DOI: http://dx.doi.org/10.7551/978-0-262-32621-6-ch047

The Relay Chain: A Scalable Dynamic Communication link between an Exploratory Underwater Shoal and a Surface Vehicle

Becky Naylor[1], Mark Read[2], Jon Timmis[1] and Andy Tyrrell[1]

[1] Department of Electronics, University of York
[2] Charles Perkins Centre, The University of Sydney

Abstract

In this paper we present the Relay Chain: a new algorithm central to a novel strategy for exploring underwater environments using a swarm of autonomous underwater vehicles (AUVs). The Relay Chain provides a mobile, scalable and dynamic communication link between the water's surface and a shoal of AUVs exploring the sea bed. The chain tracks the shoal as it explores, recruiting and returning AUVs from and to the shoal to modulate the length of the chain as required. The chain can instruct the shoal to reverse course when it travels too far from the starting point and there are no further AUVs to recruit. Given the challenging underwater environment, chain breakages are however inevitable, and as such we consider a number of recovery strategies to address chain breakages and minimise the time for which messages destined to the surface from the shoal are delayed. A simple 'turn and search' strategy is contrasted with strategies that depend on absolute positioning systems of varying accuracy. Implementing underwater absolute positioning systems is challenging, and our results highlight how accurate such a system must be to outperform more naive strategies, and hence be considered a worthwhile investment. We find the accuracy must be within 20cm, being 40% of AUV sensor range.

Introduction

Underwater exploration is a challenging task, the environment is noisy and difficult to access and navigate, and communication signals are short range due to heavy attenuation. The challenge can be met through a swarm of small autonomous underwater vehicles (AUVs) which divide aspects of the task between them and as a collective provide robustness in the difficult environment. The CoCoRo project is developing such a swarm of AUVs to address complex problems in a decentralised manner, sharing information only locally between robots (Schmickl T.*et al.* (2011)).

CoCoRo's underwater exploration strategy comprises two swarms: an *exploratory shoal* of AUVs that navigate the sea bed, and a *Relay Chain* of AUVs capable of relaying messages to one another, thus maintaining a communication link between the shoal and water's surface. We focus in this paper on the Relay Chain algorithm. The Relay Chain is a highly dynamic swarm: each link in the chain independently coordinates its movement such that the chain tracks the shoal as it explores. The chain can recruit and return AUVs from and to the shoal into the chain to ensure it does not break when the shoal travels further way, and does not needlessly occupy AUVs when it returns. When there are no further AUVs that can be recruited, the chain can instruct the shoal to reverse course, further ensuring the chain does not break.

This paper presents the core Relay Chain algorithm, and an investigation into alternative strategies to deal with breakages in the chain when they occur. Despite the Relay Chain's dynamic and robust design, performing 3D collective coordination in noisy underwater environments using robots with limited sensor and communication ranges will certainly lead to AUVs in the chain losing contact with one another. We evaluate both a relatively naive recovery strategy whereby AUVs having lost contact with neighbouring links in the chain simply turn around and search for them, and a strategy based on AUVs localising one another through an absolute positioning system. Implementing an underwater absolute positioning system is highly challenging, and our analysis considers systems of varying accuracy, thereby highlighting how accurate such a positioning system must be to outperform the more simple strategy. The evaluation is based on the time required to send a message from the shoal to the water's surface, given that it may be delayed in an AUV that has temporarily lost contact with the next link in the chain. Recovery strategies must minimise the time for which the chain is broken, allowing the shoal to quickly report findings to the water's surface.

Both the Relay Chain algorithm design and the experiments reported here are performed in the CoCoRoSim simulation, which has been developed alongside, and calibrated to, the real-world CoCoRo AUV hardware in anticipation of its completion (Read M. *et al.* (2013)). Simulation represents a powerful tool for algorithmic development and evaluation, as sophisticated performance metrics can be applied with ease, and development can focus purely on algorithmic concerns free of hardware faults, and without the added time required to reprogram, deploy and collect AUVs. It has previously been employed in developing the exploratory shoal algorithm used here (Read M. *et al.* (2013)).

Our manuscript is organised as follows. First we consider the CoCoRo AUV configurations and capabilities, and the simulation environment. We summarise the previously published shoaling algorithm, used here by the exploratory shoal, before describing the Relay Chain algorithm itself. Thereafter we report our hypothesis-driven experiments to evaluate potential chain-breakage recovery strategies, and then conclude the paper.

AUV Configuration and Simulation

The CoCoRoSim Netlogo3D simulation reflects the design of the bespoke AUV hardware developed for CoCoRo (these AUVs are named "Jeff"), and has been calibrated to provide representative behaviour of the real AUVs. The Jeff platform is 25 x 12 x 5cm in dimension, uses thrusters to propel itself forwards or backwards and control its yaw, and can adjust its depth through a buoyancy pump. Omni-directional communication with a range of 50cm is provided by a radio-frequency (RF) modulator. Six bluelight sensor systems, comprising LEDs and photodiodes, located on the front, back, left, right, top and bottom of the AUV provide detection of obstacles and other AUVs. Each sensor system can perceive the distance to obstacles within an angle of 120 degrees and a range of 50cm; exact triangulation within this cone is not possible, only distance can be discerned (Scuola Superiore Sant'Anna and CoCoRo partners (2012); Universit libre de Bruxelles and CoCoRo partners (2013)).

The CoCoRoSim Netlogo3D simulation reflects sensor and actuator capabilities of Jeff, and its physics engine has been calibrated against empirical data generated from the previous CoCoRo AUV design (Read M. *et al.* (2013)). The water in the simulator causes drag for translational and rotational movement. The robot arena (Figure 5) is a rectangular tank with a fixed 'water level' as the top surface, the walls and floor of the tank are represented as solid surfaces which are detectable by the bluelight sensors.

Exploratory Shoal

The exploratory shoal uses an algorithm based on Reynolds' boids (Reynolds, 1987), the implementation of which is described in Read M. *et al.* (2013); only a brief summary is provided here. An AUV's movement is dictated by the weighted sum of three desired trajectories, calculated based on rules for cohesion, alignment and separation. Separation provides a trajectory away from other AUVs found within a threshold distance, and is designed to prevent collisions. The cohesion trajectory points towards other AUVs within detection range, and prevents AUVs losing contact with one another. The alignment trajectory reflects the average orientation of neighbouring AUVs, and seeks to align AUV orientations. The trajectories for each rule are calculated at regular time intervals, and their weighted sum governs AUV thruster and buoyancy pump actuation.

Separation and cohesion rules are informed by distance data provided by the bluelight sensors, with distance measures beyond 95% of the maximum sensor range excluded to ignore sensor noise. The separation threshold is 30% of sensor maximum range. The alignment rule uses data obtained from AUVs sharing their heading information with one another over RF.

Relay Chain Algorithm

The Relay Chain algorithm controls the initial formation and the maintenance of a communication link between the exploratory shoal and the surface.

Formation

A group of AUVs is deployed on the water surface, some of these will form the Relay Chain and the rest will join the exploratory shoal. The process is shown in Figure 1. All of the AUVs begin the process in the *explore* state (Figure 1). One AUV is chosen to be the 'seed' to start the formation. This becomes the *start* chain AUV (Figure 1). The *start* AUV remains stationary in the centre at the surface of the water. It is representative of any surface vehicle such as a fixed base station or a boat. The algorithm can also work with a mobile surface vehicle, with the chain following the vehicle in the same manner as the exploratory shoal, but for simplicity it remains stationary in the present experiments. This *start* AUV requests a number of vehicles to go to specific depths, each represents one layer of the chain. These will be the *middle* chain AUVs and the last one the *end* chain AUV (Figure 1).

For example in a water depth of 1m, five AUVs are recruited by the seed to initialise the chain. The remaining vehicles follow the chain downwards and form the exploratory shoal. The AUVs head to a target depth calculated using a user defined distance between AUVs. Once in position each AUV in the chain communicates with the AUVs either side of it, to keep at a depth halfway between the two of them. Each AUV in the chain knows the ID number of the 'next' and 'previous' AUVs in the chain, this is labelled from top to bottom so next is the one below and previous is the one above it. So that the algorithm remains decentralised each AUV is only aware of its local position and does not know globally where it is within the chain. An image of chain and shoal after initialisation is shown in Figure 2, chain AUVs are black and shoal AUVs are white.

Chain States

The Relay Chain algorithm splits the AUVs into task driven subgroups of: the exploratory shoal, the Relay Chain and navigation. These are the highest level states *explore*, *chain* and *navigate* in the state diagram (Figure 1).

Chain AUVs maintain a *position* state, to identify whether they are the *start* AUV at the top of the chain, a *middle* AUV or the *end* AUV at the bottom of the chain. The start and

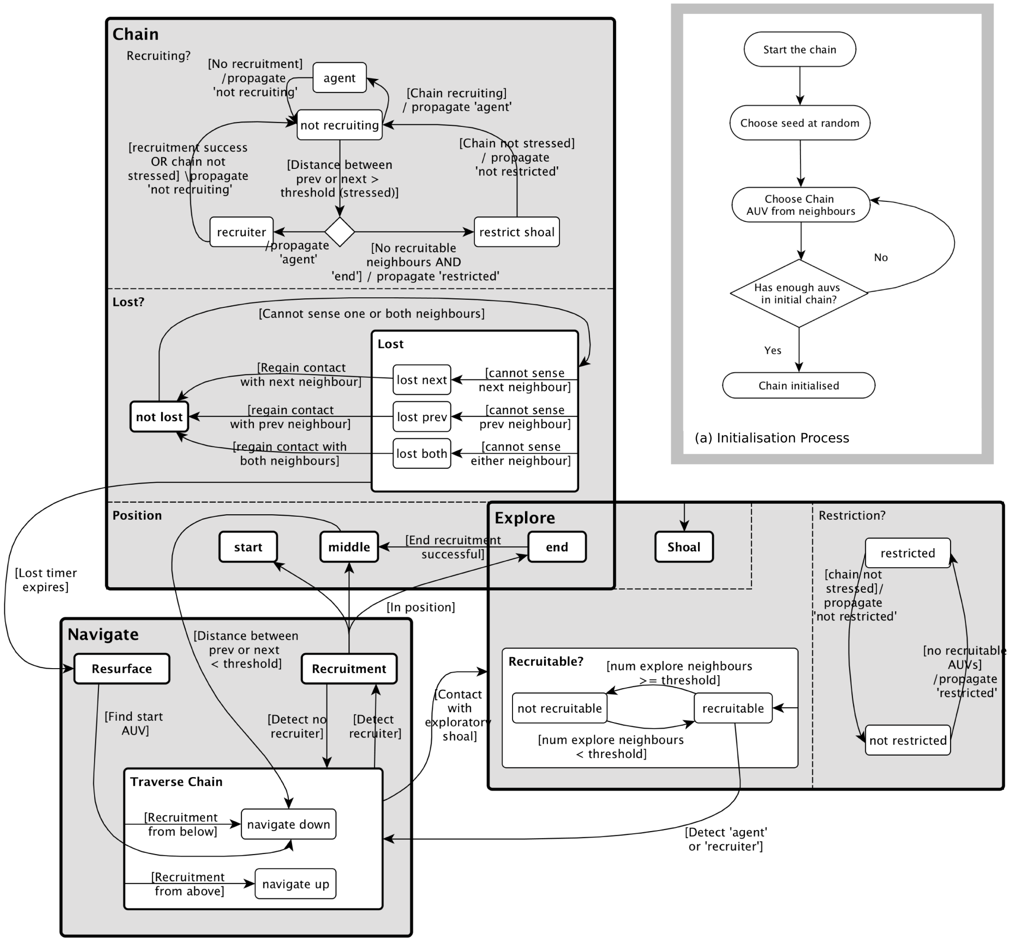

Figure 1: Complete state diagram illustrating the operation of the Relay Chain algorithm. The three core states are *chain* AUVs in the Relay Chain, *explore* AUVs in the exploratory shoal and *navigate* AUVs moving between *chain* and *explore*. The initialisation flow chart is shown in (a)

.

end AUVs are special cases. As discussed in the initialising section the start AUV remains at the water's surface and initialises chain formation. The *end* AUV is both a member of the chain and the exploratory shoal (see Figure 1). This means that it follows the shoal as it explores the environment and provides a link to the rest of the chain.

To maintain contact with the exploratory shoal the chain can dynamically change length by recruiting more AUVs or having AUVs leave the chain. The chain must add more AUVs to extend if the shoal moves away from the start AUV, so the chain has 'too few' members. It can also gain 'too many' members, for example if the shoal changes direction and returns towards the chain. Whether the chain has 'too many' or 'too few' AUVs is determined by thresholding the distance between chain AUVs.

To determine their desired position all *middle chain* AUVs monitor the distance and heading to the AUVs on either side of them in the chain. They aim to stay in a position halfway between the AUVs on either side of them. If the halfway distance exceeds a 'stress threshold' then it will stay nearer the 'previous' neighbour (one above in the chain). This is so that the 'stress' is transferred down the chain, nearer to the exploratory shoal; making it easier for AUVs to be recruited directly into the chain.

When the distance between the shoal and *end* chain AUV reaches a threshold the *end* AUV in the chain switches to the *recruiter* state (in the *Recruiting?* section in *chain* Figure 1). It propagates a message through the other chain AUVs to become *agents* to direct any *recruitable* AUVs to the site of the *recruiter*. The *recruitable* AUV will enter either *navigate up* or *navigate down* state depending on the direction it is given by the 'agent'.

A *recruitable* AUV is typically one from the exploratory shoal (*explore* state in Figure 1), but an AUV that is already

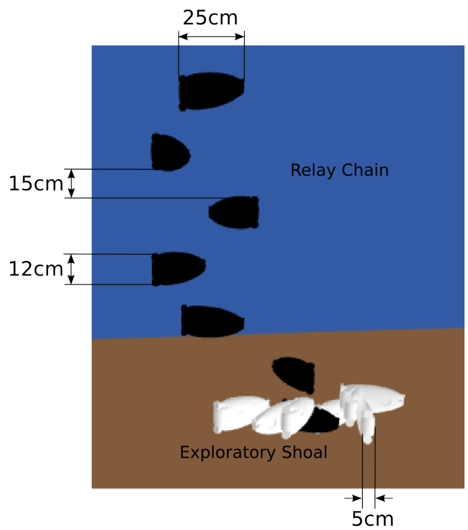

Figure 2: CoCoRo simulation environment, Relay Chain robots are black and exploratory shoal are white.

traversing the chain can also be recruited. If a *recruitable* AUV senses a *recruiter* or *agent* it enters the *traverse chain* state. Once it finds a *recruiter* the AUV will enter the *recruitment* state and manoeuvre into position to join the chain (as shown in the *navigate* state in Figure 1). All available AUVs will respond to a *recruiter*, but once the first responder is in position it joins the chain and the recruitment process stops (unless more AUVs are required). When a *recruitment* AUV can no longer see a *recruiter* it returns to the *traverse chain* state. When a *traverse chain* AUV cannot see a *recruiter* or *agent* and has contact with the exploratory shoal then it returns to the *explore* state. Once recruited, the new AUV will become the *end* AUV and the original *end* AUV will change state to become a *middle* AUV. (*Position* section in *chain* state Figure 1.

Once *recruitment* is successful, or if the distance between the end AUV and its previous neighbour has reduced below the 'stress threshold' by movement of the shoal, the *end* chain AUV switches to *not recruiting*. It will then propagate this message to the other chain AUVs to switch to the *not recruiting* state from the *agent* state (as seen in the *recruiting?* section in *chain* Figure 1).

If there are insufficient *recruitable* AUVs within range of the *end* chain AUV then the *end* AUV will pass a *restrict* message to the shoal. The number of *recruitable* AUVs required in the shoal is determined by a user specified threshold, its setting depends on the number of AUVs being used. Here for example with 15 AUVs in the simulation the 'recuritable threshold' is 2. The *end* AUV transmits a message to the shoal AUVs in range to enter the *restricted* state. They propagate this to other shoal members, outside of the range of the *end* chain AUV (in *restriction?* section in the *explore* state in Figure 1). The entire shoal is either *restricted* or *not restricted*, though individual AUVs may have opposing states while the message propagates through the shoal. Once in the *restricted* state the exploratory shoal AUVs will turn around (180 degrees) and head back towards the chain. This reduces the distance between the chain members and minimises the risk of the chain breaking.

To identify whether there are unnecessary AUVs in the chain each of the *middle* position chain AUVs compares the distance between themself and the AUVs either side of them in the chain to a 'slackness threshold'. Both this and the 'stress threshold' are a proportion of the bluelight range of the AUVs. If an AUV is too close to its neighbour, according to the threshold, the AUV will send an RF message to the next and previous AUVs in the chain announcing that it is leaving. The next and previous IDs of both neighbour AUVs are updated to remove the AUV from the chain. It will then enter the *traverse chain navigate* state and *navigate down* to the exploratory shoal as shown in Figure 1. Once it has contact with the exploratory shoal it changes to the *explore* state to join the shoal. The *traverse chain* AUVs move alongside the chain, as shown in Figure 3.

As shown in Figure 1, all *chain* AUVs have a *lost* status. A *chain* AUV is considered *lost* if it loses bluelight contact with either of its neighbours. This is further split to identify on which side the AUV has lost contact with. If an AUV remains *lost* for a given duration (e.g. 1 minute) then it is told to *resurface*. AUVs in the *resurface* state head to the surface at the coordinates they are currently at. They then use GPS to find the *start* AUV at the surface of the water, then enter the *traverse chain* state to rejoin the exploratory shoal or be recruited into the chain.

All of the AUVs utilise shoaling behaviours as described in the Exploratory Shoal section. The parameters are set differently depending on the desired behaviour, by weighting each of the rules' importance. For example; in the unrestricted *explore* state AUVs have a balance between cohesion, alignment and separation to produce shoaling behaviour. When in the *traverse chain* state the separation rule is most important, so that the AUV keeps moving rather than being attracted to a chain AUV and staying on one of the layers. The cohesion has a weight half that of the separation, so that the AUV follows the chain and does not head off in some other direction. The alignment rule is given a weight of 0, because the AUV is not in a shoal to align with.

Recovering lost AUVs

Knowledge of position is important for the Relay Chain algorithm, because the AUVs aim to maintain a position based on their neighbours. Using only the onboard sensors it is possible to work out the necessary heading locally. An AUV triangulates where the neighbour is by detecting which of its bluelight sensors can sense the neighbour. A problem arises when an AUV moves out of bluelight range of one or both of its neighbours. It no longer has any positional information

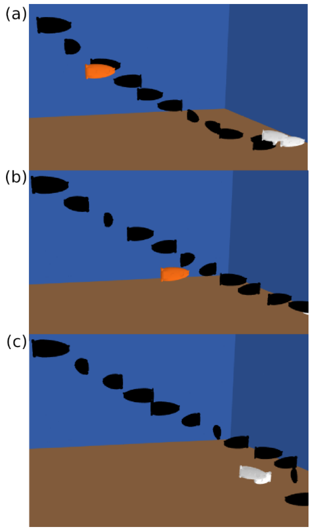

Figure 3: An AUV traversing down the chain to the exploratory shoal, the traversing AUV is shown in orange. Stage (a) shows it leaving the chain after the distance between it and its neighbours reduced below a threshold. Stage (b) shows the progress down the chain and stage (c) shows it joining the exploratory shoal.

about the neighbour(s). Such an AUV is considered 'lost' and each AUV has a lost status to store which neighbours (above or below) it has lost.

If one or more AUVs are lost, communications can no longer be passed along the chain. To solve the problem of finding the chain when lost, two alternative strategies are proposed and compared.

The first strategy is called 'Turn and Search', it is the simpler of the two strategies and can be performed without any additional hardware. In Turn and Search, when an AUV becomes lost it turns around by 180 degrees with a random offset of $\pm$ 5 degrees (chosen using a uniform distribution). A 100s timer is started, again with an added uniformly chosen random component of up to 150s. If the timer expires and the AUV has not refound the chain then it turns again. This process continues until the chain is found. An example path is shown by the trace in Figure 4. The AUVs still use their shoaling parameters and avoid collisions.

In Turn and Search recovery, if the AUV remains lost for over a threshold time of 400s then it resurfaces to find the chain from the top. The threshold can be adjusted depending on the desired task and importance of refinding the chain.

The second strategy is Absolute Positioning (AP). While this may generally seem a more suitable positioning solution than the use of bluelight triangulation, the accuracy of such systems is limited. As the AUVs being used are small the accuracy of the position is important. This strategy requires additional hardware, so is only desirable if Turn and Search is ineffective.

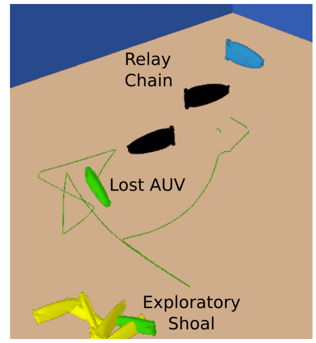

Figure 4: Example path taken by a lost AUV using Turn and Search recovery. The green AUV (fourth from the top) is lost and the trail is the path it has taken.

Positioning underwater cannot be achieved with GPS but there are different solutions including hydro-acoustic, image based and inertial systems. Of these, acoustic systems are most applicable to CoCoRo AUVs and tank based test scenarios. Image based systems use differences in the environment surrounding the AUVs, for example the seabed (Garcia et al. (2001)). This is not well suited to the tank environment used for testing the CoCoRo AUVs. Inertial systems are unlikely to offer any advantage over the existing system. They use measurements from gyroscopes and accelerometers on the AUV, so errors accumulate (Garcia et al. (2004)).

Acoustic systems use a set of beacons with fixed positions around the area in which the AUVs operate. The AUVs communicate acoustically with these beacons and the time for the signal to arrive is used to calculate the position (Alcocer et al. (2006)). The beacons can be attached to buoys with GPS receivers, so that they know their absolute position.

To provide specifications for a recovery system we compare the success of each method. We test three hypotheses:

1. 'Using AP rather than Turn and Search recovery will reduce the mean time to wait before a message can be sent along the chain.'
2. 'The higher the accuracy of the AP the lower the average time to wait before a message can be sent along the chain'.
3. 'As the accuracy of the AP is reduced there will be a point when using AP only when an AUV is lost outperforms using AP all of the time'

To test these hypotheses we run two experiments. In the first Absolute Positioning (AP) is used to control the AUVs all of the time and in the second a combined bluelight triangulation and AP method. In experiment 2 the AP is only used when the AUV is lost.

In both experiments the results are compared to bluelight triangulation with Turn and Search recovery and to a control using 'random' movement.

Method

The simulator maintains absolute position information for each AUV, so when lost the AUVs can head back directly to their neighbour. In reality underwater systems will not have this type of sub-centimetre accuracy. To achieve this discrepancy in accuracy an offset is added to the absolute position value of both the lost AUV and the neighbour that it is heading back towards. This offset is randomly chosen from the specified range to each of the x, y and z coordinates. The value is chosen with uniform probability because this is the worst case scenario. This means that the AUV knows the position of its lost neighbour (or neighbours) within a cube centred on its real position. A different offset is used for the AUV's own value and each of its neighbours' values. Each time the AUV becomes lost a different offset is generated. As the AUVs have a depth sensor accurate to 0.1mm no offset is added to the lost AUV's own depth coordinate.

The experiment is run in a tank with a width and depth of 5m and a water level of 1m. All AUVs begin at the water level, in the centre of the tank with randomised starting headings. There are 15 AUVs, of which 5 (including the seed) are set to specific depths as described in the initialisation section.

The seed is fixed in the centre of the tank at the water surface. The 4 other chain AUVs are initialised to depths 15cm apart. The rest of the AUVs enter the 'explore' state and head down the chain. They are either recruited or form the exploratory shoal. At least 2 of the AUVs must be in the *explore* state. This provides enough length in the chain to extend to each corner of the tank when exploring. An example of this can be seen in Figure 5.

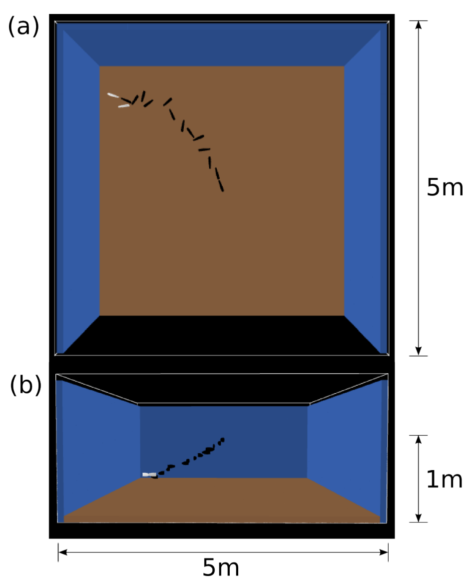

Figure 5: The Relay Chain in the tank used for experiments (a) top down view (b) front view.

In experiment 1 'partial AP' the AUVs will use bluelight triangulation when they are in range and AP when they are lost. For experiment 2 'full AP' the AUVs will always use AP to hold their position in the chain. Although they do not use bluelight triangulation the AUVs can still be 'lost' because once out of bluelight range they are out of communication range.

The simulation is run for 1 hour in simulation time. 1hr is sufficiently long for the shoal to explore the bottom of the tank. The simulation is repeated 500 times for each of the different AP accuracies. Offsets of between 0cm and 50cm are tested, 50cm represents a substantial offset as the AUVs are only 25cm by 12cm by 5cm. The range of accuracies is compared to the system without any absolute positioning, using the Turn and Search recovery method and also to a control version with 'random walk' movement with all AUVs in the Turn and Search recovery state at all times. The AUVs are given an initialisation period of 1 minute to allow them to reach their desired depths, then enter the recovery state for the remainder of the simulation run.

The success of the system is measured using the average time that it takes to send a message from one end of the chain to the other. This is important because the chain is necessary to convey messages, if the message takes too long then the information contained may no longer be valid.

The time between sending and receiving comprises two parts; firstly the time to physically transmit the data along the chain and secondly the time to wait until there are no lost AUVs so all AUVs can communicate with their neighbours.

The time to send the message itself is small compared to the potential time to wait until there are no lost AUVs. A typical message is 50 bytes and the AUVs can transmit at 28.8Kb/s. Each transfer will take 0.0139s and the chain can need no more than 13 transfers from one end to the other. Assuming negligible processing time the entire process will take at most 0.181s. The likelihood of the chain breaking while a message is physically being propagated is small, and the message can be resent if this occurs. The 'time to wait' component is therefore the critical factor, rather than the 'transmit time'.

The waiting time is determined by the contiguous time that the chain is broken for. For example if the chain is broken for a total duration of 60s then it is much preferable to have six 10s blocks than one 60s block. The chain is able to send a message in the gaps between the 10s blocks, rather than waiting the entire 60s before sending a message.

Results

The results using full AP are shown in Figure 6. In the graph 0 (no offset) is most accurate, each subsequent value is the offset in cm. The first column shows the control of 'random' movement when all AUVs are in Turn and Search recovery. The second column shows the system with no positional information other than the local bluelight triangulation.

The A test (Vargha A. *et al.* (2000)) is a non-parametric effect magnitude test which allows the comparison of the distribution of results for a comparison parameter with the distributions using alternative values. The A test indicates the probability that a randomly chosen sample from one distribution is larger than a randomly chosen sample from the other. A value of between 0 and 1 is returned by the test, with values higher than 0.71 or lower than 0.29 indicating 'large' differences between distributions. Large differences are assumed to be significant (Vargha A. *et al.* (2000)).

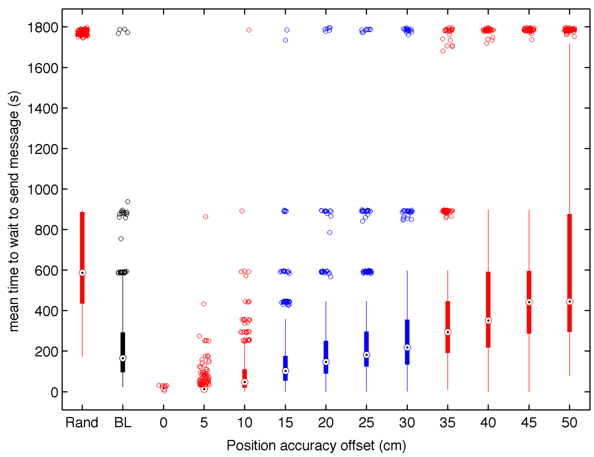

Figure 6: Mean time to wait before a message can be sent using full AP. Rand is random movement, BL is bluelight with Turn and Search (this is the A test comparison distribution) and numerical values are the AP offset.

The A test responses using different comparison values are provided in table 1. They show whether the difference between each column is significant. The colour of the column in Figure 6 represents the A test score of each distribution with a comparison to bluelight with Turn and Search (black in the graph). 'Red' columns are significantly different to the comparison distribution, 'blue' columns are not (values for each column are given in table 1).

From these the Turn and Search recovery method is significantly better than the random control. AP is significantly better than Turn and Search with an offset of up to 10cm, but significantly worse with an offset of 35cm or greater. An offset of 40cm or greater is not significantly different to the random control. When considering how much offset it takes before the system does not perform as well, 5cm is not significantly different to having no offset.

The results for partial AP (experiment 2) are shown in Figure 7. The A test responses are shown in table 1. For the partial AP, unlike full AP, all offsets are significantly better than the random control. Comparing the BL column with the AP columns shows that the AP values between 0 and 20 are significantly better than the bluelight with Turn and Search. For partial AP offsets of 5 and 10cm are not significantly different to having no offset.

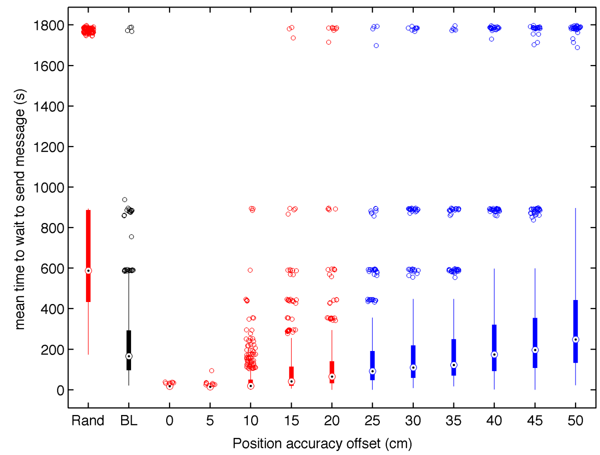

Figure 7: Mean time to wait before a message can be sent using partial AP. Rand is random movement, BL is bluelight with Turn and Search (this is the A test comparison distribution) and numerical values are the AP offset.

Discussion

The results presented above provide responses to the three hypotheses presented earlier. *Hypothesis 1 'Using AP rather than Turn and Search recovery will reduce the mean time to wait before a message can be sent along the chain.'* is shown to be true for both full and partial AP when the offset is 10cm or lower. The median average time to send a message (of the 500 runs) is 17.8s for full AP with 0 offset and 17.2s with partial AP compared to 166.4s with Turn and Search.

With too high an offset AP performs no better (or in the full AP experiment worse) than Turn and Search. This means for the hypothesis to be correct, the chosen AP system must be able to deliver accuracy to within 10cm for full AP and 20cm for partial AP.

Hypothesis 2 'The higher the accuracy of the AP the lower the average time to wait before a message can be sent along the chain' is also shown to be correct. It is true for both full and partial AP. There is a marked reduction in performance as the accuracy is decreased.

The system can function without significant degradation in the time to send a message, with an offset of 10cm when compared to a 0cm offset. When compared with a bluelight only system, an offset of up to 20cm is shown to be preferable. From this we can specify a system with an accuracy of 20cm or better be used.

Hypothesis 3 is 'As the accuracy of the AP is reduced there will be a point when using AP only when an AUV is

Exp	Comparison	A test result												
		Rand	BL	0	5	10	15	20	25	30	35	40	45	50
Full	Rand	0.50*	0.10	0.00	0.00	0.02	0.05	0.08	0.13	0.16	0.24	0.29	0.38	0.41
AP	BL	0.90	0.50*	0.00	0.03	0.19	0.33	0.46	0.57	0.61	0.72	0.76	0.83	0.84
1hr	AP 0	1.00	1.00	0.50*	0.33	0.81	0.98	0.99	0.99	1.00	1.00	1.00	1.00	1.00
Part	Rand	0.50*	0.10	0.00	0.00	0.01	0.03	0.04	0.05	0.09	0.09	0.15	0.16	0.21
AP	BL	0.90	0.50*	0.00	0.00	0.09	0.19	0.24	0.32	0.38	0.42	0.51	0.56	0.63
1hr	AP 0	1.00	1.00	0.50*	0.35	0.61	0.83	0.94	0.98	0.99	1.00	0.99	0.99	1.00

Table 1: The A test results for wait time experiments with different parameters set as the comparison distribution, here marked with *. Values higher than 0.71 or lower than 0.29 are significantly different (in red).

lost outperforms using AP all of the time'. Comparing the partial and full AP results shows that for values of 0, 5 and 10 there is no significant difference. For values of 15 and above the partial system is significantly better than the full system. An offset value of 15cm represents the switching point in performance. This result can be explained because with full AP the positioning is always altered by the offset. In partial AP the AUVs are able to use their bluelight sensors to triangulate their relative position, so are not affected by the offset at this time.

The clusters in Figures 6 and 7 can beexplained by AUVs getting lost and not finding the chain again. The highest valued cluster is at the duration of the run.

In summary, for AP to be worthwhile for recovery of lost AUVs to the Relay Chain it must be accurate to within 20cm otherwise bluelight triangulation is preferable.

Conclusion

The Relay Chain has been shown to be a decentralised method of providing a link between AUVs exploring the environment and the surface of the water. It is scalable, resizing as the shoal moves around the environment.

Restriction of the shoal's movement if there are not enough recruitable AUVs prevents the Relay Chain breaking and keeps the shoal in communication range of the surface vehicle. Messages can be propagated in either direction along the chain, relaying exploration data from shoal to surface or instructions to the shoal.

A disadvantage of the Relay Chain is that the number of AUVs used restricts the number available in the shoal. This is a result of limited communication range underwater.

Extensions where the chain is likely to be successful include fault tolerance, because the decentralised nature of the algorithm means an AUV can be removed and the chain reposition and continue operating. The introduction of a more challenging environment with low visibility water should be handled by the same recovery mechanisms used for lost AUVs in clear water.

The challenges of underwater communication and positioning mean that the CoCoRo Jeff AUVs can lose sight of their neighbours in the chain. Once lost, using only the available sensors it is difficult to refind the chain. Two alternative recovery strategies are used to overcome this. Either method provides some robustness to lost AUVs, though Absolute Positioning gives the best performance and is preferable to bluelight when the accuracy is within 20cm.

Acknowledgements

We thank Christoph Möeslinger and colleagues at the University of Graz for their contribution to CoCoRoSim. We also thank all other members of the CoCoRo Consortium.

This work is supported by the EC funded CoCoRo Project, GA 270382. JT is also part funded by the Royal Society and the Royal Academy of Engineering.

References

Alcocer, A., Oliveira, P., and Pascoal, A. (2006). Underwater acoustic positioning systems based on buoys with gps. In *Proceedings of the Eighth European Conference on Underwater Acoustics*, volume 8, pages 1–8.

Garcia, J.-E., Kyamakya, K., and Jobmann, K. (2004). Positioning issues in underwater acoustic sensor networks. In *WPNC 2004*, pages 129–136.

Garcia, R., Batlle, J., Cufi, X., and Amat, J. (2001). Positioning an underwater vehicle through image mosaicking. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 3, pages 2779–2784.

Read M. *et al.* (2013). Profiling underwater swarm robotic shoaling performance using simulation. *TAROS 2013*, pages 456–462.

Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. *SIGGRAPH Comput. Graph.*, 21(4):25–34.

Schmickl T.*et al.* (2011). Cocoro – the self-aware underwater swarm. In *SASOW 2011 Fifth IEEE Conference on*, pages 120–126.

Scuola Superiore Sant'Anna and CoCoRo partners (2012). Propulsion mechanism for swarm swimming and buoyancy-elastic mechanism. *CoCoRo Deliverable 1.2*.

Universit libre de Bruxelles and CoCoRo partners (2013). Report or publications on collective measures of the environment. *CoCoRo Deliverable 5.2*.

Vargha A. *et al.* (2000). A critique and improvement of the cl common language effect size statistics of McGraw and Wong. *J. of Edu. and Behav. Stats.*, 25(2):101–132.