of frames, where the first frame contains state information and the schedule, and all following frames contain user data. The number of frames per packet can be chosen to meet the application. The length of a time slot is defined by the number of bytes per packet divided by the data rate. The schedule length was set to 12 slots for the following experiments. To provide data integrity and robustness to the scheduling algorithm, the first frame of a packet (containing a status byte, the sender ID, time stamp and the schedule) is protected with a checksum. The user data block is unprotected to avoid wastage of bandwidth and processing time. If required data protection, such as cyclic redundancy check, can be added by higher protocol levels. A request frame is 4 bytes large, containing a status byte, the sender ID, requested slot and a checksum. A short time gap between the request slot and the main time slot provides robustness against synchronisation errors.

Clock synchronisation of the logical clock was achieved by applying a maximum rule to the internal clock and the received time stamp. Sufficiently accurate synchronisation of the time slot metronome timer was achieved by averaging the internal clock source with the arrival times of received messages. After power up, the time slot clock sources were found to synchronise up to an acceptable level within a few seconds.

After power up a node waits a random time for received messages from other nodes. If such messages are received, it synchronises its clock and formulates a request as outlined in the algorithm description. If no messages could be received for an extended period of time, the node enters itself into its own schedule and

starts broadcasting. This mechanism ensures bootstrapping at the first power up of the network and also provides reliable network restarting in case of a network breakdown.

As a first experiment one node was powered up. As soon as this first node started transmitting, a second node was powered up within range of the first node. After the initial waiting period the second node was quickly and reliably integrated into the schedule. Furthermore, adding a third and a fourth node resulted in quick adaptation of the schedule to include the new nodes. After adding all nodes a node was switched off. After 3 schedule rounds the remaining nodes successfully removed the missing node from their schedule and stabilised on a new schedule without the removed node. Networks were successfully grown or shrunk by switching nodes on or off.

In a second experiment nodes were rearranged to form all possible connected network graphs for four nodes. The behaviour of nodes was found to be identical to the simulation.

5.9 Summary

An efficient TDMA scheduling algorithm DAOS has been presented and discussed. By taking advantage of the collision behaviour typically found in phase and frequency modulated radio systems, a modification to the original algorithm (PDAOS) has been furthermore presented and discussed. Both algorithms are able to rapidly adjust to changes in the network topology and achieve a very good performance well within the theoretical upper bounds. While DAOS requires large schedules being sent with every message and suffers lower performance for dense network, the modified PDAOS algorithm maintains almost constant performance and only requires short schedule lengths as low as 16 slots.

Chapter 6

Conclusions

The previous chapters presented a small submersible robot suitable for swarming applications, underwater communication links, explored the backgrounds of group communication in swarms and presented two ad hoc TDMA scheduling algorithms. This chapter summarises the results and shows how the individual parts can be put together to a coherent system.

6.1 Results

Communication was identified as a key requirement for swarms of underwater robots. It was shown that despite the difficulties of underwater communication, there are various possibilities for implementing short range communication links with reasonable bandwidth. Experimental results were presented for optical communication, radio communication and communication by electric conduction (return current). The design and implementation of prototypes for a long-wave radio module and an optical module showed that space and power requirements for these communication modules are minimal. This is important for miniaturisation of autonomous submarines. Chapter 2 argued that

a small size of submersible robots of 50 cm and 5 kg is an important factor for the practicality of a swarming system. Ranges of more than ten metres were achieved with very low power requirements. A communication range of 10-20 m is considered sufficient with regard to realistic applications of an autonomous swarm.

Robotic swarms have particular requirements regarding communication. The all-to-all omnicast communication mode has been identified as a suitable mode in chapter 3. Upper and lower bounds have been presented; it was proven that omnicast can be achieved in linear time, more precisely in $2n - 2$ time steps for n nodes. A geometric approximation indicated that it is advisable to use relatively short range links in order to minimize the time it takes for all nodes to receive a global update from all other nodes. Also with regard to local information exchange short links are beneficial. The theoretical considerations in chapter 3 assumed knowledge of the network topology and also assumed that an optimal solution can be found. However, a practical implementation of an ad hoc TDMA algorithm presented in chapter 5 confirmed in a simulation that the performance is best for networks with low degree and large diameter, i.e. networks that use short range links as opposed to long range links that span the entire network.

Two TDMA scheduling algorithms were introduced, DAOS and PDAOS. Simulation results demonstrate that both algorithms have very good performance for solving continuous omnicast. Typically the required number of time slots to achieve global information exchange is less than the number of nodes in the network, especially for networks with low degree. Start-up and adaptation to changing topologies is extremely quick - good performance can be maintained even during radical changes in the network structure. The DAOS algorithm converges to collision free solutions and is based on a more general graph-topological network model. The PDAOS algorithm assumes a geometric/radiometric network model, which is derived from experiments with the digital long-wave radio modules presented in chapter 3. It is necessary that the signal strength of received messages can be measured. If the requirements can be met, PDAOS offers better performance for networks with high degree by employing a spatial reuse technique. Furthermore, it adapts better to changing networks - the performance is minimally affected even during dramatic changes in network density. An additional side effect of PDAOS is that the schedule length can be reduced, which means that there is less communication overhead per package.

This thesis offers all required parts to implement a communication system which is suitable for a swarm of small submersible robots. All components have been discussed in detail and tested in experiments or simulations. Furthermore, limitations of the components with regard to underwater swarms have been discussed. As a demonstration that ad hoc networking can be achieved, the DAOS algorithm was implemented on the long-wave radio module presented in chapter 3. For small networks the behaviour is identical to the simulation. Experiments with larger networks could not be conducted; however, it is reasonable to assume that the simulation results of networks with over 100 nodes also apply to a real network.

6.2 System integration

As discussed previously, due to severe constraints posed by the underwater environment, there is no perfect solution to all communication needs. A realistic implementation of an underwater swarm robot therefore needs to include several different communication modalities and has to make use of synergetic effects. Even on a submarine as small as 50 cm it is easily possible to include a long-wave radio module, a multi-transmitter optical communication module and a return current communication system. Both the return current system and the optical system can be used to detect nearby obstacles and give information about the relative position and pose of other submarines. If combined with an acoustic relative positioning system as described in [28][27], the difference in time of flight between radio waves and acoustic waves can be used to estimate distances between robots. Additionally, if e.g. the long-wave communication system uses DAOS for collision free TDMA scheduling, the schedule can also be applied to acoustic emissions in the active acoustic positioning system. This ensures that locally only one node emits acoustic signals, which simplifies bearing estimation. Vice versa, positioning information obtained from acoustic measurements can be used to improve the accuracy of the PDAOS algorithm by combining the measured signal strength that is required with the distance to the respective node. There are many more examples of beneficial synergetic effects if the different subsystems are combined intelligently. One more example is the clock synchronisation performed by the DAOS algorithm running on radio or optical

links. Synchronised clocks are very useful for a range of applications, i.e. phased acoustic arrays spread over multiple vehicles.

6.3 Outlook and future work

Some parts of this thesis opened new questions that motivate further investigation. The implementation of a complete return current communication module will require additional research. Identifying suitable modulation schemes and obtaining more data on the return current effect and electrical noise in seawater and in other specific environments could be of interest.

The implementation of the DAOS and PDAOS algorithms on all available communication channels should be pursued further. Experiments in the ocean with a larger number of nodes would be an interesting next step. High frequency acoustic communication seems promising with regard to range, bandwidth and interference properties; more research in that area might reveal if acoustic communication can be scaled to large swarms.

Finally, the tight integration of the described communication systems into swarm control, sensing applications and navigation is a wide open research field. This thesis only described how the given communication systems can deal with changing network topologies. A swarm of mobile robots can apply the converse way of changing the network topology actively to achieve better communication performance or reliability. An example that has been mentioned is the creation of a high speed optical backbone network, using dedicated submersibles with a set of fixed narrow beam transmitters and receivers. The question of how these units can be positioned while the swarm is moving opens up interesting control problems. A further example is the aforementioned use of data couriers, i.e. robots that collect large amounts of information in different locations of the swarm and travel around to transport and redistribute this data.

All the described developments culminate in the final design of the *Serafina Mk II* submersible. More challenges will arise during the development of the final system. If all the communication systems that were described and other sensors can be combined to one integrated system, it is expected that swarming with a large number of submersible robots will be possible. It is hoped that one day such a swarm will be useful for oceanographic sciences.

Glossary

Ada 2005 Current revision of Ada, a high level real time programming language, 120

ALOHA OSI layer 2 network protocol, 104

AUV Autonomous Underwater Vehicle, 29

AWGN Additive White Gaussian Noise (channel), 80

BlueTooth A short range wireless network standard for peripheral devices, 44

broadcast one-to-many or one-to-all communication (see also multicast), 88

Butterfly graph A special graph of radius 1, 96

CDMA Code Division Multiple Access, 104

convergecast many-to-one or all-to-one communication, 88

CPU Central Processing Unit. The core component of a microcomputer, 34, 39

CSMA/CA Carrier Sense Multiple Access/Collision Avoidance, 104

DAOS Distributed Ad hoc Omnicast Scheduling. A deterministic TDMA scheduling algorithm, 108

DSSS Direct Sequence Spread Spectrum, 105

FDMA Frequency Division Multiple Access, 105

FPGA Field Programmable Gate Array. A microchip which can be programmed to implement arbitrary logic in hardware, 33

GPS Global Positioning System. A satellite based installation that allows small receivers to determine their longitude and latitude with 10 m accuracy, 30

Hamiltonian graph A graph which has a path which visits every node of the graph exactly once., 94

IrDA Infrared Data Association. A protocol for wireless data communication via modulated infrared light, 47

kbps kilobit per second, 47

LAN Local Area Network, 44

LED Light Emitting Diode, 47, 71

MIMO Multiple Input Multiple Output, 104

multicast one-to-many communication, 88

omnicast many-to-many or all-to-all communication, 88

OSI Open Systems Interconnect. A standardised layer model for communication networks., 89

PDAOS Pruned Distributed Ad hoc Omnicast Scheduling. A variant of DAOS, 115

RISC Reduced Instruction Set Computer, 39

RMS Root Mean Square, also known as quadratic mean, measures the magnitude of a varying quantity, 54

ROV Remotely Operated Vehicle, 29

Sensor Network an installation of a large number of relatively simple sensor nodes connected by a communication network, 19

Serafina Name of a 50 cm large submersible designed for swarming applications, 34

Serafina Mk II see also Serafina. Second version of the swarming submersible robot., 39

Swarm a group of individuals that aggregate and travel in the same direction, 20

TDMA Time Division Multiple Access, 25, 103

TTL Transistor-Transistor Logic, 85

UART Universal asynchronous receiver / transmitter. A type of serial communication interface, 73

unicast point-to-point communication, 88

UUV Unmanned Underwater Vehicle, 27

Bibliography

[1] Marcel Babin and Dariusz Stramski. Light absorption by aquatic particles in the near-infrared spectral region. *Am. Soc. of Limnology and Oceanography*, 47(3):911–915, 2002.

[2] J. W. Bales and C. Chryssostomidis. High bandwidth, low-power, short-range optical communication underwater. In *Proc. 9th Int. Symp. on Unmanned Untethered Submersible Technology*, 1995.

[3] Reuven Bar-Yehuda, Oded Goldreich, and Alon Itai. On the time-complexity of broadcast in multi-hop radio networks: an exponential gap between determinism and randomization. *J. Computer and System Sciences*, 45(1):104–126, 1992.

[4] Pradeep Bhatta, Edward Fiorelli, Francois Lekien, Naomi Ehrich Leonard, Derek Paley, Fumin Zhang, Ralf Bachmayer, Russ E. Davis, David M. Fratantoni, and Rodolphe Sepulchre. Coordination of an underwater glider fleet for adaptive ocean sampling. In *Proc. International Workshop on Underwater Robotics*, 2005.

[5] H. Bleckmann and G. Topp. Surface wave sensitivity of the lateral line organs of the topminnow aplocheilus lineatus. *Naturwissenschaften*, 68/12:624–625, 1981.

[6] Danilo Bruschi and Massimiliano Del Pinto. Lower bounds for the broadcast problem in mobile radio networks. *Distributed Computing*, 10(3):129–135, 1997.

[7] ORION Executive Steering Committee. Ocean observatories initiative science plan. Technical report, ORION Executive Steering Committee, 2005.

[8] Cree. Cree XLamp XR-E LED Data Sheet, 2006.

[9] Reinhard Diestel. *Graph Theory*. Springer-Verlag, 2005.

[10] Matthew Dunbabin, Jonathan Roberts, Kane Usher, Graeme Winstanley, and Peter Corke. A vision for saving the reef the starbug auv. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, 2006.

[11] Dynatron AG. Datasheet DMR01 Low Power Multichannel Receiver.

[12] E. Fiorelli, N.E. Leonard, P. Bhatta, D. Paley, R. Bachmayer, and D.M. Fratantoni. Multi-auv control and adaptive sampling in monterey bay. *IEEE Journal of Oceanic Engineering*, 31/4:935–948, 2006.

[13] Michael R. Frater, Michael J. Ryan, and Robin M. Dunbar. Electromagnetic communications within swarms of autonomous underwater vehicles. In *WUWNet '06: Proceedings of the 1st ACM international workshop on Underwater networks*, pages 64–70, New York, NY, USA, 2006. ACM Press.

[14] Iris Gaber and Yishay Mansour. Centralized broadcast in multihop radio networks. *Journal of Algorithms*, 46(1):1–20, 2003.

[15] Leszek Gasieniec and Andrzej Lingas. On adaptive deterministic gossiping in ad hoc radio networks. In *SODA '02: Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 689–690, Philadelphia, PA, USA, 2002. Society for Industrial and Applied Mathematics.

[16] Jimmi Gronkvist. Novel assignment strategies for spatial reuse TDMA in wireless ad hoc networks. *Wireless Networks*, 12:255–265, 2006.

[17] S. L. Hakimi and E. F. Schmeichel. Gossiping in radio networks. *Ars Combinatoria*, 35-A:155–160, 1993.

[18] Ted Herman and Sébastien Tixeuil. A Distributed TDMA Slot Assignment Algorithm for Wireless Sensor Networks. In *ALGOSENSORS*, pages 45–58, 2004.

[19] Owen Holland and Chris Melhuish. Stigmergy, self-organisation, and sorting in collective robotics. *Artificial Life*, 5/2, 1999.

[20] Shahab Kalantar. Distributed shape control of homogeneous swarms of autonomous underwater vehicles. *Autonomous Robots (intl. Journal)*, 22:37–53, 2006.

[21] Shahab Kalantar. *Field-Coupled Deformable Formations of Autonomous Submersible Robots*. PhD thesis, The Australian National University, Research School of Information Sciences and Engineering, 2006.

[22] Shahab Kalantar. Motion planning for small formations of autonomous vehicles navigating on gradient fields. In *Proceedings fo the International Symposium on Underwater Technology, Tokyo*, 2007.

[23] A. Kaya and S. Yauchi. An acoustic communication system for subsea robot. *OCEANS Proceedings*, 3:18–21, 1989.

[24] R. D. Keynes and H. Martins-Ferreira. Membrane potentials in the electroplates of the electric eel. *Journal of Physiology*, 119:315–351, 1953.

[25] Daniel B. Kilfoyle and Arthur B. Baggeroer. The state of the art in underwater acoustic telemetry. *IEEE Journal of Oceanic Engineering*, 25(1):4–27, January 2000.

[26] Lawrence E. Kinsler, Austin R. Frey, Alan B. Coppens, and James V. Sanders. *Fundamentals of Acoustics*. John Wiley & Sons, 3 edition, 1982.

[27] Navinda Kottege and Uwe R. Zimmer. Acoustical localization in schools of submersibles. In *Proceedings IEEE OCEANS'06*, 2006.

[28] Navinda Kottege and Uwe R. Zimmer. Mls-based, distributed bearing, range, and posture estimation for schools of submersibles. In *Proceedings 10th International Symposium on Experimental Robotics (ISER)*, 2006.

[29] Dariusz R. Kowalski and Andrzej Pelc. Faster Deterministic Broadcasting in Ad Hoc Radio Networks. *SIAM Journal Discrete Mathematics*, 18:332–346, 2004.

[30] Sandeep S. Kulkarni and Umamaheswaran Arumugam. Tdma service for sensor networks. In *ICDCS Workshops*, pages 604–609, 2004.

[31] Naomi Leonard, R. Bachmayer, J. Graver, E. Fiorelli, P. Bhatta, and D. Paley. Underwater gliders: Recent developments and future applications. In *IEEE International Symposium on Underwater Technology*, 2004.

[32] Kristina Lerman, Chris Jones, Aram Galstyan, and Maja J Matarić. Analysis of dynamic task allocation in multi-robot systems. *International Journal of Robotics Research*, 25/3:225–242, 2006.

[33] Lumileds. Luxeon III Emitter, Technical Datasheet DS45, 2004.

[34] MAXIM-Semiconductors. MAX3120 Low-Profile, 3V, 120 μA IrDA Infrared Transceiver, 1998.

[35] Microchip. MCP2120 infrared encoder/decoder, datasheet, 2001.

[36] H. Momma and T. Tsuchiya. Underwater communication by electric current. In *OCEANS*, 1976.

[37] Brian L. Partridge and Tony J. Pitcher. The sensory basis of fish schools: Relative roles of lateral line and vision. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, 135/4:315–325, 1979.

[38] James Preisig. Acoustic propagation considerations for underwater acoustic communications network development. In *Proc. The First ACM International Workshop on UnderWater Networks (WUWNet)*, 2006.

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

[40] Daniel L. Rudnick, Russ E. Davis, Charles C. Eriksen, Charles C. Eriksen, and Mary Jane Perry. Underwater gliders for ocean research. *Marine Technology Society Journal*, 38/1:48–59, 2004.

[41] Felix Schill, Jochen Trumpf, and Uwe R. Zimmer. Towards optimal TDMA scheduling for robotic swarm communication. In *Proceedings Towards Autonomous Robotic Systems*, 2005.

[42] Felix Schill and Uwe R. Zimmer. Distributed dynamical omnicast routing. *Complex Systems (intl. Journal)*, 16(4):299–316, 2006.

[43] Felix Schill and Uwe R. Zimmer. Effective communication in schools of submersibles. In *Proceedings IEEE OCEANS'06*, 2006.

[44] Felix Schill and Uwe R. Zimmer. Pruning local schedules for efficient swarm communication. In *Proceedings of the International Symposium on Underwater Technology, Tokyo, Japan*, 2007.

[45] Felix Schill, Uwe R. Zimmer, and Jochen Trumpf. Visible spectrum optical communication and distance sensing for underwater applications. In *Proc. ACRA 2004*, 2004.

[46] C. W. Schultz. Underwater communication using return current density. In *Proceedings of the IEEE*, 1971.

[47] Rodolphe Sepulchre, Derek Paley, and Naomi Ehrich Leonard. Stabilization of planar collective motion: All-to-all communication. *IEEE Transactions on Automatic Control*, 52/5:811–824, 2007.

[48] Gursharan S. Sidhu, Richard F. Andrews, and Alan B. Oppenheimer. *Inside AppleTalk*. Addison-Wesley, 1990.

[49] E. D. Smith. Electric shark barrier: initial trials and prospects. *Power Engineering Journal*, 5:4, 1991.

[50] J. R. Solberg, K. M. Lynch, and M. A. MacIver. Robotic electrolocation: Active underwater target localization with electric fields. In *IEEE International Conference on Robotics and Automation*, 2007.

[51] R. Somaraju and J. Trumpf. Frequency, temperature and salinity variation of the permittivity of seawater. *IEEE Transactions on Antennas and Propagation*, 54(11):3441–3448, 2006.

[52] Ram Somaraju and Felix Schill. A communication module and TDMA scheduling for a swarm of small submarines. *Tr. J. of Electrical Engineering and Computer Sciences*, 15 (Special Issue on Swarm Robotics):283–306, 2007.

[53] Bulent Tavli. Broadcast capacity of wireless networks. *Communication Letters*, 10(2):68–69, 2006.

[54] Bulent Tavli and Wendi B. Heinzelman. Energy and spatial reuse efficient network-wide real-time data broadcasting in mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 5(10):1297–1312, 2006.

[55] Maurice Tivey, Paul Fucile, and Enid Sichel. A Low Power, Low Cost, Underwater Optical Communication System. *Ridge 2000 Events*, 2(1):27–29, 2004.

[56] J. Walther. Applications of underwater fields. *OCEANS*, 3:167–170, 1971.

[57] Douglas C. Webb, Paul J. Simonetti, and Clayton P. Jones. Slocum: An underwater glider propelled by environmental energy. *IEEE Journal of Oceanic Engineering*, 26/4:447–452, 2001.

[58] Edward C. Whitman. Sosus: The "secret weapon" of undersea surveillance. *Undersea Warfare*, 7/2, 2005.

[59] Ying Xu. An o(n1.5) deterministic gossiping algorithm for radio networks. *Algorithmica*, 36:93–96, 2003.

[60] Hubert Zimmermann. OSI reference model: The ISO model of architecture for open systems interconnection. *IEEE Transactions on Communications*, 28/4:425–432, 1980.