Underwater swarm robotics: Challenges and opportunities


DOI: 10.4018/978-1-5225-1759-7.ch096

©2017, IGI Global

Reproduced with permission.

Downloaded from DRO:
http://hdl.handle.net/10536/DRO/DU:30103219
Chapter 26

Underwater Swarm Robotics: Challenges and Opportunities

Matthew Joordens  
*Deakin University, Australia*

Benjamin Champion  
*Deakin University, Australia*

**ABSTRACT**

Underwater swarm robotics today faces a series of challenges unique to its aquatic environment. This chapter explores some possible applications of underwater swarm robotics and its challenges. Those challenges include the environment itself, sensor types required, problems with communication and the difficulty in localisation. It notes the serious challenges in underwater communication is that radio communications is practically non-existent in the underwater realm. Localisation also becomes problematic due to the lack of radio waves as GPS cannot be used. It also looks at the platforms required by underwater robots and includes a possible low-cost platform. Also explored is a method of swarm robotics control known as consensus control. It shows possible solutions to the challenges and where swarm robotics may head.

**INTRODUCTION**

Swarm robotics is the study of how large number of relatively simple physically embodied agents can be designed such that a desired collective behavior emerges from the local interactions among agents and between the agents and the environment. (Şahin, 2005)

Underwater swarm robotics then, consists of a number of small underwater robots that are aware of each other’s location and that work together for a common goal.

From the above definition we see that the agents cooperating in the swarm are themselves both simple and of lower individual cost than one complex robot that is able to complete the required task on its own. They are designed either to replace a single large and complicated agent, or to achieve a goal that it is not possible for a single agent, however complicated, to perform. The advantage of using a swarm over a single agent is that if any one of the agents in a swarm becomes inoperative, the remaining swarm can still complete the task. This dictates that the agent in a swarm must be of lower cost so that its demise does not greatly affect the task or the budget. (Faigl, Krajnik, Chudoba, Preucil, & Saska, 2013)
For centuries explorers have crisscrossed this globe on which we live with great success. Today we can map almost all of the Earth’s land masses. But this only accounts for about 29% of the surface of the Earth. For the remaining 71% relatively little is known. In 1953 Mount Everest, the highest point on the surface of the earth, was reached. 1960’s saw an even greater accomplishment, that of descending into the Marianas trench. At about 11 km in depth, the Marianas Trench is the deepest point on the Earth. As the pressure in water increases by one atmosphere for every 10m, to get to this depth a submarine had to be designed and built to withstand pressure from the surrounding water of about 11,000 times the pressure at sea level. It is this pressure that limits a lot of exploration of the seabed. Vessels used to carry people down to these extraordinary depths and pressures are just as extraordinarily expensive. Because of the expensive and dangerous nature of underwater exploration, little of what the oceans hold have been explored.

Recreational Ocean divers can reach depths of only 40 m and for only a few minutes at a time. (Richardson, 2008) Commercial divers have been known to reach depths of 300 m and that is as far as an unprotected but fully life supported person can reach. To explore further down requires the use of submarines and robots.

Exploration at shallower depths can also benefit from the robot. Oil rig and pipe line inspection are two examples where robots can help.

Robots are becoming more and more prominent in undersea exploration as they are cheaper than submarines and keep human beings out of harm’s way. Most robots are actually remote-control vehicles used for visual inspection, such as the exploration of the Titanic at a depth of 3840m. (Ballard, 2008)

The problem with the remote-controlled vehicle is that it needs a surface vessel, kilometres of control cable and various personnel to control and maintain it. The remote-controlled vehicle certainly allows a person to explore environments that one could not normally go to. Exploration of
the surface landmasses took centuries, so how long will exploration of, not just the seabed, but all that the ocean holds take? Enter the autonomous robot. An underwater robot with its own artificial intelligence can take on much of the exploration role without the need for constant, direct supervision. But one needn’t stop there. With a swarm of such robots, surely this exploration can occur at a greater rate. Indeed, with a swarm of robots, one is not limited to mere visual exploration. Autonomous underwater vehicles (AUV) are platforms which are self-propelled, unmanned and un-tethered. (An et al., 2004) These vehicles will be able to be used in areas such as underwater mapping and inspection, marine habitat monitoring, shallow water and mine countermeasures, (Curtin & Catipovic, 1993) search/rescue (Yuecheng & Liang, 2004) and other scientific endeavours in underwater research. The use of underwater swarms will be able to greatly enhance work and research in the underwater environment.

**SWARM ROBOTICS APPLICATIONS**

The underwater swarm opens the number of applications available to us.

**Ship Inspection**

One such application is in the area of counterterrorism. It is possible for a terrorist group to place a limpet mine on the bottom of the vessel, such as that of an oil tanker. The easiest way to find such a device is to allow divers to inspect the bottom of the vessel once it is docked in a harbour. This is however, still a dangerous task for the divers and if the vessel has already docked in the harbour it is potentially too late to find the device. It is required to stop the vessel at sea, and inspect the vessel with divers or with a remote-controlled inspection robot. This is expensive and dangerous work and can take days as the right weather conditions are required.

With a swarm of inexpensive and small underwater robots the task becomes much simpler. Before a ship can enter a harbour it must take on a pilot. At the time a pilot is transferred to the vessel a swarm of small robots can be dropped over the side. This swarm can form a “U” shape conforming to the underside of the vessel and as the vessel passes by the bottom of the ship can be quickly inspected. The swarm of robots can then be scooped up by the pilot vessel and the data collected can be reviewed for a potential problem. If that problem is found the vessel can then be stopped to resolve the issue.

**Undersea Harvesting**

The ability to perform harvesting on the seabed is another potential application. One example of this is off the island of Hawaii. Through volcanic action significant portions of the seabed around Hawaii are strewn with manganese nodules. Manganese is a valuable commodity and it is lying there ready to be picked up. Unfortunately the manganese nodules are either to widely spread or too deep to allow divers to be used to collect the nodules economically. Remote-controlled robots are also not economical enough. The use of a swarm of small robots would be able to harvest areas of manganese nodules in an economical fashion. (Shaneyfelt, Joordens, Nagothu, Prevost, et al., 2008)

**Oil Platform Maintenance**

The underwater structure of oil platforms today are maintained by commercial divers and by remote-controlled robots. This is dangerous and tedious work for divers and difficult with a remote-controlled robot. Commercial divers are kept in a high-pressure environment equal with the pressure at that depth that they must work for days. With a swarm of robots constantly cleaning the underwater rigging it may be possible to reduce the use of commercial divers and possibly that of the remote-controlled robot.
SWARM ROBOT PLATFORM

As the agent, or rather the platform/robot it works through, should be small and cost-effective, it is therefore limited in the type of platform that can be used. If the robot is going to work in close cooperation with the other robots in the swarm it must be capable of fairly precise motion. There are currently a number of robot platforms that use biological propulsion methods. They imitate fish or stingrays in the way they move. These systems are not yet precise enough for close cooperation. What is left then is the propeller and possibly a water jet as the two main propulsion methods.

This is not to say that other propulsion methods are not usable, just that they are more limited and so this chapter will concentrate on propellers as the main propulsion method.

Motion in water is equivalent to motion in air, it is just that one medium is denser than the other. A platform then can be built much like an aircraft, but with much smaller control surfaces. It could have a main propulsion propeller to thrust it forwards with wings, or other control surfaces, to direct its motion. Like an aircraft this system requires forward momentum to maintain control of its posture. This is a disadvantage that makes station keeping impossible. Therefore, just as is seen in most current aerial swarm robotics, a helicopter or quad copter system is used.

Due to the density of water a minimum of three propellers is required; two for differential steering and one for depth.

This suggests that the platform for an underwater swarm research robot then, is small, self-contained and has three propellers for motion. The robot is however not limited to this configuration. The robot could be built with a single propeller and fins used to control direction. (Gadre, 2003; Hagen, 2003; Jalbert, 2003; Nagahashi, 2003; Tsukioka, 2003) One could add a second vertical thruster to control depth. Fins would still be needed to control the direction. (Kawasaki, 2003). Each additional thruster added gives the platform more agility and maneuverability. With at least three vertical thrusters both the pitch and roll of the platform can be controlled. Adding to this at least 2 horizontal thrusters and one can begin to control navigation in three dimensions and perform some station keeping. (Frey, 2003)

If one wishes to be more bio inspired then the thrusters can be replaced with flippers. One such robot with six flippers can achieve up to 5 degrees of freedom and can also use the flippers as legs to walk along the bottom of the seabed. (Georgidas, 2004)

To pursue research with underwater swarm robots the author built low-cost robots using little more than PVC piping, bilge pumps, and computer cooling fan blades. This kept the cost low so that more robots could be built. (Joordens & Jamshidi, 2009) Figure 2 shows the configuration of this robot with 3 vertical thrusters and 3 horizontal. It was decided that, as this platform was for use as part of a swarm, a more generalised approach was to be used and hence the idea of fins or flippers was adding a complexity that was not required. As it was considered that some station keeping was required it was decided to use the 3 vertical thrusters and 2 horizontal. (Frey, 2003)

As this platform was designed to be used for the research in the control of swarm robotics it was not necessary to design for extreme depth. Having said that an analysis of this chassis design concluded that this chassis could reach depths of 100m. (Nowak, Ayhan, Derric, Daniel, & Joordens, 2008) Bilge pump motors were used with the impellers replaced by computer cooling fan blades to create a low-cost thruster that would work a few meters underwater, see Figure 2.

This design could be improved with a thruster at right angles to the horizontal thrusters to create sideways motion. This would allow the robot to perform station keeping.

The problem with this design is that it must be self-sufficient and self-contained. It must carry all processing power, location and environment sensors and a power supply. The most important
one of these is the power supply. The longer one wishes the robot to operate independently, the larger the power supply must be and this limits the minimum size of the robot. With the advance of battery technology this is becoming less and less of a problem.

The next issue for the researcher is communications with the robot. As will be discussed later, underwater communications is problematic as radio has a very limited range underwater. Once underwater robotic swarms are advanced enough for commercialisation they may not need to communicate back to a land or floating base station until the task is complete. But the researcher desires to have information ready at hand so as to be able to quickly and easily communicate with a swarm to give it new operating instructions or fixes as required. The use of radio frequencies below about 100 MHz may be used in freshwater or chlorinated pools. Unfortunately the range of these frequencies in saltwater will not extend much beyond 500 mm as the author determined experimentally. It is for this reason that a tethered robots may be an option.

The tethered robot has the advantage of being able to use a floating land-based power supply to provide all power requirements. It has no problems with communications and processing power needn’t be housed inside the robot itself. This allows the researcher to have full control of and full communications with each robot. The researcher then, is able to leave the robots underwater and modify their behaviour at any time. One excellent, low-cost commercial platform is the VideoRay™ inspection robot. The concern with this type of robot is the tether itself. Whilst the tether allows the robot to be able to access all the power it needs the tether is a physical limit on its range. There is also the problem of tethers tangling with many tethered robots in the water. For this reason a swarm of underwater robots can only have the few tethered robots as part of the swarm.

This can be a positive. The tethered robot, having a good power supply, can be the workhorse of a swarm, whilst the untethered robots can operate as inspection or scout robots.

Having decided on robot platform, research in underwater swarm robotics will normally take one of two approaches. Either the construction and application of a real robotics swarm or simulation. The physical robotics swarm has the advantage of applying swarm theory to a real application. Its main drawback is the cost of all the units and downtime due to building, maintaining and altering the robot platforms for different tasks.
The simulation allows as many robots to be simulated with very little cost. There is no maintenance and any alterations are done in software. The disadvantage is in the simulation itself. Simulations can be tweaked so that they work. This cannot be done in the real-world and therefore a simulated swarm may not represent a real system. Despite this, most research into underwater swarm robotics uses a simulator of some sort. (Prevost, Joordens, & Jamshidi, 2008)

There is a possible solution to this dilemma. The combination of both simulation and real applications.

Use of tethered robots, being controlled from a central computer, allows the real-world and the simulation to be combined. Once the software to control a tethered robot has been designed it can then be configured so that a software package can create as many instances of that robot control software as there are robots. Once this is possible it is then also possible for this software controller to control a virtual, or simulated, robot. The author used this approach to use one or two tethered VideoRay™ robots plus a tethered version of the previously described swarm robot platform and many more simulated robots to operate as one swarm. (Joordens & Jamshidi, 2010) See Figure 3

The simulator of course needs to know how the robots perform in the real world to be able to simulate them. There are many fine physics engines available that will allow one to describe the behaviour of the simulated robots in great detail. Fortunately, there is a simpler method. By filming the real robots moving next to a scale and then analysing the video footage one can determine the robots’ maximum accelerations and velocities, both linear and angular. With these constants determined it is relatively simple to build a simulator around them. Figure 4 shows one combined simulator/controller. It has two sets of controls for the two real robots (robots labelled 1 and 2), and allows the remaining robots to be simulated.

The controlling software for each robot is not aware that it is either real or simulated and does not know which of the other robots are real or simulated. It treats every other robot in exactly the same manner. The simulator’s operation can actually be used as verification of the simulator. If

Figure 3. Author’s robot with the VideoRay™ inspection robot
the simulated robots function equivalently to the real robots then it can be seen that the simulation is running accurately.

A simulator built on this principle allows the benefits of the simulator’s low cost and the benefit of real robots being used in the real-world allows the best of both worlds.

SENSORS

The variety of sensors that can be used on an underwater robot is large, but not as large as sensors used on robots above water. The medium of water means that most radio and high-frequency based sensors are not usable or can only be used with great difficulty. It is therefore useful to understand which sensors can be used and how.

The first and simplest sensor that any underwater robot will use is a pressure sensor. This allows the pressure of the water and hence the depth of the robot to be determined. The next simplest sensor is the compass. The beauty of these sensors is they are consistently accurate over time. No matter how long the robots have been in the water and accessing data from these sensors, the readings will be just as accurate as when the robot was first turned on.

Both the sensors are valuable tools in determining the location of the robot, but unfortunately they are not enough. With today’s technology most robots will use some sort of GPS to determine their location. However, since GPS is dependent on radio it is completely useless underwater. That is not to say an underwater robot should not have a GPS on board. GPS is very useful if the robot is able to come...
to the surface and use the GPS to fix its location. Therefore, in order to help the robot to determine its location it requires the use of an Inertial Motion Unit (IMU). By using an IMU the robot can determine its direction, its accelerations in three dimensions and its rotational accelerations in three dimensions. With this it is then able, with the addition of a clock, to determine its position using dead reckoning. The issue with using an IMU is that its measurements are subject to drift over time. Therefore the longer it is used the less accurate it becomes.

The next important sensor for the underwater robot is sonar. Whether the sonar unit be single beam and single echo, or a sonar multiarray, it is able to give a measurement between objects around the robot and the robot itself. This system does not deteriorates over time and can be used to detect features in the environment to help the localisation of the robot. If it is used to map an environment however, then its accuracy is only as good as the sensors that are used to localise the robot. In the same way a camera system can be used to detect the robots environment. The use of the camera system however often means that more processing power must be used. It is also dependent on the environment. In tropical waters a camera may be able to see objects up to 40m away, but in colder waters and in harbours it can be limited to 500mm or even no visibility at all. Thus the use of a camera must be carefully considered with the environment in mind.

The sensors mentioned form the basis of the sensors required for an underwater robot. In the use of these sensors one can determine that some sensors lose accuracy over time and others not. It is important to consider this during data collection.

As the sensors provides some overlap of data, a very simple data fusion algorithm can be used that takes into account the degradation in time. One can consider the sensors to be of one of two types; absolute and relative to time. The confidence of any data from a sensor will depend on its type and on time of use and a weighting factor can be used on that data.

For absolute sensors the weighted data can be given as;

\[
wa = da \times ka
\]

(1)

where \(wa\) is the weighted data; \(da\) is the sensor’s data; and \(ka\) is the level of confidence (0 – 1).

For relative sensors the weighted data can be given as;

\[
wr = dr \times kr \div t
\]

(2)

where \(wr\) is the weighted data; \(dr\) is the sensor’s data; \(kr\) is the level of confidence (0 – 1); and \(t\) is the time since data collection started.

To fuse the data then;

\[
w = \frac{\sum_{i=1}^{na} wa_i + \sum_{i=1}^{nr} wr_i}{Ka + Kr}
\]

(3)

\[Ka = \sum_{i=1}^{na} ka_i\]

(4)

\[Kr = \sum_{i=1}^{nr} (kr_i \div t_i)\]

(5)

where \(w\) is the weighted data; \(na\) is the number of absolute sensors; \(nr\) is the number of relative sensors; \(Ka\) is the sum of the absolute confidence; and \(Kr\) is the sum of the relative confidence/time.

The confidence in the relative sensors is divided by time as the more time the sensor has been collecting data the less accurate it is and the less confidence one has in it. (Shaneyfelt, Joordens, Nagothu, Prevost, et al., 2008)

**PROBLEMS WITH UNDERWATER COMMUNICATION**

One of the largest problems that exists while designing and developing underwater robotics is how the robots are able to communicate back
to the surface and/or with each other in the case of a swarm of robots. In land, air or even in outer space a lot of different solutions have been developed to overcome this problem, the most prominent one being the use of radio waves to communicate between the different parts of the system that are not directly attached to one another. Most wireless systems use this form of wireless communication, whether it be through a simple remote controller such as is used to control RC planes, cars, boats or by using more advanced systems such as Xbee™ modules which allow for more complicated messages to be sent between the subsystems. Most households and businesses uses this form of communication by utilising 2.4GHz or 5GHz Wi-Fi routers to connect to the internet and to network their various devices together. These communication systems essentially operate by having an omnidirectional antenna attached to the transmitter and the receiver, which allows for the transmission and retrieval of information from all directions. Then by utilising a high communication frequency, a large amount of data is able to be transferred between the two of the units on the network. Unfortunately in the underwater domain, this common method of communication is currently not a feasible solution.

**Issues with Traditional Wireless Communication**

The main issue preventing this type of communication is that water has a very good ability to absorb high frequency radio waves, much better than that of earth’s atmosphere. This property of water is used by the use of the common household microwave oven. A microwave oven operates by bombarding an object with microwaves, a high frequency (2.45GHz) form of radio waves. These microwaves are absorbed by the water molecules inside the object causing them to heat up, thus heating the entire product up as this excess heat energy is transferred to the rest of the object from the water molecules inside of the object. Unfortunately for robotics, this also means that any radio waves transmitted by the robotic system are also absorbed by the surrounding water molecules, causing the range of high frequency radio waves to be very short as they are quickly absorbed by these molecules. It has been shown that when a standard 2.4GHz transceiver is used underwater, a range of about 0.15m can be acquired(Joordens & Jamshidi, 2010). This range is clearly not large enough for practical communication as the robots would have to be very close to one another to be able to communicate effectively, thus greatly reducing the tasks that the robots are able to easily achieve, as well as causing the robots to be potentially damaged when they communicate with one another due to the requirement of the robots being within close proximity of each other. It can be noted that as the communication frequency decreases, the range of communication increases. This is can be shown, as when a frequency of 100MHz is used, a range of 2m – 3m can be acquired (Joordens, 2008; Schmickl et al., 2011; Shaneyfelt, Joordens, Nagothu, & Jamshidi, 2008). If a lower frequency of 75MHz is used it is possible to get a range of 2.7m(Shaneyfelt, Joordens, Nagothu, & Jamshidi, 2008) between the different vessels. It also should be noted that the type of water the transceiver is in will also make a difference, as when the same 75MHz transceiver is also used in a saltwater pool, the range of communication greatly reduced to only 0.3m(Shaneyfelt, Joordens, Nagothu, & Jamshidi, 2008), compared to the 2m – 3m that was acquired in a chlorinated pool. It is possible to get longer ranges than those presented here by increasing the power that the transmitter uses to relay the information, but from the vast difference in ranges that are obtained compared to what is achieved when using the same transmitters in air, this is not a practical solution. Following this trend, it can be seen that the lower the frequency the better the range underwater. Because of this, low frequencies are generally used to communicate wirelessly underwater, with the frequency of the wave generally being in the acoustic range.
Communicating with Light

There has been other forms of wireless communication that have been attempted between underwater vessels, as opposed to the standard RF communication methods that have been mentioned above. One of these types of systems has been developed by the CoCoRo group, and involves using flashes of light at certain frequencies or intensities to transmit data from one robot to another robot (Schmickl et al., 2011; Sutantyo, Levi, Moslinger, & Read, 2013). Experiments have shown that using this type of communication a bit rate of up to 119kbs (Schmickl et al., 2011) can be achieved, meaning that it could be used for many robotic applications as a significant amount of information can be transmitted in a small period of time between one or more robots. This method is not without its drawbacks though, as it is shown that the range of communication is also very dependent on the type of water that the system is in, with variables such how well the water reflects the signal can make a big difference on the range of communication that can be achieved. Communication ranges of 1.2m can be acquired in one type of water, where ranges of as little as 0.4m is achieved when the water quality is reduced (Schmickl et al., 2011). If the water becomes murky, this type of communication could be unable to be used at all, as the light from the emitter will not be able to easily penetrate through the water (Shaneyfelt, Joordens, Nagothu, & Jamshidi, 2008), and thus not be received by the receiver. The range of using this communication system can also be increased by increasing the brightness of the emitting device, but this is not linearly scaled, as the above mentioned ranges are from a 40mW LED, where a range of only 2.1m has been achieved if a 400mW LED is used. The CoCoRo system do not rely on this type of communication alone, as the light based communication is only used for high bit rate transfer of information in a local sense, with the an acoustic communication system still being utilised when global data or long range data needs to be communicated between two or more robotic systems. This shows that the solution to the communication issues could involve having two different communication methods, one for global and one for local communications.

Communicating with Sonar

Acoustic or sonar communication works on the same theory as a normal wireless transceiver, this being that there is a nonphysical link between two antennas on the vessels, but at a much lower frequency than that used by similar wireless systems on the land. It is well known that this type of communication works well under water as it is used by many underwater mammals, such as whales and dolphins. A relatively large range can be obtained with this form of communication but it is not without its drawbacks. In robotics the largest drawback within these frequencies is that as the frequency decreases, so does the bitrate that the systems are able to communicate at. In the case of acoustic communication, the bit rate can get down to as low as 30 bytes per second (Joordens & Jamshidi, 2010), which is too slow for many robotic applications as it limits the type of information that can be sent, such as video, as well as lowering the updating rate of the information between the different subsystems as it takes so long to relay the information. Other issues are also present with this form of communication such as the speed in which the wave travels, being very slow compared to its high frequency counterparts (Shaneyfelt, Joordens, Nagothu, & Jamshidi, 2008), and these frequencies of waves are also prone to reflections from hard surfaces such as off pool walls and floors, which can introduce a considerable amount of noise into the communication system (Joordens & Jamshidi, 2010). This makes it much harder to test the platforms using this type of communication as the walls of a pool will cause this interference introducing unacceptable noise into the system.

This interference is not always caused by the initial wave, but could also be caused by interfer-
ence from other transmitters in the environment, effectively causing the different signals to get merged together so it is perceived as noise, meaning that the information is not properly received by some of the systems. This is created when one or more transmitters tries to communicate at the same time, thus causing the interference as they are transmitting information on the same frequency. Most people would have observed a similar situation by trying to talk at the same time as someone else on a hand held radio. The third person in the network is unable to understand what either person is saying, as both messages reach them at the same time resulting in a combined signal. This is a big issue when multiple sources require updated information, such as when a swarm of robots is considered, or an underwater sensor network. In the underwater setting missing packets of information can be a bigger issue due to the slow communication rates that can be achieved. When the land or air based scenario is considered, missing a single message can be seen as only a small issue as new information will be received in a relatively short amount of time, updating the old information in the system, so small amounts of packet loss is acceptable. There are multiple techniques that can be used to try and overcome this issue, a simple one of these techniques being token passing.

Token passing is essentially where one of the systems holds onto a token that enables it to communicate for the time that it has the token. This is not a physical object, rather a virtual one. When a system does not have the token, it is unable to transmit information and is only able to receive information. The token is passed from one system to another when a predetermined time elapses. Using this technique means that a system will never be communicating at the same time as another system, and thus the interference is eliminated. An advantage with using this type of system, as opposed to one where the systems are able to pass the token on at the end of their communication is that all of the systems do not have to be within communication range of each other before they can communicate, such as in many underwater communication cases. If one system does not receive information form the other system it doesn’t matter as it still has to wait for its turn to communicate, ensuring that a device that is in range of both systems will not receive two different messages at once.

Types of Communication

There are many different methods that have been developed to try and overcome the communication problem in underwater robotics. These methods can be split into two different forms of communication, being direct communication and indirect communication (Dorigo, 2013; Jevtic, Gutierrez, Andina, & Jamshidi, 2012). Direct communication is the most common form of communication and involves two objects, in this case robots, communicating dynamically between one another. This is similar as to how two humans are able to communicate by talking to one another. This form of communication is often accomplished by using radio waves, as described earlier, but can be achieved by utilising other methods such as a tether connecting the two systems together. Due to the issues that are present using this form of communication underwater, other methods of communication have been developed generally falling under the category of indirect communication. Indirect communication is where the environment is manipulated by one of the parties to achieve communication between the different systems. Some examples of this type of communication is robot grouping, intensity of light modifications, environment manipulation, etc. By using this method it is possible for a large amount of information to be communicated at a single point in time. There are some major drawbacks present when this type of communication is used though, such as the time between messages...
Underwater Swarm Robotics can be very large and infrequent, it is quite possible for the robot to easily miss messages and it is much more complicated for the message to be passed between the different robots in the system.

### Tethering

There are many methods that have been trailed and tested to overcome the issues surrounding direct RF communications underwater, such as robots surfacing to communicate with each other (Ali & Hassanein, 2008; Porfiri, Roberson, & Stilwell, 2006). Each of these different types of methods have their own pros and cons, but that there has not yet been a standard set on how communication should be achieved. The most common solution to avoid the communication problem that is associated with underwater robots is to use a tether tied to a base station that is above the water line to communicate between the robots (Joordens & Jamshidi, 2009, 2010). Many robots can be attached to a single station, or the stations can communicate between each other above the water line using traditional methods. The advantage of using this type of system is that the communication speeds are only limited to what can be transferred down the tether, which should be large enough for any robotic application that can be achieved by any other form of communication in any other medium that is tried and tested today. Resorting to this method is not without some major drawbacks through, making some types of robotics much harder to accomplish.

The major drawback that this type of system has is the range the robots are able to travel away from the base station, being limited by the length of the tether. This means that to achieve a significant range on the robot, either the base station has to be able to move with the robot, or a very long tether has to be attached to the robot. Either of these methods have large issues, such as coordinating the movement of the base station with the movement of the robot, as well issues surrounding the tether such as having to retract it without causing damage to the tether or the robot. This leads into the next issue that is associated with this type of tethered communication, tangling.

Since using a tether to communicate requires a physical link between the robot and the base station, it is possible for this tether to get wrapped around objects in the environment, potentially meaning the robot is unable to be recovered without assistance or potentially harming either the robot or the tether. It is also possible for the tether to knot, potentially casing damage as the tether is wound or unwound. Damage to either the tether or the robot can be time consuming, or possibly unrealistic in certain situations, as issues such as the cost to repair or retrieve the robot becomes larger. If this type of communication is used with multiple other robots in a swarm, particularly if they are connected to the same base station, the different tethers can potentially get tangled around one another thus interfering with the operation of the robots.

### Future Work

Recently there has been some advances in underwater communication technology, but this technology has yet to hit the main stream market, with a lot being yet untested on moving underwater platforms such as an underwater robot. There are many different methods that have been attempted to solve this problem, some being more efficient than others. An example of one of these is a method that removes itself away from the main stream way of thinking, by not simply relying on the analogue methods that most current underwater communication systems are based off. Instead a system has been developed that is more like a broadband system and is combined with specially created antennas that are tuned so as they are optimised for underwater use. This method involves using different frequencies to send the information, leading to more of a parallel like transmission of information. Using this method data rates of up to 1 Mbps should be able to be achieved, over
ranges of 100m to 1000m (Kelley & Naishadham, 2013). By researching new approaches the into the communication issues present in complicated areas, such as underwater, could mean that the communication issue present in the underwater domain could become less prevalent in the not too distant future. By increasing this type of research in communication poor areas such as underwater, the new techniques developed could be applied to the “standard” environment, increasing the communication speeds even more allowing more and more complicated tasks to be performed.

**UNDERWATER LOCALISATION**

As a need for robotics in the underwater domain becomes more and more prominent, a need for the robot to be able to localise itself in the environment has become more and more of an issue. This becomes especially evident in swarm robotics as many swarm algorithms require the robots to know where they are in relation to their counterparts, such as in formation control. Many other robotic applications also require the robot to know its exact location in the environment, such as if the robot is trying to map its surroundings.

**Dead Reckoning**

The most common form of localisation in underwater robotics is to use dead reckoning. This method was used by marine craft for many years before the advancement of GPS, making dead reckoning relatively redundant for basic navigation in the land or air based setting. Even so, submersibles such as submarines still rely on the dead reckoning method of tracking as GPS signals are of a frequency that is too high to significantly penetrate the surface of the water and reach the underwater vessel below, making the GPS signal virtually useless. Dead reckoning essentially operates by taking the last known position of the vessel, and adding how far the vehicle has travelled since the last reading to the old position. This will give the new position of the vehicle relative to where it was when the process was started. This method is then repeated continuously until a target has been reached, whether this be a known location, a period of time, etc. This method is not without issues though. To be able to calculate how far a vessel, or in this case a robot, has travelled sensors are needed that are able to measure the distance that the robot has travelled, as well as the orientation that the robot has travelled in. Some of the sensors that can be used are accelerometers, magnetometers, gyroscopes, etc. which are prone to drift over time. This means that a compacting error from the sensor measurements will occur, causing values calculated from the sensor readings to change without relation to the vehicle’s movement over time. A simple experiment that can be performed to observe this is to set up an IMU with some basic tracking software on it and put it in a static place, this being a place where the device will not be able to move, such as on the top of a sturdy table. It should be noticed that after a period of time the device will think that it has translated a certain distance, changed its orientation or both, where in fact it has not done so. There are many ways to help to try and reduce this noise, such as by implementing a kalman filter onto the sensors signal or the calculated information(Kleeman, 1992), but noise will most likely still be a significant factor within the system, over a long period of time. There are many other issues that can be present with this form of tracking such as the vessel drifting or being pushed around by currents which may not be detected by some sensors such as encoders, as well as other sources of noise that can be introduced into the system, such as slip if an encoded motor is used to determine the distance that the robot has travelled. To overcome these errors a GPS or similar reading should be taken at consistent intervals to recalculate the actual position of the vehicle. As mentioned earlier taking these extra measurements can prove to be a challenge.
Underwater Swarm Robotics

Figure 5. Basic dead reckoning example

for underwater vehicles, and might require the entire vessel to surface, or for the vessel to send up a float with a GPS receiver on it to be able to receive the GPS signal.

Simultaneous Localisation and Mapping (SLAM)

Another form of localisation that is being implemented in recent years in Simultaneous Localisation And Mapping (SLAM). There are two types of SLAM methods that are often used, one being laser or ultrasonic SLAM (Misono, Goto, Tarutoko, Kobayashi, & Watanabe, 2007; Williams, Newman, Dissanayake, & Durrant-Whyte, 2006), and the other being visual SLAM (Aulinas et al., 2011; Salvi, Petillot, & Batlle, 2008).

SLAM essentially operates by initially placing the robot in an unknown location. When the robot starts, it begins detecting the different obstacles that surround the robot in the environment using either a vision or a distance sensor. A local map can then be constructed based off where these obstacles are in relation to where the robot thinks it is in the map (Williams et al., 2006). Initially the robots position will be at the origin of the map, unless some other global positioning method is also incorporated into the process, such as a GPS. The map generating process is completed every cycle of the program. Each time a local map is generated, it has to be compared to the previously generated maps to determine what objects from this map can be used as features, these being points on the map that can be referenced by the algorithm, and what objects are either noise or moving obstacles and therefore should be removed from the map. This is generally completed using one of two methods, either a kalman filter (Salvi et al., 2008; Williams et al., 2006) or a particle filter (Bruno & Oussama El, 2009). Once the program has been able to determine which obstacles are features, the robot is then able to place itself in the global map. This is accomplished by determining which features are the same ones as the previously detected features. Once this has been determined, the current range and angle data received from the sensors in relation to the features can be used by the robot to place itself on the map.

Visual SLAM operates in essentially the same manner, as it still detects different features in the world and then tries to place the robot on a map relative to these features. Visual SLAM can generally be accomplished with much cheaper hardware (Salvi et al., 2008), as a simple webcam can be used to achieve SLAM. The biggest difference is that visual processing also needs to be completed to find the features, which can mean that this version of SLAM becomes much more
computationally expensive. Visual SLAM is also more susceptible to noise, as it is common for two objects in the environment to be visually similar and therefore making it harder to distinguish between the different features in the environment.

There are some major drawbacks that are associated with SLAM, and can occur regularly in the underwater situation. SLAM works off the assumption that the environment that the robot will be in is feature rich, this meaning that there are a lot of obstacles or points of interest that the robot can be constantly detecting and therefore is able to place itself on the map. It also assumes that the environment is relatively static, meaning that there are not many objects that are moving. If objects move, stop and then move again, such as what underwater animals are prone to do, these objects can be detected as features when they are stationary, meaning that the robot will try and place itself in an entirely different part of the map when object starts to move again, as the feature that the robot was using to place itself no longer exists in the same location. If the robot enters a situation where there are not many or any of these features present, it will become lost and not able to insert itself onto the map, which could lead to all kinds of issues such as the robot thinking that it is not moving when it actually is, causing the robot to travel large distances off course. There are also some major environmental challenges that have to be overcome, such as the strong attenuation and scattering of light that occurs underwater which majorly affects vision based SLAM (Salvi et al., 2008), as well as the potential for bubbles and other floating debris to be picked up by the range finding sensor.

**Commercial Systems**

To overcome the GPS denied environment that exists underwater, some commercial tracking systems have been developed. Most of these systems relay on time difference of arrivals (Larsen, 2000) triangulation algorithms. This essentially operates by having a system sitting on the surface of the water that send a message down through the water, typically transmitted using ultrasonic waves. The signal is then picked up by the intended device which is able to send a time stamped message back to the base station sitting on top of the water. This information is then used by the base station to calculate where the robot is relative to the system’s own position. There are some major issues that surrounding this type of positioning system, such as that generally these systems are designed to only communicate with one robot at a time as well as the other issues surrounding ultrasonic waves mentioned in the communication section. The ability to only communicate to one robot at a time is a major downfall if multiple robots are being utilised, such as if a swarm of robots are used, as there can be a significant amount of time where the robot is unable to locate itself in the environment, during which time the robot could significantly drift off course. Using Ultrasonic waves also means that the frequency in which wireless communication could be used between the underwater robots is limited, as interference will could be generated if the same or similar frequency

*Figure 6. Example of a laser SLAM map (Beeson, 2015)*
Underwater Swarm Robotics

is used as the positioning device. This issue is not present in a GPS system as the frequency of a GPS is significantly different to the frequencies that are normally used by robots to communicate between one another. Ultrasonic waves underwater also have a tendency to bounce off hard surfaces, such as the walls and the floor of a pool, which can make this type of system hard to implement in controlled test conditions.

Combining Systems

It has been shown that there are major drawbacks with any of the individual systems that are used to track robots, particularly in the underwater domain, and as such it can be a good idea for a robot to use a combination of systems when tracking its own position underwater. It is commonplace in all forms of robotics to combine dead reckoning with other types of tracking systems to overcome the drift issues that is present when dead reckoning is used, while ensuring that the robot has a good idea where it is at all times. A common method is to combine dead reckoning with SLAM to help overcome the robot becoming lost in feature poor environments as well as the robot suddenly translating itself to some other part of the map if one of the features start to move, such as if the system mistook an animal for a feature. This is accomplished by checking the odometry received from SLAM against the odometry received by dead reckoning algorithm and looking for large discrepancies. Dead reckoning can also be combined with commercial systems to overcome the time delays that exist between readings by using the position information obtained from the commercial unit to reset the robots position. This is commonly done with GPS systems to overcome the time delay between readings as well as to try and eliminate the discrete jumps that a GPS system is prone to. It could also be used when a commercial system is used to track the robots position underwater.

CONSENSUS CONTROL

The final hurdle for the designer of an underwater swarm robotics system is the control method. The two main types of control methods are either a centralised agent acting as the master, or a decentralised method.

Centralised control is often the easiest method of control allowing one system to have a big picture view and to be able to control all the other robots in this big picture. Centralised control systems must also be able to nominate another robot as the master controller if the current master control robot ceases to function. The master control robot must also have good and quick communications with all the other members of the robotic swarm. This poses a concern in the underwater environment where, as has been seen, communications can be a large problem.

In a decentralised control system, each robot in the swarm must make its own decision on how it can further the goal of the swarm. In decentralised control there are no large repercussions if an individual robot ceases to function. There is no need to pass the token of control to another robot. As each robot is able to use its own sensor system and its relationship to the other robots to make a decision that communications burden is reduced. Indeed if a robot is able to locate its neighbouring robots in a swarm then there may be no need for communications at all. It is for these reasons that a decentralised control system is best for the underwater robotic swarm.

Consensus control is such a decentralised control system. It tries to operate in the same manner as, for example, a football team. The aim of the football team is to get a ball to one end of a football field to score. Each player on the team is able to locate the ball, fellow team members and opposing team members in relation to the playing field. Players will often shout to each other to give each other updated locations on where they are and to give the opportunity for the ball
to be passed to them. Each player then, has all the information needed to make a decision based on aims of the whole team.

Consensus control endeavours to operate in a similar fashion. (Joordens & Jamshidi, 2010) It has an aim, or goal, for the swarm to pursue. Each member of the swarm knows what that goal means for itself. The only other information about the swarm that each member of the swarm needs to know is the location of each other swarm member, and possibly their current state. Ideally each member of the swarm knows exactly where each other member is at all times. (Olfati-Saber, Fax, & Murray, 2007; Wei, Beard, & Atkins, 2005) Consensus control tries to get as close to this ideal state as is possible (Namerikawa & Yoshioka, 2008; Wei & Beard, 2008).

Consensus control shares the information required between all members of the swarm which then allows each swarm member to decide its own course of action in the best interests of the swarm.

The following case study seeks to examine how swarm control using consensus control performs. This case involves the use of robots in a controlled swarm to patrol an area of water. (Joordens & Jamshidi, 2010) The patrol area was a square of 2 x 2 metres which was chosen as the experiment was to be performed in a pool which did not allow much more room. Two real robots were used and the remaining robots were simulated such that they could interact with the real robots as explained in an earlier section. To patrol this area each robot was to move along the perimeter of the patrol square such that all the robots were evenly spaced. To ensure that collisions did not occur, each robot, simulated or real, was given a different depth to operate at. This allowed the system to see where collisions would occur, if all the robots were at the same depth, without these collisions actually occurring. The robots must be able to move to the patrol area and begin patrolling, inserting themselves in between the robots that are already patrolling. To this end each robot needs to know where the patrol area is and where each of the other robots are. As these robots did not have sufficient sensors to determine where the other robots are, communications were required to share this information.

The real robots positions were determined with the use of a Tritech MicroNav™ navigational system. This system uses a central sonar buoy and transponders on each real robot to return a coordinate for that robot. This was able to update the position of the real robots in half second increments. This system was simulated with the simulated robots. It was assumed for this experiment that each robot was able to locate its own position and then share that information with the other robots. This meant that the communication system had to allow each robot to transmit its location and its identification. This reduced the burden on communications, requiring only 10 bytes of information to be transmitted by each robot.

Consensus control requires four steps to set up its operation (Wei & Beard, 2008).

**Step 1:** Determine the cooperation objective.

As the robots know the patrol path they must follow, the main control for this case is the distance between the robots which must be equal. Hence:

\[
J = \frac{p}{n} - \left( \frac{1}{n} \sum_{i=2}^{n} VR_i - VR_{i-1} \right) / n
\]  
(6)

where \( VR_i - VR_{i-1} \) is the distance between two Robots, one following the other; \( p \) is the total distance of the path; \( n \) is the number of Robots; and \( J \) is the cooperation constraint.

For the goal of the swarm to be successful, the cooperation constraint, \( J \) must approach zero. For this to happen the distance between each robot must be equal to the total distance of the patrol path divided by the number of robots in the swarm. It will be noted that this particular formulation of the control conditions allows a variable number of robots in the swarm. This allows the swarm to
complete its objective when the number of robots in the swarm vary with both robots that cease to function and robots are added to the swarm.

**Step 2: Informational requirements**

The information that needs to be shared between all the robots has already been defined as the position of each robot and the robots identification.

**Step 3: Centralised strategy**

As consensus control is a decentralised process all members must know what the overall control strategy to base decisions upon it. In this case the strategy is that all robots in the patrol path must be equidistant from each other.

**Step 4: Consensus building**

To maintain the centralised strategy, this must be broken down for each robot. In this case each robot must remain a certain distance from the robot in front. The robot knows the total length of the patrol area and, through the communications between all robots, knows the total number of robots present. It can divide the path length by the number of robots to determine the distance it must remain behind the robot in front.

Figure 7 shows the robots positions in the patrol path using the consensus control developed. It shows that while they are following the correct path, they are still bunching up and colliding with each other. This is not due to the consensus control, but to the communication system.

As seen previously communications are not easy in the underworld environment. This case study took this into account and caused all communications to run at 300 board, assuming the use of low cost underwater modem system. This created a huge time delay in the communication system.

A robot would transmit its three-dimensional coordinates and its identification number. The robot with the identification number one above the transmitted identification number would be allowed to transmit its data. If that robot failed to transmit then, after a time delay, the next robot up would transmit. After the robot with the highest identification number had transmitted its data, and a significant delay occurred, the robot with the lowest identification number would transmit. This system allowed each robot to determine the full number of robots operating in the swarm. It also allowed for missing robots to be skipped and for the system to be restarted the lowest numbered robot. This communication system did have a number of built-in delays but it did not have the requirement to send acknowledgement signals.

For the number of robots used in this case study there was a two second delay between a robot transmitting its position and transmitting its latest position. In two seconds these robots could travel up to 600 mm which was about twice the robots length.

This meant that there was an error in the positional information of a robot of up to 600 mm and it is this that caused the bunching up and collisions seen in Figure 7. To resolve this issue a prediction system is required.

*Figure 7. Robots patrolling area with Consensus control alone*
Each robot timestamps each transmission it receives and keeps in memory the last two positions for each of its fellow robots. A robot can then estimate the position of its fellow robots using a linear extrapolation as follows:

$$\begin{aligned} P_n' &= P_n + \left( P_n - P_{1n} \right) \times \left( T - t_n \right) / \left( t_n - t_{1n} \right) \\
\end{aligned}$$

where $N$ is the robots ID; $P_n'$ is the estimated X, Y, Z coordinates of robot $n$; $P$ is the last known position of robot $n$; $P_{1}$ is the next to last known position of robot $n$; $t$ is the time of $P$; $t_{1}$ is the time of $P_{1}$; and $T$ is the current time.

Figure 8 demonstrates that with this simple prediction system, the consensus control works very well. There are now no collisions but there is still some bunching up and spreading. This is due to the prediction system using a purely linear approach and not taking into account the shape of the patrol path which would mean that when robots were turning a corner of the patrol path the predicted position would be better than no prediction but still wrong. The control, and hence the spacing of these robots, could be improved with an improved prediction system.

In this case however, this option was not pursued as the linear system did a reasonable job which was sufficient for this real-world application. It also meant that the same system could be used for different patrol paths.

**FUTURE RESEARCH DIRECTION**

There is an enormous potential in underwater swarm robotics. The problems however are just as enormous. The biggest area of concern is in underwater communications. For underwater swarm robotics to come into its own, underwater communications needs to be greatly improved. This is the most significant area required for further research.

The next area of research required is that of the power supply. Current batteries are too large for the requirements of swarm robotics. Swarm robotics needs the use of small robots and this dictates smaller batteries. Due to this same requirement required in many other technologies today, most notably in tablets and smart phones, there is good progress in this direction.

As the above two problems are solved in the future, hopefully the near future, the next area of development will be the control system. As the swarm robots get smaller and cheaper the possible applications will expand, requiring better control systems.

**CONCLUSION**

In all areas of robotics, swarm robotics seems to be the way of the future. This is no different with the underwater environment. Underwater swarm robotics will be able to learn from its aerial and terrestrial cousins as these are currently far more advanced. To get to that position however, many problems unique to the underwater environment
will have to be conquered. Better communication systems will have to be devised, improved and new sensors designed for the underwater environment will have to be created and improved location detection systems will be needed.

Even with these problems however, new underwater robots are being developed and better swarms are being created. Swarm technology is also progressing in the area of interactions between swarms and soon we should see underwater swarms being able to cooperate with land-based robotic swarms.

Researchers into underwater swarm robotics have great challenges before them but also great opportunities for advances in this exciting realm.

REFERENCES


Underwater Swarm Robotics


**ADDITIONAL READING**


KEY TERMS AND DEFINITIONS

Consensus Control: The decentralized system used to control a swarm of robots.
GPS: Global Positioning System.
IMU: Inertial Measurement Unit.
Kalman Filter: An algorithm used to reduce the amount of noise in a stream of data.

ROV: Remote Operated Vehicle.
SLAM: Simultaneous Localisation and Mapping.
Swarm Robotics: The study of groups of robots that cooperate to achieve a shared goal.
Tritech MicroNav™: Trademark of Tritech International Limited.
VideoRay™: Trademark of VideoRay LLC.