Aberystwyth University

Robotic experiments with cooperative aerobots and underwater swarms

Honary, Ehsan; McQuade, Frank; Ward, Roger; Woodrow, Ian; Shaw, Andy; Barnes, Dave; Fyfe, Matthew

Published in:
Robotica

DOI:
10.1017/S0263574708004505

Publication date:
2009

Citation for published version (APA):
Honary, E., McQuade, F., Ward, R., Woodrow, I., Shaw, A., Barnes, D., & Fyfe, M. (2009). Robotic experiments with cooperative aerobots and underwater swarms. *Robotica*, *27*(1), 37-49. https://doi.org/10.1017/S0263574708004505

Robotic experiments with cooperative Aerobots and underwater swarms

Ehsan Honary, Frank McQuade, Roger Ward, Ian Woodrow, Andy Shaw, Dave Barnes and Matthew Fyfe

(Received in Final Form: February 27, 2008. First published online: April 16, 2008)

SUMMARY

SciSys has been involved in the development of Planetary Aerobots (arial robots) funded by the European Space Agency for use on Mars and has developed image-based localisation technology as part of the activity. However, it is possible to use Aerobots in a different environment to investigate issues regarding robotics behaviour, such as data handling, limited processing power, and limited sensors. This paper summarises the activity where an Aerobot platform was used to investigate the use of multiple autonomous unmanned underwater vehicles (UUVs) by simulating their movement and behaviour. It reports on the computer simulations and the real-world tests carried out and the lessons learned from these experiments.

1. Introduction

The use of multiple autonomous robots for practical applications has become more important over the years.[1-5] The main purpose of the multiple UUV (unmanned underwater vehicles) operations (MuOps) project within the BAUUV[1] program was to identify and de-risk critical UUV components and to raise the technology readiness level of the components required for the operation of multiple cooperative vehicles. Autonomous underwater operations are an attractive solution for challenges confronted in modern battlefields. A swarm of UUVs can collectively cooperate together to carry out missions or to survey a targeted area.

Robots operating on land, underwater, air or space can all benefit from collective activity, and despite the differences between the environments, the same principles and solutions can be applied to all of them.

The overall aim of the study was to evaluate the effectiveness of multi-robot solutions within an operational environment. In particular, the study emphasised the current state of technology as opposed to future theoretical possibilities. As a result, the study consisted of a literature review followed by a trade-off analysis to select the most appropriate operational problems. Various solutions were then investigated using simulations. Finally, suitable algorithms and strategies were selected based on simulation results, which were later validated with physical representative robots.

The MuOps study consisted of three main phases:

1. *Operational concepts*: A series of operational scenarios were considered as part of the MuOps study.

2. *Simulation of underwater vehicles*: This included a five-degrees-of-freedom dynamics model including a representation vehicle, drag, buoyancy and inertia along with an ocean environment model and a Proportional-Integral-Derivative (PID) model. A behaviour-based approach was used in conjunction with minimal collective robotics principles to guide a swarm of robots underwater. Example scenarios were simulated; the swarm was responsible to detect, classify and dispose of objects in the target area.

3. *Demonstration with physical hardware*: The algorithms were subsequently put onboard the Aerobots to validate the simulation results with real-world experiments. The Aerobots were used as an analogy to UUVs to move freely in a 3D space. They were tracked with a 3D positioning system and used horizontal and vertical thrusters to navigate the robots through the lab environment.

---

[1] The BAUUV program (Battlefield Access Unmanned Underwater Vehicle) was led by SEA on behalf of UK MOD. SciSys Ltd. Led the team to complete the MuOps study with contributions from University of Wales, Aberystwyth (UWA) and SCS.

4. This paper focuses on the implementation of the command and control algorithms and their integration onto the Aerobot platforms.

2. Trade-off Analysis

Various strategies (such as global versus local strategies and classical versus behavioural) were explored and evaluated in the context of the study. A set of potential scenarios was proposed based on the operational concept.

Table I. Various categories that the behaviours are scored against.

The criteria	Range
Difficulty of implementation in simulation	1=low, 5=high
Difficulty of implementation with Aerobots	1=low, 5=high
Difficulty of implementation in real-world environment	1=low, 5=high
Dependency on sensor technology	1=low, 5=high
Likelihood of success	5=low, 1=high

For each scenario, a set of behaviours was identified. Examples of behaviours required included deployment, navigation to target, flocking, searching, identification and tracking, localisation, obstacle and collision avoidance and inter-robot and external communication. The behaviours listed for each scenario were then categorised into the following:

- *Common behaviours*: These include behaviours such as deployment, localisation, internal and external sensors, flocking and obstacles avoidance.
- *Basic behaviours*: These include behaviours such as altitude control, depth control and heading control.
- *Specific behaviours*: These include behaviours such as disposal, hazard detention, searching, area division and threat identification.

A trade-off analysis was carried out to identify a suitable scenario that was both *useful* and *achievable* and that would be used for the remainder of the study. A *difficulty score* was used in the trade-off analysis which is calculated as follows. Each behaviour was scored separately based on different categories.

Each scenario was then scored based on the required behaviours in that scenario. The criteria, along with the range of score for each criterion, are shown in Table I. Note that the *difficulty of implementation* means the current state of the technology and it shows the difficulty of realising the complexity of the tasks required for each behaviour. Each scenario was also scored for usability. If it was very useful for operational purposes it scored 5 and if not so useful it scored 1. To calculate the final score for each scenario, the difficulty score was combined with a usability score using the following equation:

$$\text{Final Score} = \frac{\text{Total Score for Behaviour}}{\text{Usability}} \tag{1}$$

The behaviours were evaluated one by one and a table was created based on the scores and later, the final score was calculated for each scenario. The table showed that search scenarios are the most suitable and are within the capabilities of the current state of technology.

3. Current State of the Art

The selected scenario was primarily involved with searching and resource management. Searching methods can be divided into two categories: global and local algorithms.

- Global algorithms are generally used to find the most optimal way of covering an area. These are also used when information is readily available.
- Local algorithms rely on as little information as possible to do the search. Although, as there is not much information about the area, the results can be sub-optimal. Since the environment is usually not understood well, the preferred method is to use local algorithms. Hence only this category is explored in this section.

The problem of multi-robot exploration or coverage with limited range communications is stated as follows:

> *n identical robots set out to explore or cover an unknown area. Each robot is equipped with sensing, localisation, mapping and limited-range communication capability. Design a coordinated algorithm to carry out the mission reliably and quickly.*

A few examples of the work carried out in this field are presented below:

- Work carried out by Rekleitis *et al.*[6] involved a set of robots that covered an area by dividing it into cells. The robots only used local communications that were available only within line of sight. The robots used cellular decomposition (based on Boustrophedon decomposition) to try and cover each cell in the area. Covering all cells leads to full coverage. Each robot had an internal representation of the world which was updated and shared whenever two teams of robots came within line of sight of each other and joined into one team. This research was carried out successfully in simulation in a 2D environment. They claim that this algorithm greatly reduced repeated coverage which is the main concern when local algorithms are used.
- Similar work to the above was also carried out by Sheng *et al.*[7] They used a distributed bidding system and local sharing of the internal representation of the environment to navigate around the environment. This work has been demonstrated by a 2D simulation. They claim that the distributed bidding algorithm is efficient and simpler compared to existing multi-robot exploration algorithms and that it requires significantly less communication.
- Other work on area coverage and localisation carried out by Rekleitis *et al.*[8] uses multiple robots to reduce the odometry error of other robots when they move through an environment. Basically, one robot moved while the other robot(s) observed its movement and then they all moved in turn. This greatly reduced the odometry error in comparison with pure dead reckoning. Although, it is assumed that the participating robots have full communication capability. This work is very promising and inspirational, though not necessarily beneficial to the underwater robotic world as the stationary UUVs are prone to currents affecting their positions.
- Work has also been carried out on algorithms that integrate graph theory and virtual function field approach.[9] They used this algorithm in a 2D simulation for the deployment of a set of robots in constrained and unconstrained environments. Their simulation results showed that their algorithms were valid and that it enabled the system to reconfigure itself such that the area covered by the system could be enlarged.

Most of the above work has only been carried out in simulation and sometimes in a 2D lab environment. Implementing these methods in the challenging environment of the underwater world is not trivial. Searching capability is an important behaviour used by the UUV swarm in various scenarios and as a result it required thorough attention during the study.

For the purpose of efficient searching and maximum coverage, random ad hoc swarming might not lead to optimum results. On the other hand, a formation hold by the UUVs can help maximise coverage and reduce multiple visited areas.

4. Formation Flying

In the context of underwater robotics, there are two main uses for formation control:

1. Optimised configuration of robots to maximise the coverage of an area in a given time.
2. The creation of a formation as a desired behaviour, for example the geometrical formation of a set of UUVs that act as a sensor group. If the formation is required for coverage of an area, then this behaviour is closely related (or even merged) with the searching behaviour explained in the previous section.

There has been extensive research on formation robotics with various algorithms being developed for 2D and 3D environments. Flocking and local minimal algorithms using only local knowledge have been used and tested on real robots for formation control.[10,11] There has also been work on the integration of formation control with other behaviours. For example, work has been carried out on wheeled robots that simultaneously incorporate reactive behaviours for formation control, obstacle avoidance and navigational goals.[12] These robots try to keep in formation as they navigate through the environment. Fredslund and Mataric[10] have implemented a local minimal algorithm both in simulation and on wheeled robots for creation of geometric formations.

For the underwater world, UUVs have been used to research formation control and adaptive sampling with a group of underwater gliders.[13] These underwater gliders use the same principles as aerial gliders except that they use lift to create motion in water. Underwater gliders control their movement by changing their buoyancy alone and hence follow a saw tooth trajectory instead of using conventional propulsion mechanisms. Research has been carried out to make a group of gliders follow temperature gradients while maintaining prescribed formation orientation and vehicle spacing. The researchers claimed that their results show a good potential for cooperative formation control in gradient climbing and feature tracking for physical and biological processes.

During this study, a number of formation flying techniques were tested until a version was selected which performed well in all simulations. This particular algorithm was a cross between the classical Reynolds flocking algorithm[14] and lane survey algorithms used for area search.

5. Simulation Architecture

The simulation tool is divided into four primary blocks with their corresponding sublevel modules as shown in Fig. 1 (left).

1. *Dynamics module*: Dynamics of all the UUVs and the underwater environment simulation:
(a) *Environment model*
(b) *Translational dynamics*
(c) *Attitude dynamics*

2. *Guidance and sensors*: Guidance, navigation and control management:
(a) *Sensors and navigation*
(b) *Behaviour*: Controlling the overall behaviour of a UUV
(c) *Guidance, control (PID) and propulsions models*

6. Guidance Architecture

The purpose of the guidance algorithms was to generate the target Cartesian coordinates for the future locations of the vehicle. The control module (either in the simulation or in the real world) was then responsible for implementing the described commands. By continuously generating a series of target positions, the UUVs can be controlled to move and function within the environment.

During the simulations, the calculation of the eventual behaviour of UUVs was carried out in the guidance algorithms. This means that commands for the actuators and payloads were all issued within this block. Special care was taken to create an architecture which allowed the generation of multiple scenarios, based on different numbers of UUVs and different behaviours that could be active at any one time.

The architecture of the guidance algorithms was based around the implementation and use of multiple behaviours for different purposes. The core of the architecture was designed using DAMN[15] (distributed architecture for mobile navigation) and is shown in Fig. 1 (right).

7. The Arbiter

Generally, groups of distributed behaviours communicate with a centralised command arbiter. Each behaviour is responsible for the control of certain aspects of the UUVs' functionality. The behaviour then sends its output to the arbiter which generally satisfies the behaviour's objectives. The arbiter is then responsible for the combination of the outputs from the various behaviours and generates an action based on these inputs.

The outputs of all behaviours are combined in the arbiter based on modes and weights. Certain behaviours can have precedence over others. For example if the maximum depth control needs to be active, then it would override another behaviour for the calculation of the vertical component of the commanded position (Z). In this case it would not make sense 40 *Robotic experiments with Aerobots* Fig. 1. (Left) Simulations modules and their interconnection. (Right) Guidance architecture. to combine any other behaviour's command with maximum depth control.

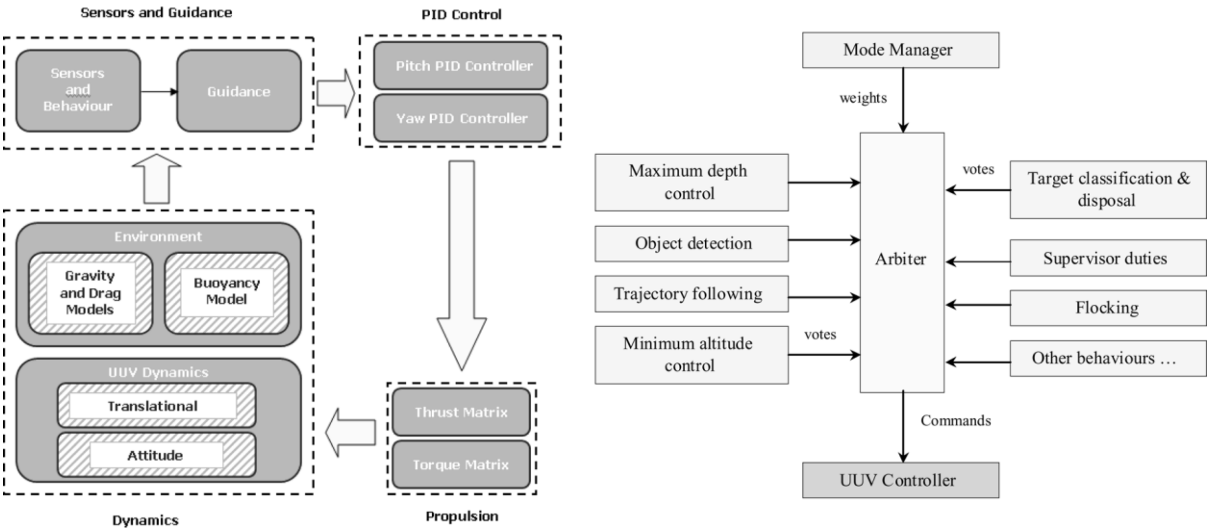

Fig. 1. (Left) Simulations modules and their interconnection. (Right) Guidance architecture.

Each behaviour outputs a target position (in appropriate dimensions) and these are combined with preset weights, which were found by experimentation. Generally, a higher weight was given to trajectory following than to other behaviours forcing the robots to strongly follow the trajectory rather than wander off when performing formation flying. In some cases, special functionalities (such as activating actuators) take place within the behaviour itself and are not passed to the arbiter, since a particular behaviour might be the only behaviour that controls an actuator.

Modes controlled the overall behaviour of the UUVs. Examples were trajectory following while formation flying, classifying, disposing, replenishing and so on. The update frequency was chosen as 1 Hz as through experimentation it was proved to be the most efficient for running the simulations in order to provide reasonable and accurate data.

With an arbiter, the UUV can benefit from the dynamic interactions between various behaviours as well as having rigid control which can be the result of deliberative control imposed by the arbiter.

8. Influential Parameters

In the context of the reference scenario, there were a variety of scenarios that could take place. Certain parameters have more impact on the overall performance of the system than others. The following are perhaps the most important parameters in the context of MuOps:

- Cooperation versus non-cooperation
- Heterogeneous swarms versus homogeneous swarms
- Variations in number of UUVs
- Cooperation using various algorithms such as flocking, formation flying and leader–follower
- Impact of communication (or lack of it) on cooperative solutions
- Analysis of coverage, detection, classification and object disposal rates under different scenarios

9. Study Scenarios

The intention of this study was to shed light on previously mentioned issues and to provide results based on simulation and then analyse and justify the conclusions.

For this, two sets of scenarios were selected:

- *Scenario 1: Detection (Coverage)*: This contained the following cases: single UUV, non-cooperative UUV and cooperative UUV swarm.
- *Scenario 2: Detection, classification and disposal*: This contained the following cases: single disposal UUV, homogeneous disposal swarm (all members identical), heterogeneous disposal swarm (with different types of robots).

10. UUV Types

This section briefly summarises the three main types of UUVs used during the simulations and demonstrations. The actual requirements for each UUV depend on the selected scenario and solution.

- *Type I*: Detection UUV: This UUV was equipped with appropriate sensors (such as sonar) for object detection purposes.
- *Type II*: Communication UUV: This UUV was capable of communicating with the outside world as well as with other UUVs. Hence, it was the point of contact for a group of UUVs.
- *Type III*: Classification & Disposal UUV: This UUV is capable of classifying detected objects. Once it has classified the object as requiring disposal it can move on to dispose it by using its finite number of 'disposal' payloads. This UUV needs to be re-armed each time the payloads are depleted, by returning to a pre-defined replenishing point.

11. Simulation of Scenarios

11.1. Detection

As part of the simulation phase, scenarios were simulated case by case. Key parameters (such as drift, number of UUVs and communication) were varied for multiple runs and the results were graphed and analysed. Measurements of performance such as coverage and mission time were used to compare the results between the cases.

The UUVs and ocean environment were simulated using Matlab-Simulink. The results were later visualised in a SciSys in-house 3D visualisation program. As part of this exercise different cooperative algorithms such as hybrid flocking, leader–follower and formation flying were considered. Continuous and periodical drift was added and the effects on the overall behaviour of the swarm were examined. Some representative results are presented below.

11.2. Results with drift

A number of simulation runs with an ocean model and noncooperative behaviour were carried out. In order to compare formation flying with non-cooperative behaviour as far as coverage was concerned, it was beneficial to consider a relatively realistic environment where navigation drift was present. The drift was calculated based on the travelled distance and consisted of the following:

1. Bias: The bias was added to the velocity at each time step and was calculated based on the nominal speed of the vehicle and time. The error consisted of a magnitude and a direction. The magnitude was set to be a percentage of the travelled distance, and each UUV could have a different bias. The magnitude of the bias was randomly selected for each UUV with mean of 0.2 ± 0.02%. A direction was also selected randomly between 0 and 2π. The error was then added to each axis based on the magnitude and direction.

2. Random noise: This was a random velocity added at each time step to reflect noise. The power of this noise was set to 0.001 m/s.

In line with how UUVs operate in the ocean, the drift was reset when the simulated GPS update reset was received by the NAV module from external devices.

The results of formation flying and the drift case are shown in Figs. 2 and 3 for an area of 400m × 600 m. As it can be seen, the difference between the coverage rates is very small. There was bias towards the development of this difference as time goes by, though at the end both swarms manage to cover the whole area. Figure 3 (left) shows the coverage percentages over time. At the end of each raster leg, the drift error gets reset. When more drift was introduced with no reset, the effect of the error was more pronounced (Fig. 3 (right)).

Figure 4 (left) shows the coverage over time for cooperative and non-cooperative cases. As it can be seen, the difference between formation flying and non-cooperative swarms was not large when drift was present, although the cooperative case had a faster execution time (see Fig. 4 (right)). Since the swarms turn at different times, the difference between their coverage is periodical as shown in the graph. This difference is increased over time but returns back to almost zero at the end. The important result was the difference of coverage at the end of the run. In this case, this difference was very small. With the algorithms and sensors implemented, errors in coverage and position drift due to navigation errors were similar. However, other implementations (such as relative position sensors) may be able to minimise vehicle navigation drift.

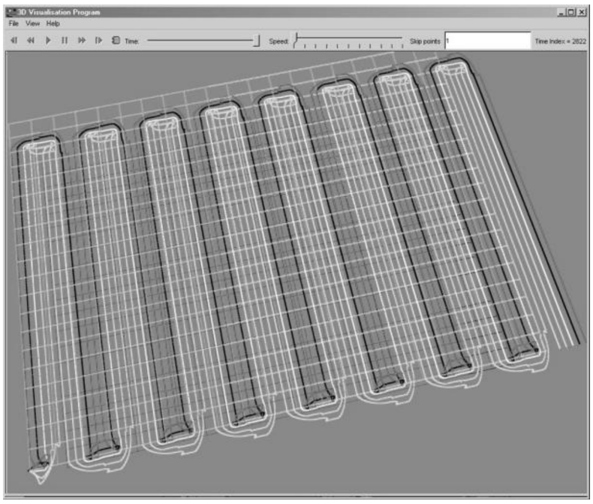

Fig. 2. Swarm of five UUVs, while formation flying. The drift is set as 0.2%.

11.3. Classification and disposal

Classification and disposal scenarios introduced the concept of homogeneous and heterogeneous swarms. A heterogeneous swarm was tasked to detect the objects in a pre-defined area with detection UUVs (Type I) and then classify and dispose of targets with dedicated disposal UUVs (Type III). Simulated sensor models with probabilities of success were used to determine if a detection or classification had been successful. Detection, classification and disposal rates were simulated based on probability models. A disposal UUV could potentially run out of disposal payloads. If it did, then it had to go to a pre-defined location to replenish and then continue with the mission.

Figure 5 shows typical results for a heterogeneous swarm of UUVs. A number of UUVs are performing a raster scan survey and a number of other UUVs collect commands from a supervisor UUV (nominated among the swarm members) for specific tasks, such as classifying and disposing of a target. The collective behaviour helps the swarm to achieve the task much more efficiently and quickly, minimising energy expenditure. There is no central control; the swarm reacts based on the information collected by the distributed sensors on the UUVs, a decentralised flying system. The supervisor only coordinates the classification and disposal of targets and can issue commands to set the formation of the swarm to a desired configuration. This is part of the deliberative layer that controls the behaviour of the group so that it can carry out the mission. The swarm should be optimised for the task in hand. Hence, there is a balance to be made when more sophisticated heterogeneous swarms are used.

Formation flying has certain advantages over noncooperative solutions. An example is the ability to reform the formation to a new shape, perhaps for better sonar performance – an example is shown in Fig. 6 (left).

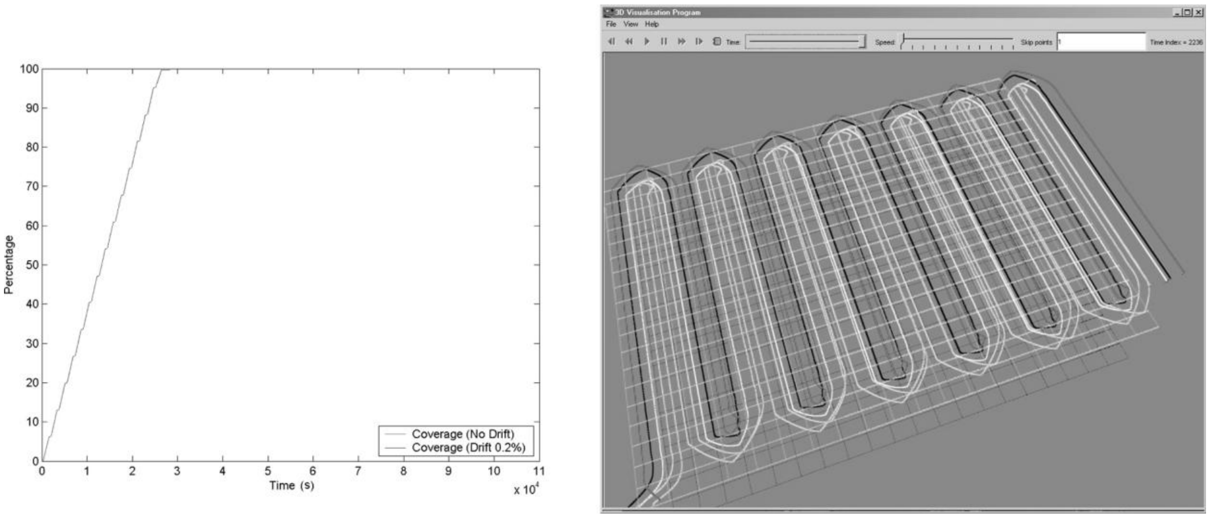

Fig. 3. (Left) Coverage over time of a swarm of five UUVs, while formation flying with drift set as 0.2% and no drift. (Right) Trajectory of the formation flying case with drift (2%) that was not reset periodically.

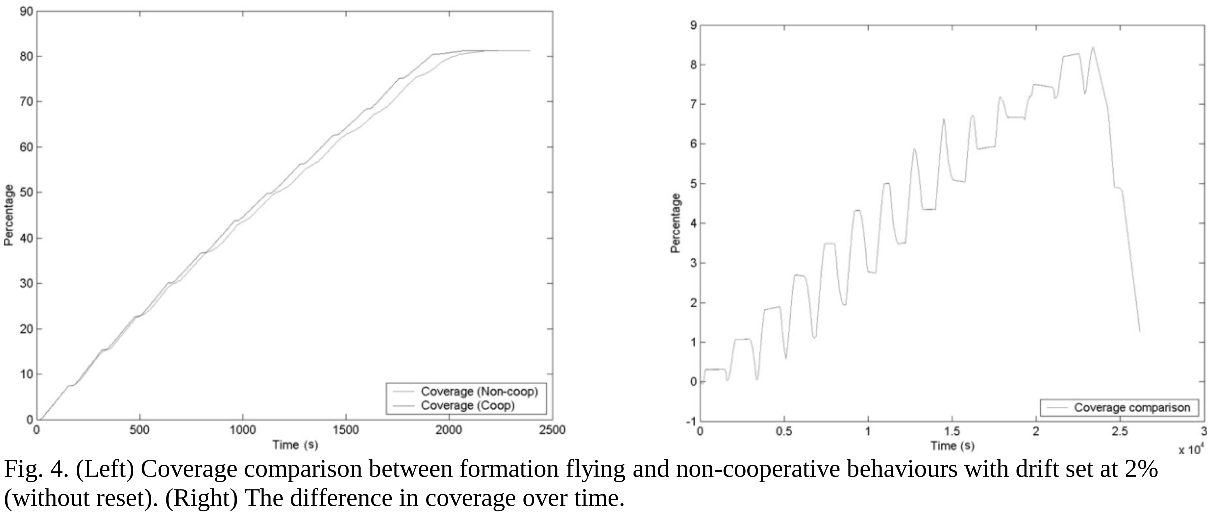

Fig. 4. (Left) Coverage comparison between formation flying and non-cooperative behaviours with drift set at 2% (without reset). (Right) The difference in coverage over time.

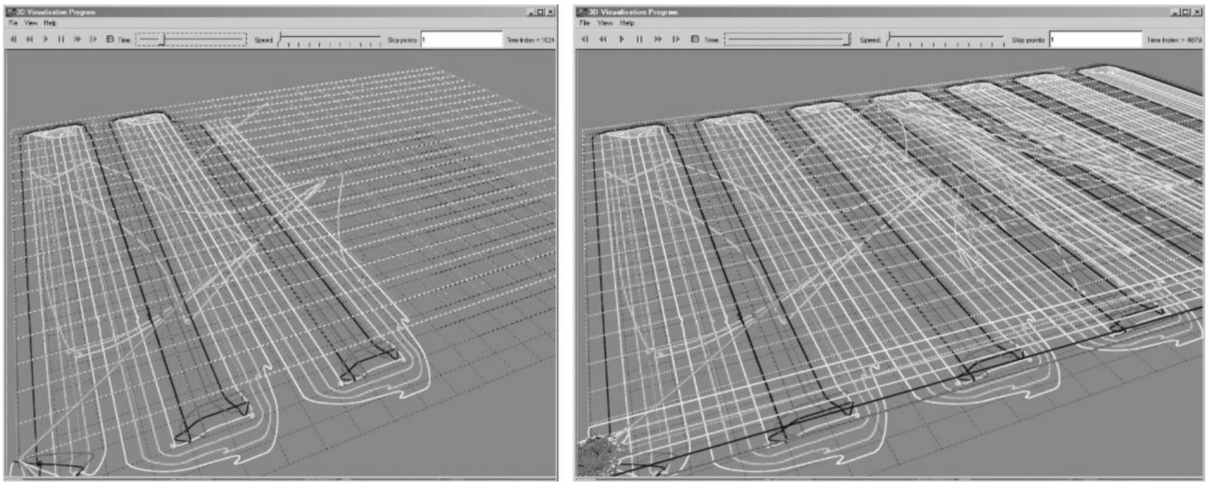

Fig. 5. A heterogeneous swarm consisting of six Type I UUVs and five Type III UUVs operating simultaneously.

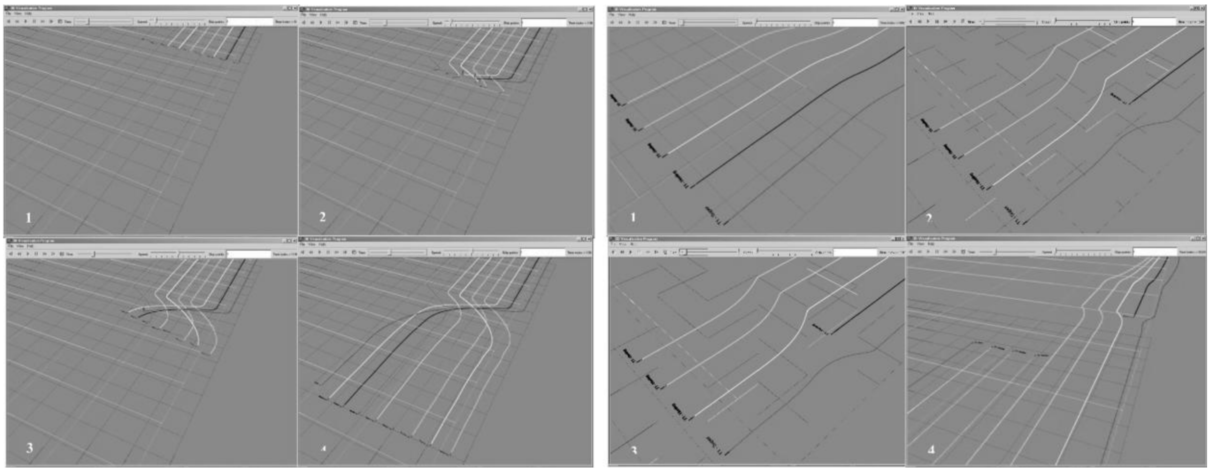

Fig. 6. (Left) A number of UUVs switch to a new formation in mid track. (Right) A UUV has failed; other members reform to cover the missing track.

Swarm member loss is also another important possibility which was addressed. Figure 6 (right) shows what happens when a member has failed. The swarm reconfigures itself based on the new situation to cover the track of the lost UUV. The communication and commanding requirements for this activity is minimal. It was possible to increase the size of the swarm in simulations without requiring an increase in CPU power and memory size on individual vehicles. The limiting factors on the size of swarm mainly depend on the resources available in the environment, such as the overall communication bandwidth, physical size of the environment and the vehicles and obstacles. The algorithms did not enforce any limits.

12. Relationship Between Transit, Number of UUVS and the Covered Distance in Search

An important parameter associated with a group of UUVs is the number to be used in a mission. What is an efficient number of UUVs to search a specific area for a given transit distance? This will then lead to finding out their required energy budget, which is an important parameter.

Suppose N UUVs are searching the area with a raster search pattern as shown in Fig. 7. To see the effect of number of vehicles (N) on range, a number of graphs were calculated for various N based on study parameters as shown in Fig. 8.

The results show that as the number of UUVs was decreased, the range requirement was significantly increased. Note that the range requirement stayed approximately the same as the number of UUVs was increased from 10 onwards. This was due to the number of legs that the swarm needed to travel. The minimum number of legs was considered to be 2 where 38 UUVs were needed to scan the area. Using this graph, the minimum number of UUVs could be selected based on a given range requirement.

The bottom graph shown in Fig. 8 is the ratio between the distance travelled to search the area and the distance travelled for transit. For a typical scenario, when a low number of UUVs are used, the search distance can be as high as eight times that of transit distance.

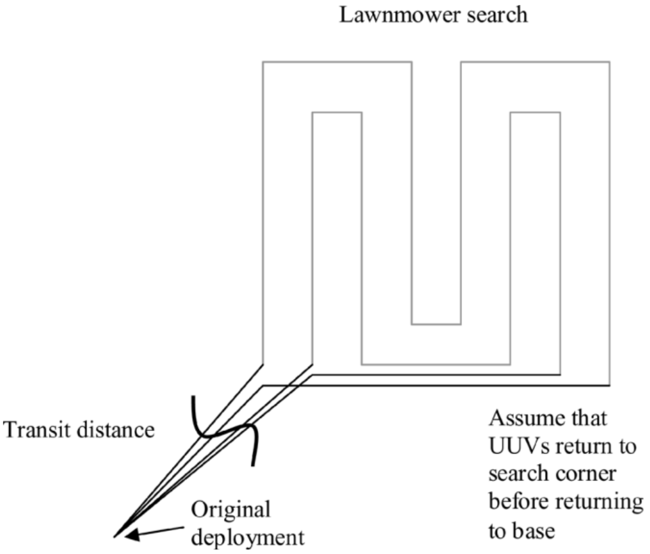

Fig. 7. Raster and transit trajectories for calculation of travelled distance.

This ratio reduces below 1 when the number of UUVs exceeds 13. This means that the transit distance of a typical UUV of a swarm consisting of 13 members is approximately the same as the distance travelled during searching. As more UUVs are used, this ratio is further reduced but not significantly.

If a single UUV searches the area with a raster pattern, the range would require a large UUV (Autosub developed by the National Oceanography Centre, Southampton, has an operational range of 400–500 km).

To better illustrate the relationship between the transit distance and the number of UUV, a new graph was calculated based on the same criteria, as shown in Fig. 9. Transit distance was graphed for each N for a set of UUV ranges. The results show that the achievable transit distance was sensitive to the number of UUVs used. For example,

- If a transit distance of 40 km is used, the graph goes to infinity. Any UUV with less than 40 km range is incapable of carrying out the task.

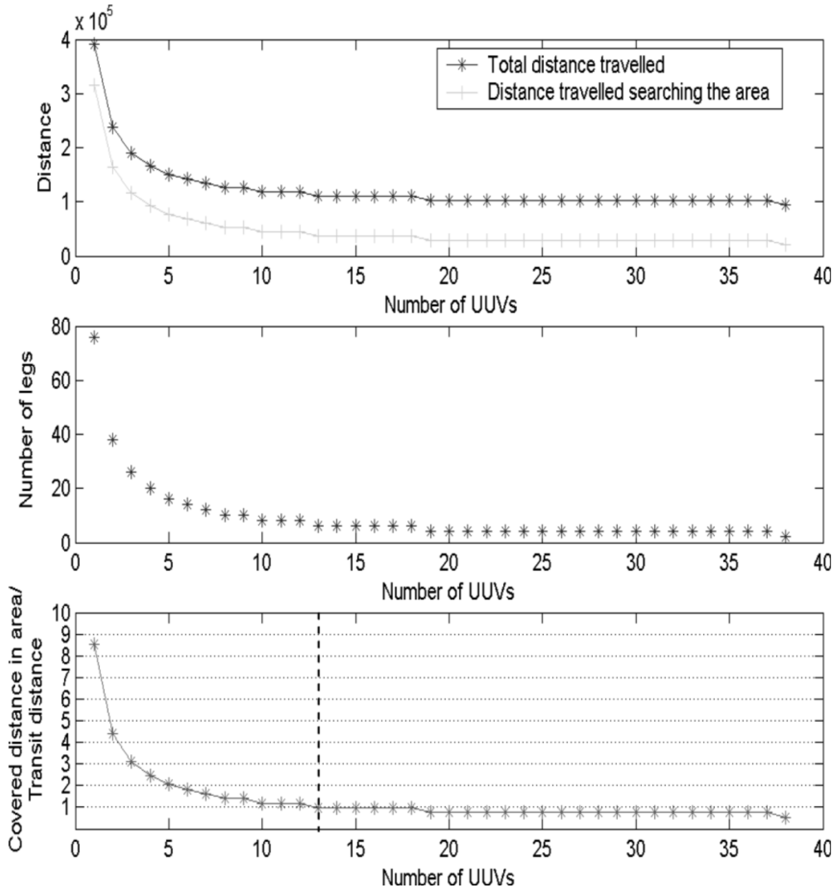

Fig. 8. The impact of using different numbers of UUVs.

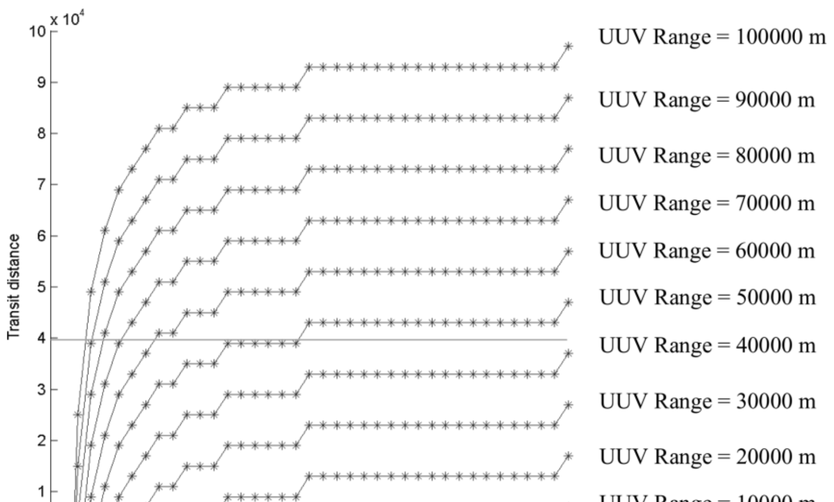

Fig. 9. Relationship between the number of UUVs and the achievable transit distance for a given UUV range.

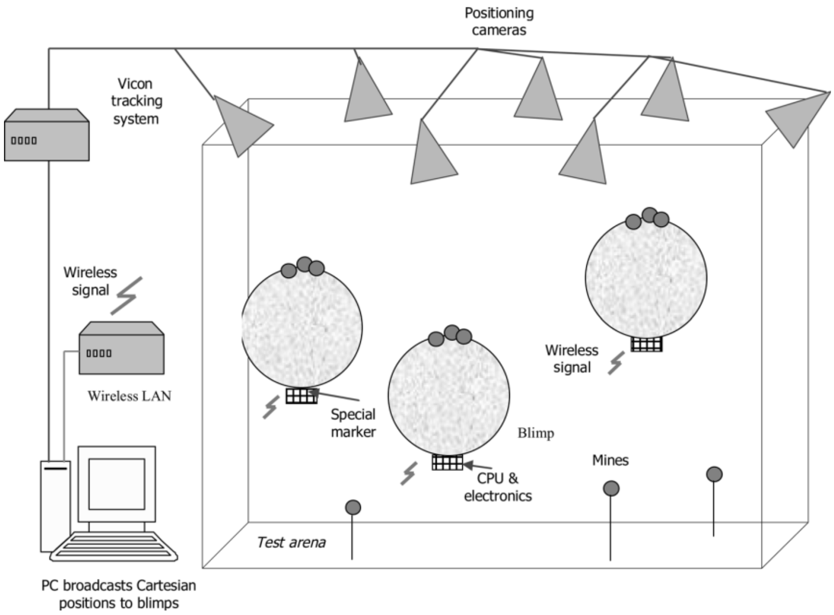

Fig. 10. Vicon system and the test environment setup.

- If a UUV with 50 km range is used, then approximately 19 UUVs are required to cover the area.
- If a UUV with 60 km range is used, then this is reduced to eight UUVs (a significant change).
- With a 100 km range, there is still a need for three UUVs.

The graph can be used the other way around as well. For a given number of UUVs and a given range, how close should they be to the target area when they are deployed? For example, if 10 UUVs are available with a range of 80 km, they should be deployed at a maximum distance of 65 km from the target area. However, if UUVs can be deployed 15 km closer to the target (50 km transit distance), then only half of the UUVs need to be deployed.

13. Demonstration with Aerobots

Simulators are powerful tools to rapidly experiment with different algorithms. However, by definition, a simulator represents an abstract representation of the real-world environment and it is desirable to validate simulation results with real-world hardware. The purpose of the third phase of the MuOps study was to demonstrate the simulation results with real-world experiments. These experiments were based on free flying balloons (Aerobots) as these are a good analogy for real-world UUVs as they can freely move in a 3D environment, and have similar issues regarding limited communication, hardware and energy resources. The Aerobots were supplied by the University of Wales, Aberystwyth (UWA). SciSys has used Aerobots for different projects in the past, such as the European Space Agency's (ESA) Planetary Aerobots,[16] and the platform has evolved through the different studies.

The tests were carried out under SciSys supervision in the Great Hall at the UWA. This was a large auditorium which proved to be a suitable environment for the setup of Vicon motion tracking cameras and provided the large environment required to test three Aerobots at once. Positioning was achieved using a Vicon motion tracking system which used 12 cameras with Infra Red capabilities to track custom reflective markers, as illustrated in Fig. 10.

The Aerobots were used to simulate the movements of the UUVs. They were preferred to wheeled robots as they can only operate in two dimensions. By running test scenarios with Aerobots that closely resembled those that were simulated for the UUVs, it enabled direct comparison and helped raise appropriate issues regarding the operation of a swarm of Aerobots or UUVs.

13.1. The Aerobot platform

The Aerobot platform was a helium-filled balloon with a PC used as the gondola. Since the platform had already been developed as part of ESA's Planetary Aerobots project, the team had extensive experience in using it. As a result, the real world risk of physical demonstration was reduced. Examples of these real-world problems were general electronics and mechanical issues associated with the development of robots.

The Aerobot was capable of controlling its height using two thrusters as well as maneuvering around with four side-thrusters. It used Linux as the operating system and communication was through a Wireless LAN. SciSys's message transfer system (MTS) was used to manage the broadcast of the communications. The Aerobots were localised using the Vicon 3D positioning system. This used 12 cameras to triangulate the position of reflective markers attached to an object at the location of an object within the target space (see Fig. 11). This position was then fed to sensor simulation modules which in turn was passed to the guidance block.

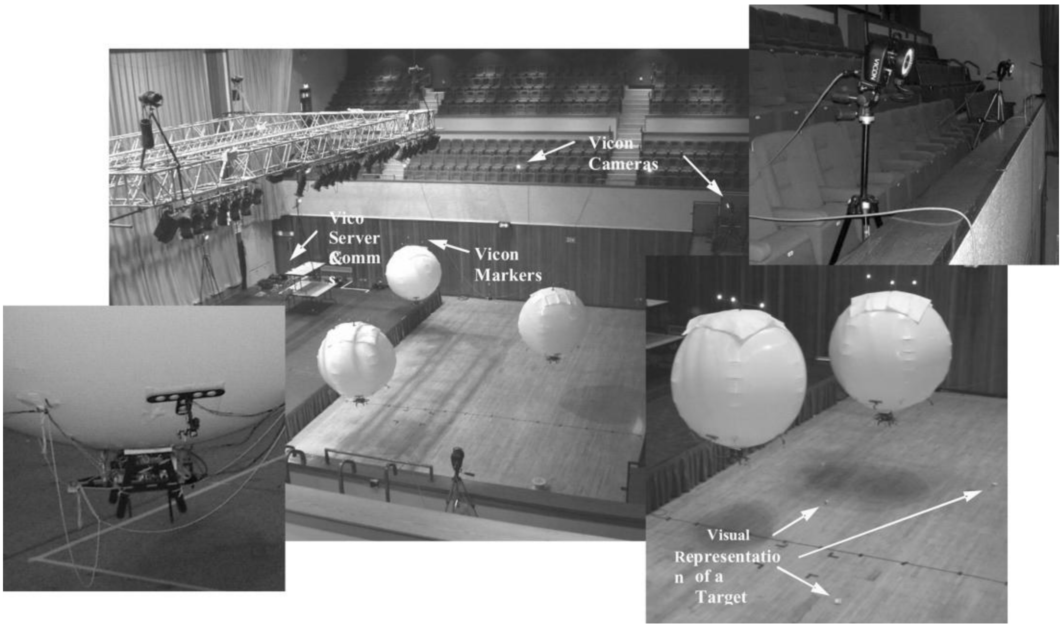

Fig. 11. Great Hall environment with the Aerobot test bed setup.

The Aerobot platform should only be compared to its equivalent UUV model in a qualitative way. It was not meant to be a direct replacement of it and as such no meaningful quantitative comparison can be made. The value of using the Aerobot platform lies in its cheaper operating costs and that it raised similar challenges as those that would be raised using UUVs. Its capability to move in a 3D environment was very valuable. The Aerobot used a decoupled thruster system, where horizontal and vertical thrusters were operated separately to control the movement of the robot. UUVs are controlled in two ways, decoupled or coupled. Underwater remotely operated vehicles (ROVs) usually operate with decoupled systems where there are separate thruster controls for vertical and horizontal movements. Another example is any underwater vehicle that operates similar to submarines where the vertical movement is controlled by a bladder system while the horizontal movement is controlled by propellers. Aerobot was an excellent substitute for these kinds of vehicles since the controls are very similar. On the other hand, the coupled system is used in vehicles that are similar to torpedoes.

Note that the simulation campaigns were carried out using a coupled control system. However, the algorithms were ported to the Aerobot platform with a decoupled control system. As it will be explained shortly, the algorithm functioned just as they did in simulations which suggested that the algorithms were not dependant on the underlying control mechanism nor was there a need for roll and pitch control in this context. The behavioural algorithms were designed from the outset to function as high-level command and control. This proved to be an efficient approach when it was necessary to port them to different platforms.

Another important component of the system was the Vicon position system, as the positioning systems used underwater are usually very different. However, any positioning system may produce errors. The Vicon system used here was no exception as it produced errors similar to underwater positioning systems. There were generally two types of errors, positioning and anomalies. The positioning error was in the order of ±5 cm on the actual calculated position. A stronger and more intermittent error was the inability of the system to get a fix on the position of an Aerobot and therefore reporting a missing position. Sometimes, the system reported positions in a reflector location in relation with the reference frame. These anomalies had to be filtered and dealt with similar to solving anomalies in underwater positioning systems.

There are also certain features that were not present in the Aerobot system. For example, latency of communications based on the variable speed of sound in water is not naturally present in the environment of the Aerobot. Viscosity of air and water are quite different as well, which requires different system dynamics. Note that errors such as these can be simulated and integrated into the system.

What mattered was that Aerobots were used in a noisy environment and that their positioning was not guaranteed. The formation flying algorithm had to perform well in this noisy environment.

14. Formation Flying With Aerobots

A series of experiments were carried out using up to three Aerobots as shown in Fig. 11. These closely reflected the scenario explored as part of the simulation campaign. Aerobots were used in non-cooperative and cooperative modes.

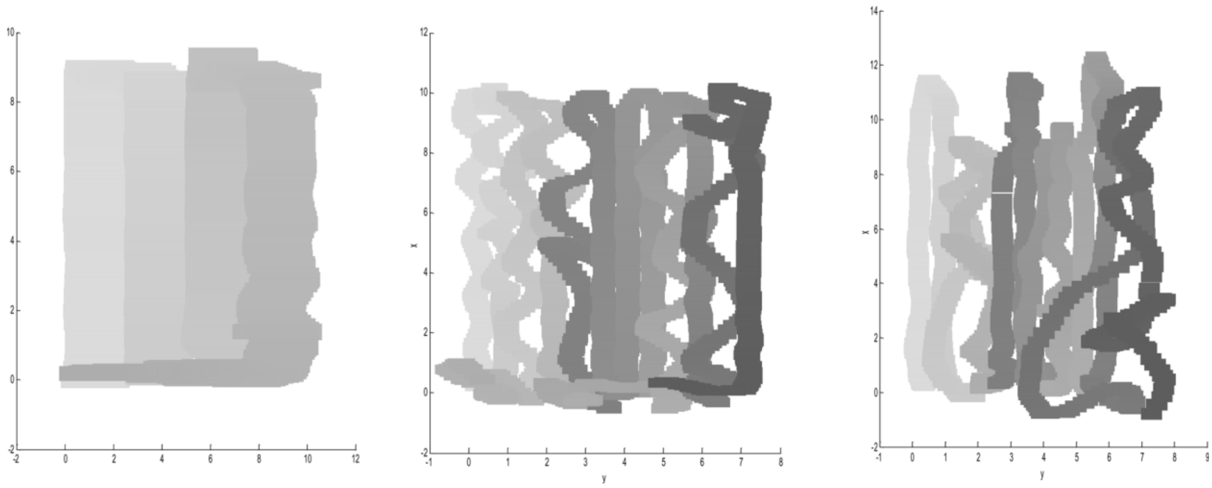

Fig. 12. (Left) Coverage of a single Aerobot with swath of 2.8m and line spacing of 2.5m. (Middle) Coverage of non-cooperative UUVs with swath width of 0.625 m. (Right) Coverage of cooperative Aerobots with swath width of 0.625m.

An example of area coverage of a single Aerobot is shown in Fig. 12 (left). The Aerobots' environment might have been very noisy especially when three of them were operated at the same time. In addition to normal sources of noise such as people movement, temperature variation, airflow, helium escaping and others, they were also affected by each other (fan air blowing through thrusters). Figure 12 (middle) shows the coverage of three Aerobots in a noncooperative mode while Fig. 12 (right) shows the results of cooperative mode. The coverage in cooperative mode is more organised as the swarm attempts to stay in a formation. If the placement of the formation is disrupted by a member, other members try to compensate for this by changing their own position. Hence, traces may show a large change, though the formation is held intact. In cooperative mode, communication among Aerobots was based on MTS which is SciSys's real-time in-house communication software infrastructure. Three blimps stayed in formation while performing a raster scan. The results are shown in Fig. 13. The formation flying results showed that the Aerobots could maintain a valid formation (in this case set as a line) and hold it despite the presence of noise in the environment. The algorithms used in simulation were directly ported to Aerobots. Tests were carried out in an environment that contained airflow caused as a result of air-conditioning units. The Aerobots managed to get back to the desired formation even when strong currents dispersed them. This was a good analogy to the underwater currents and showed that the simulation approach had been appropriate.

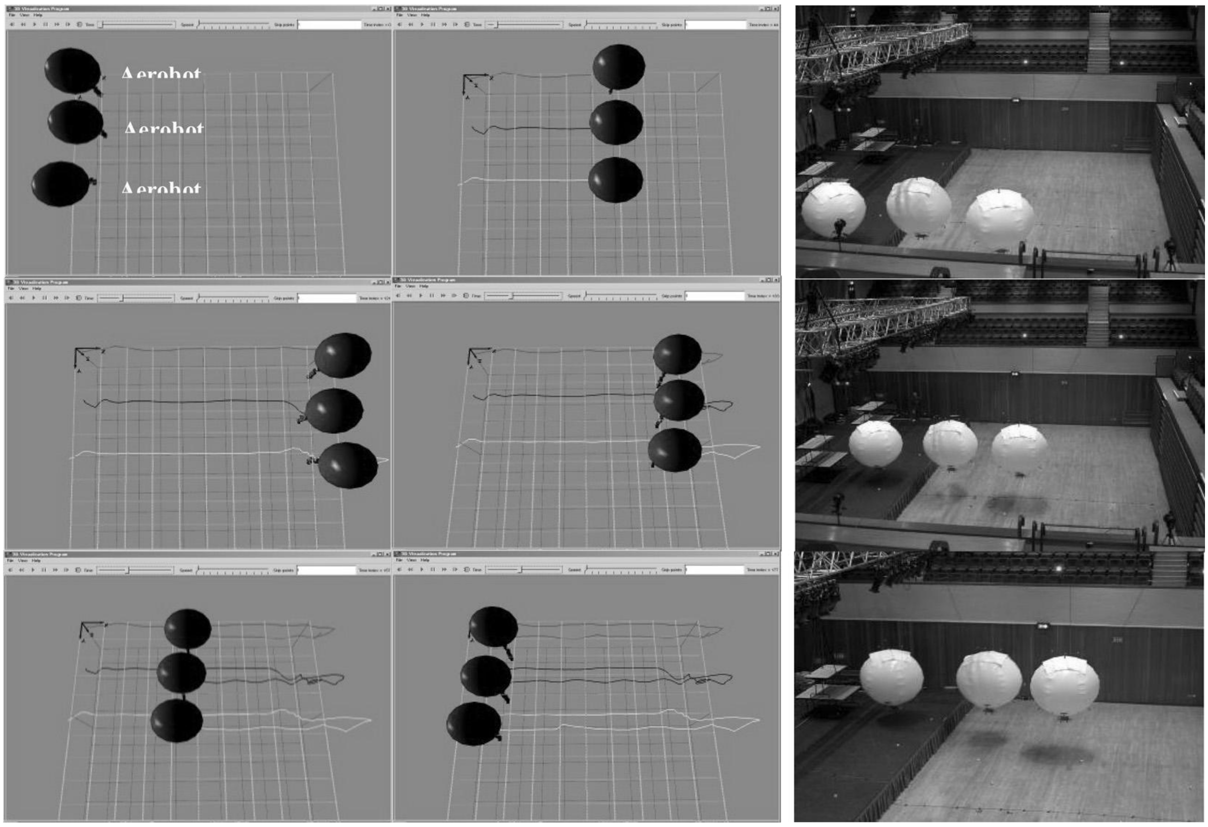

Fig. 13. (Left) Trajectory of cooperative Aerobots performing formation flying. (Right) Aerobots in action.

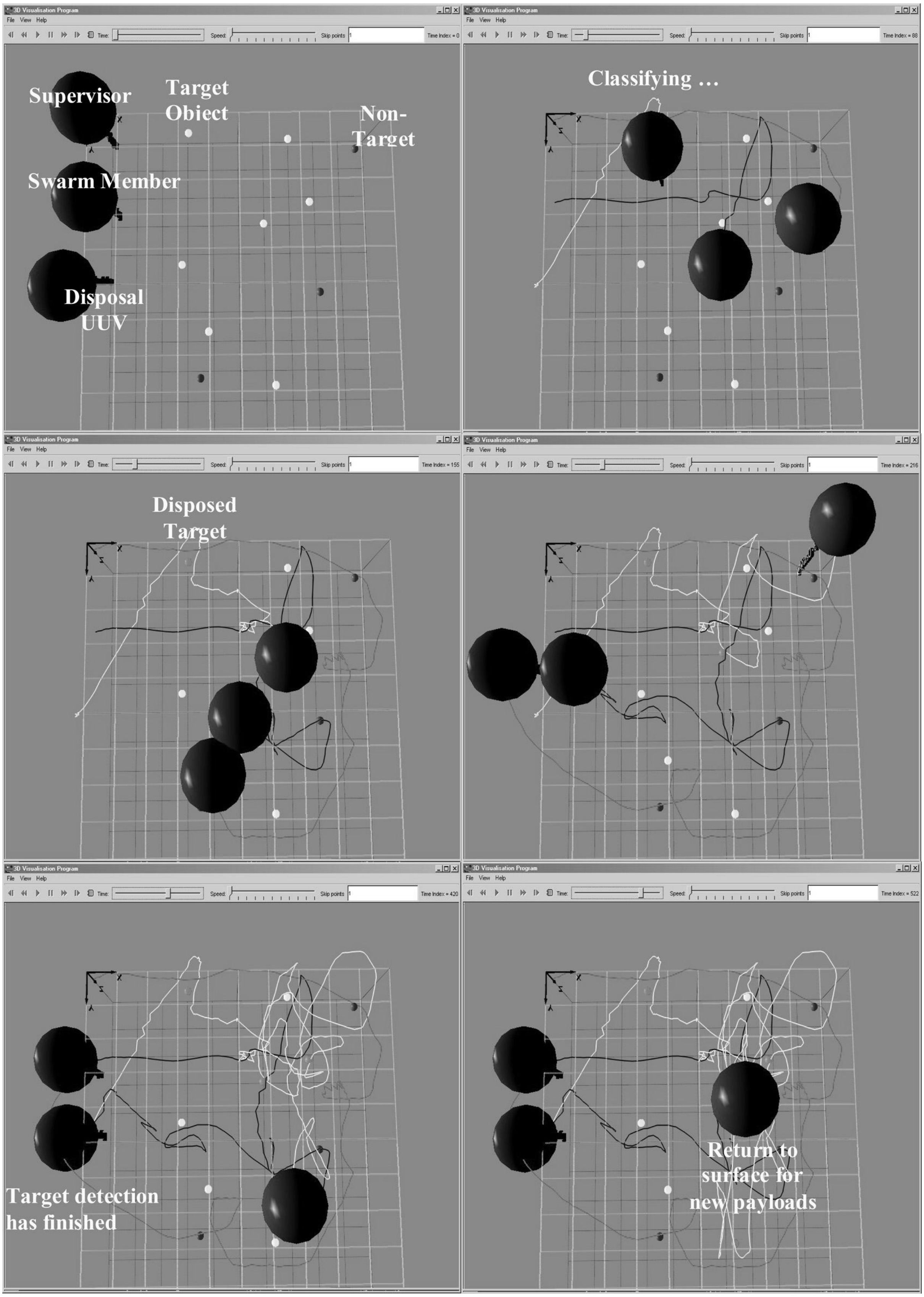

Fig. 14. Trajectory of the heterogeneous swarm that detects, classifies and disposes of objects.

Figure 14 shows the results of Aerobot demonstration for a disposal scenario. This time, a number of objects were placed in the test environment. Two detection Aerobots (Type I) were used to scan the environment for objects. One of them acted as the supervisor and another was a member of the detection team. They scanned the environment while performing formation flying. The third Aerobot was a Type III robot that was tasked to classify and dispose the detected objects. It moved towards the potential targets and started the process of classification and disposal. If it ran out of ammunition, it moved towards a central location and rose to surface to get new weapons. The focus of this experiment was the overall tasking of the group in real time and their ability to carry out the mission collectively.

15. Results

The BAUUV MuOps study explored the issues and strategies relating to the use of multiple autonomous underwater vehicles and their ability to carry out missions to achieve a desired objective. The primary question of the study was to determine if multiple vehicles would improve the mission efficiency. A number of operational concepts were considered to determine if they would benefit from UUV technology. The current state of the art in collective robotics suggests that there are many techniques developed that are yet to be exploited for the real-world missions. Behavioural and collaborative algorithms, where a number of agents could cooperate and attempt to carry out tasks together, have matured in recent years. This study, through the use of Aerobots, successfully raised the technology readiness level to TRL 4.

Given the operational concepts and the state of the field, a trade-off analysis selected an example, search, classify and disposal mission. A simulation of different strategies showed that for the example scenario:

- UUVs can benefit from formation flying algorithms to exploit sensor geometry in particular configurations, i.e., bringing multiple sensors to bear on the same target.

- The communication requirements to change a formation from one shape to another are insignificant. This can be exploited in a situation such as a loss of a UUV.

- There is a trade-off between the use of homogeneous and heterogeneous swarms as they both have their advantages and disadvantages. A heterogeneous swarm can be more cost-effective by avoiding the use of expensive units, while a homogeneous swarm may be easier to maintain and support.

- The relationship between the number of UUVs required to search an area versus the transit distance is not linear and may be optimised based on mission requirements.

- The successful demonstration of simulation results using the Aerobots highlighted that the technology was valid and that the system performed as expected given a realistic and noisy environment.

- The Aerobots could autonomously recover to a new formation given large disturbances.

- Inter-robot communications could benefit the coordination of tasks. A supervisor could command different members of the swarm for different tasks to autonomously optimise the use of resources in real time.

16. Conclusions

The use of Aerobots for robotics experiment and validation has proved to be an attractive solution. They have the necessary hardware infrastructure as well as expected limitations found on real system. Data-handling and communication issues can be thoroughly explored using these kinds of platforms. This naturally necessitates better software design and makes Aerobots a strong *test platform* that can be used for validation purposes and prototyping.

In addition, the simulation campaign showed that the use of cooperative robots in challenging environments is beneficial. In the field of robotics, recently there has been much interest in swarming technologies. As explored in this study, the underwater search missions can exploit the nature of cooperation between autonomous agents effectively. We conclude that the results of this study are very positive for the use of multiple vehicles in carrying out complex missions.

Reference

1. B. B. Werger, “Cooperation without deliberation: A minimal behavior-based approach to multi-robot teams,” *Art. Intell.* **110**, 293–320 (1999).
2. C. R. Melhuish, “Strategies for collective minimalist mobile robots,” Ph.D. Thesis (Bristol, UK: University of West of England, 1999).
3. A. J. Healey, “Application of formation control for multivehicle robotic minesweeping,” *Proceedings of the IEEE CDC Conference*, Paper No. CDC01-INV3103 (2003).
4. E. Honary, “Flock distortion: A collective approach to 3D trajectory mapping” Ph.D. Thesis (Bristol, UK: University of West of England, 2004).
5. E. Bonabeau, M. Dorigo and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems* (Oxford: Oxford University Press, 1999).
6. I. Rekleitis, V. Lee-Shue, A. P. New and H. Choset, “Limited communication, multi-robot team based coverage,” *Proceedings of the 2004 Conference on Robotics & Automation*, New Orleans, LA (Apr. 2004).
7. W. Sheng, Q. Yang, J. Tan and N. Xi, “Risk and efficiency: A distributed bidding algorithm for multi-robot coordination,” *5th World Congress on Intelligent Control and Automation*, Hangzhou, China (2004).
8. I. M. Rekleitis, G. Dudek and E. Milios, “Multi-robot exploration of an unknown environment, efficiently reducing the odometry error,” *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 2, Nagoya, Japan (1997).
9. J. Tan and N. Xi, “Peer-to-peer model for the area coverage and cooperative control of mobile sensor networks,” *Proceedings of SPIE Symposium on Defense and Security*, Orlando, FL, USA (2004).
10. J. Fredslund and M. Mataric, “A general algorithm for robot formations using local sensing and minimal communication,” *IEEE Trans. Robot. Autom.* **18**(5), 837–846, (Oct. 2002).
11. J. Wessnitzer, “Strategies for structural self-organisation in wireless networks of mobile robots,” Ph.D. Thesis (Bristol, UK: University of the West of England, 2004).
12. T. Balch and R. C. Arkin, “Behaviour-based formation control formulti-robot teams,” *IEEE Trans. Robot. Autom.* **14**(6), 926– 939 (1998).
13. E. Fiorelli, N. E. Leonard, P. Bhatta, D. Paley, R. Bachmayer and D. M. Fratantoni, “Multi-AUV control and adaptive sampling inMonterey Bay”, *Proceedings of IEEE Autonomous Underwater Vehicles 2004: Workshop on Multiple AUV Operations (AUV2004)*, Sebasco, ME (June 2004).
14. C. W. Reynolds, “Flocks, herds, and schools: A distributed behavioral model,” *Comput. Graph.* (*SIGGRAPH ’87 Conference Proceedings*) **21**(4), 25–34 (1987).
15. J. Rosenblatt, “DAMN: A Distributed Architecture for Mobile Navigation,” *AAAI Spring Symposium on Software Architectures for Physical Agents*, Stanford, CA, (1995).
16. D. Barnes, A. Shaw, P. Summers, R. Ward, M. Woods, M. Evans, G. Paar and M. Sims, “Imaging and localisation software demonstrator for Planetary Aerobots,” *Acta Astronautica*, **59**(8–11), 1062–1070 (October to December 2006).