Adaptive mission plan diagnosis and repair for fault recovery in autonomous underwater vehicles

Pedro Patrón, Emilio Migueláñez, Yvan R. Petillot, David M. Lane, Joaquim Salvi
Ocean Systems Laboratory, Heriot-Watt University
Edinburgh, Scotland, UK, EH14 4AS
Email: \{p.patron, e.miguelanez, y.r.petillot, d.m.lane, j.salvi\}@hw.ac.uk
Telephone: +44 (0) 131 451 3320
Fax: +44 (0) 131 451 4155

Abstract—This paper proposes a novel approach for autonomous mission diagnosis and repair for maintaining operability of unmanned underwater vehicles. It combines the benefits of knowledge-based ontology representation, autonomous partial ordering plan repair and robust mission execution. The approach uses the potential of ontology reasoning in order to orient the planning algorithms adapting the mission plan of the vehicle. It can handle uncertainty and action scheduling in order to maximize mission efficiency and minimize mission failures due to external or unexpected factors. Its performance is presented in a set of simulated scenarios. The paper concludes by showing the results of a trial demonstration. Observations of different environmental and internal parameters are simulated in a REMUS 100 AUV while performing a mission. These trigger a knowledge exchange between the diagnosis monitor agent and the adaptive mission planner embedded agent. Based on the observed data and the original knowledge, the experiment shows how the adaptive planner system is able to identify the gaps in the mission and adapt the platform’s mission plan accordingly.

I. INTRODUCTION

A. Motivation

Autonomous Underwater Vehicles (AUVs) have become a standard tool for data gathering for Maritime applications. In these environments, mission effectiveness directly depends on vehicle’s operability. Operability underlies the vehicle’s final availability, affordability and acceptance. Two main vehicle characteristics can improve the vehicle’s operability: reliability relates to vehicle failures due to the internal hardware components of the vehicle, and survivability relates to vehicle failures due to external factors or damages (see Fig. 1).

In recent years, emphasis for increasing (↑) AUV’s operability has been focused in increasing AUV’s survivability by reducing (↓) the susceptibility and vulnerability of the platform [1]. Recent approaches in rules of collision [2] and wave propagation techniques [3] for obstacle avoidance and escape scenarios [4] have focused on reducing susceptibility by looking at the adaptation of the vehicle’s trajectory plan.

However, when active and passive measures fail to protect the vehicle, or unexpected hardware failures occur, the focus of the mission should shift to ‘reconfigure’ itself to use alternative combinations of the remaining resources. The underwater domain has scarce bandwidth and tight response constraints to keep the operator in the loop. In such challenging environment, autonomous embedded recoverability, is a key capability for vehicle’s endurance. This can be achieved via adaptation of the vehicle’s mission plan.

Current AUV mission plan solutions are procedural and static [5]. If behaviors are added [6], they are only to cope with possible changes that are known a-priori by the operator. In order to achieve higher levels of autonomy, an evolution in plan adaptability from current waypoint-based approaches on trajectory planning to a declarative goal-based solution for adaptive mission planning is required.

Declarative mission planning for unmanned vehicles leverage from the research carried out on autonomous planning generation [7]. However, almost as important as the mission plan generation (if no more), is the capability of the mission to adapt and recover from failures and external changes. The aim is to be effective and efficient and a mission plan costs time to prepare. This time has been already invested once (to compute the mission that is now failing), so it might be more efficient to try to reuse previous efforts by repairing the mission. Also, commitments might have been made to the current mission plan: trajectory reported to other intelligent agents, assignment of resources or assignment of part of mission plan to executors, etc. Repairing an existing mission ensures that as few commitments as possible are invalidated. Finally, several planners (usually autonomous and human planners combined) could be performing together to achieve the goals. In such cases, it is more likely that a similar mission plan will be accepted by the operator than one that is potentially completely different.

B. Related work

Space operations, and Mars rovers in particular, have been the impetus behind autonomous planning and adaptability for
unmanned vehicles. In these environment, the main motivation for mission plan adaptation is to reduce vehicle idle times and the handling of opportunistic events in order to maximize science return. One of the first systems that handle a certain level of planning adaptation in a real scenario was the Automated Schedule and Planning Environment (ASpen) [8]. This system classified constraint violations and attempted to repair each by performing a planning operation. However, the set of repair methods were predefined and the type of conflict determined the set of possible repair methods for any given conflict. The system could not guaranteed that it would explore all possible combinations of plan modifications or that it would not retry unhelpful modifications. The Remote Agent Experiment (RAX) [9], which flew on Deep Space-1, demonstrated its ability to autonomously control an active spacecraft. This project highlighted the problem of using a batch planner on a reactive system. It took hours to replan after updates had invalidated the original plan.

Another system, called CLEaR [10], used the CASPER system [11] in conjunction with a sequencer/controller in order to provide a planning and execution framework for closed loop commanding. The deliberative and reactive methods operated in parallel at run time to determine how to best respond to failures and take advantage of opportunities. The system was used by [12] and [13] who highlighted the difference between 'local' conflicts - errors that require changes only to the currently executing part of the plan - and 'global' conflicts - errors that occur which require changes to future parts of the plan. Local conflicts were managed by the executive while global ones were passed back to the planner to be fixed. Van der Krogt later formalised this two levels of handling events in what was called executive repair and planning repair [14]. His approach combined unrefinement and refinement stages in order to provide faster performance than planning from scratch. However, it failed to produce the most optimal plan. This could be considered an issue in domains requiring optimality. This is not generally the case in unmanned vehicle mission plans where optimality can be sacrificed for operability. The approach was compared with the GPG and the Sherpa systems. In the GPG [15], based on the graphplan planner [15], the unrefinement stage is done only on the initial plan and never on any of the plans produced by a refinement step. In the GPG [16], based on the graphplan planner, the unrefinement stage is done only on the initial plan and never on any of the plans produced by a refinement step. The Sherpa [17], based on the LPA* algorithm used in [3] for adaptive trajectory planning, could only be applied to problems in which actions have been removed from the domain description.

Recently, Fox proposed an on-board planning assistant to the operator to adjust and repair plans on-board [18]. The system was developed to handle idle times due to conservative mission planning and plan failures on the Beagle 2 Mars spacecraft. The approach relaxes methodological constraints and fills opportunity gaps only in situations where resources would otherwise go unused. A wider discussion on other related research that goes from strong executors and formal planning approaches to strong deliberators can be found in Knight’s review of the field [19].

In the underwater domain, several challenges have been identified requiring adaptive planning solutions of the mission [20] [21]. The potential benefits of the adaptive planning capabilities have been promoted by Rajan [22]. Together with Fox and other authors, they have started using sensor data to adapt the decision making on the vehicle’s mission [23], [24].

C. Contribution

In this paper, we present an approach for adaptive mission planning for AUVs. It uses the potential of semantic knowledge representation and diagnosis monitoring in order to respond to feedback in the platform capabilities and orient the decision making process. At the decision level, partial order planning provides solutions to mission actions that take the environment and the internal status of the vehicle into account while maintaining the AUV operative. The main contribution of this paper is to increase platform’s operability and to maintain the originally established commitments by re-using of previous planning effort.

The paper is structured as follows: Section II describes our evolved view of the sensing-decision-acting loop for adaptive mission planning for AUVs. Section III focuses on the semantic representation of the vehicle’s knowledge and the detection of possible failures. Section IV concerns the reasoning for knowledge consistency and mission planning algorithms for mission repair. The transition from mission planning to execution in a real platform is described in Section V. Section VI describes the experimental results in simulation and in a real scenario. This paper ends in Section VII with the conclusions and future work.

II. DECISION MAKING LOOP

The human capability of dealing with and understanding highly dynamic and complex environments is known as situation awareness ($SA_H$). $SA_H$ breaks down into perception of the environment, comprehension of the situation and projection of the future status. According to Boyd, decision making occurs in a cycle of observe-orient-decide-act (OODA) [25]. The Observation ($O_{oba}$) component corresponds to the perception level of $SA_H$. The Orientation ($O_{oda}$) component contains the previously acquired knowledge and understanding of the situation. The Decision ($D_{oda}$) component represents the $SA_H$ levels of comprehension and projection. This last stage is the central mechanism enabling adaptation before closing the loop with the final Action ($A_{oda}$) stage. Note that
it is possible to take decisions by looking only at orientation inputs without making any use of observations.

Based on the autonomy levels and environmental characteristics, $S_A$ definitions can be directly applied to the notion of unmanned vehicle situation awareness ($S_A$V) [26]. The levels of situation awareness for individual unmanned vehicle systems ($S_A$) range from full human control to fully autonomous unmanned capabilities (see Fig. 2).

In current implementations, the human operator constitutes the $D_{OODA}$ phase. The embedded architecture is formed by the world model ($M_{WM}$) and the mission plan executive ($M_{ME}$) modules. $M_{WM}$ stores $\mathcal{OB}_{ooda}$ and $\mathcal{OR}_{ooda}$ and $M_{ME}$ performs $\mathcal{OA}_{ooda}$. When high-bandwidth communication links exist, the operator remains in the $OODA$ loop during the mission execution (see Fig. 3). Examples of the implementation of this architecture are existing Remote Operated Underwater Vehicles (ROVs). However, when the communication is poor, unreliable or not available, the operator tries, based only on $\mathcal{OR}_{ooda}$, to include all possible behaviours to cope with execution alternatives. This has unpredictable consequences, in which unexpected situations can cause the mission to abort and might even cause the loss of the vehicle. Examples of this architecture are current implementations for AUVs (see Fig. 4).

In order to achieve autonomous adaptive mission planning, two additional components are required: a status monitor ($M_{SM}$) and a mission plan adapter ($M_{MP}$). The status monitor reports any changes detected during the execution of a plan. These modifications might change $S_A$V perception. When the $M_{ME}$ is unable to handle the changes coming from the $M_{SM}$, the $M_{MP}$ is called to generate a new modified mission plan that agrees with the updated constraints (see Fig. 5). The RAX project was the first attempt of implementing this type of architecture on a real environment. However, tight time deadlines, more restricted communications and environment constraints existing in general AUVs applications have led to the research of new approaches.

III. ORIENTATION AND OBSERVATION

$S_A$V consists in making the vehicle to autonomously understand the 'big picture'. This picture is composed of the experience achieved from previous missions and the information obtained from the sensors while on mission.

A set of ontologies is developed in order to represent the knowledge information required in $S_A$V. $TBox$ and $ABox$ statements make up an ontology. These components contain respectively the terminological and assertion concepts and relationships for each particular context. A direct correspondence of $\mathcal{OR}_{ooda}$ with $TBox$ and $\mathcal{OB}_{ooda}$ with $ABox$ can be established.

Two ontology types for $S_A$V are defined (see Fig. 6):

- The Application ontology represents the particular vision of $S_A$V for each agent or module involved in the decision making process. This knowledge is only related to the agent reasoning process and therefore domain independent.
- The Core ontology represents the domain-dependent knowledge. In this study case, this corresponds to the underwater environment. The Core ontology extends from other well established Upper ontologies such as the Suggested Upper Merged Ontology (SUMO). It also makes use of Utility ontologies such as the Ontology for Geography Markup Language of Open GIS Consortium (GML-OGC). The Core ontology enables the transfer of knowledge between agents.

An Adapter is capable of parsing raw data from sensors into $ABox$ assertions of the Core ontology during the mission. As discussed, embedded $D_{ooda}$ and $A_{ooda}$ for adaptive mission planning occurs at the liaison between the $M_{MP}$ and the $M_{ME}$ modules. This section describes the main concepts handled by these agents at the Core and Application levels, and briefly describes the $M_{SM}$, the other decision making agent involved in the mission adaptation process.

A. Core ontology

Domain concepts are mainly handled by agents to interchange concepts between them and by the $M_{ME}$ to interact with the functional layer. They are used to ground the abstract planning concepts managed by $M_{MP}$ to a particular domain. This allows transition from the $D_{ooda}$ phase to the $A_{ooda}$ phase of the $OODA$ loop. Some of the concepts identified in the autonomous underwater vehicle domain are: Platform (static or mobile - ground, air, underwater), payload (hardware with particular properties), module (software with specific capabilities), sensor (a device that receives and responds
to a signal or stimulus), waypoint (position in space with coordinate and tolerance), coordinate, velocity,...

B. Planning ontology

The planning ontology is the Application ontology containing the planning concepts handled by $M_{MP}$. In order to provide a solution to a mission failure, the decision making requires, not only concepts to generate a mission, but also concepts capable of representing incidents or problems occurring during the mission. Some of the most important concepts identified for mission plan adaptability are: Resource (state of an object in the environment - physical or abstract), action (resource modifier), catalyst resource (resources that are not consumed for an action but needed for the proper execution of the action), plan gap (actions that may no longer be applicable), execution (successful action execution), failure (an unsuccessful execution), incident (combination of failures)....

C. Diagnosis system

The aim of the ontological diagnostic model is to model the behaviour AUV internal systems, in order to monitor the health of the vehicle and to report any critical or incipient status. To model the behaviour of all components and subsystem considering form sensor data to possible model outputs, the diagnostic domain ontology is designed and built based on ontology design patterns [27]. Ontology patterns facilitate the construction of the ontology, promotes the main goal of reusing, coordinate and tolerance), engine, motor, etc. These concepts are based on the system observation design pattern described in [28].

Two important concepts in the Core ontology are the concepts of Critical status and Incipient status. They describe the status of the hardware components that are malfunctioning or not performing normally respectively. Instances from these concepts will be generated by the $M_{SM}$ to communicate the availability of hardware components to the $M_{MP}$ during the adaptation process.

IV. Decision

A. Mission plan adaptation

Following classical planning problem representation, an instance of a vehicle mission problem can be simply defined as $\Pi = \{P, A, I, G\}$, where $P$ is the set propositions defining the available resources in the vehicle, $A$ is the set of actions, $I$ is the initial platform state and $G$ is the set of possible mission accomplished states. $D = \{P \cup A\}$ defines the mission domain and $P = \{I \cup G\}$ the mission problem. Given an instance $\Pi$, the mission generation problem consists in finding if there exists a mission plan ($mp$), using $a \in A$, such that satisfies any $g \in G$.

Several approaches exist in the artificial intelligent (AI) literature capable of solving this problem. In a real environment where optimality can be sacrificed by operability, partial ordered planning ($pp$) is seen as a suitable approach because it produces a flexible structure capable of being adapted (see Fig. 7). The implemented approach can deal with extensions from the classical representation. It can handle durative actions, fluents, functions and different search metrics in order to minimise resource consumption, such as remaining vehicle’s battery, or total distance travelled.

Fig. 8 shows the benefits between replanning and repairing a mission plan. At the initial loop, a partial ordered plan $pp_0$ is generated satisfying the given mission domain and problem

![Diagram](image-url)

**Fig. 6.** $SA_V$ concept representation (Core, Application ontologies), instance generation (adapter) and handling (reasoning, inference and decision making agent).

**Fig. 7.** Ordering constraints (grey-arrows) and interval preservation constraints (black arrows) between actions in a partial ordered plan representation of an autonomously generated AUV mission.

**Fig. 8.** Replan and repair processes for mission plan adaptation.
Π₀. The pp₀ is then grounded into the minimal mission plan m₀p₀ including all constraints in pp₀. At an iteration L, the knowledge base is updated by the diagnosis information dₐ providing a modified mission domain and problem Π₀'Lₐ₊₁. From here, two mission recovery options are possible: The mission replan process generates a new partial plan pp'Lₐ₊₁, as done at the first stage, based only in the Π₀'Lₐ₊₁ information. On the other hand, the mission plan repair process re-validates the original plan by ensuring minimal perturbation of it. Given a mp to Πₗ, the mission repair problem produces a solution mission plan mp' that solves the updated mission problem Π₀'Lₐ₊₁, by minimally modifying mp.

When a mission failure occurs during the execution, two possible repair levels can be identified: mission execution repair and mission plan repair. Execution repair changes the instantiation mp such that either: an action aᵢ that was previously instantiated by some execution α is no longer instantiated, or an action aᵢ that was previously instantiated is newly bound by an execution α already part of the mp. Plan repair modifies the partial plan pp itself, so that it uses a different composition, though it still uses some of the same constraints between actions. It might also entail the elimination of actions which have already been instantiated.

Executive repair will be less expensive and it is expected to be executed by Mₘₑ. Plan repair, however, will be computationally more expensive and requires action of Mₘₚ. The objective is to maximise the number execution repairs over plan repairs and, at the plan repair level, maximise the number of decisions reused from the old mission.

B. Mission plan refinement

A practical approach following the previously described concepts has raised interest recently in the AI community providing a novel solution for all drawbacks identified during the replanning process. This set of methods is known as plan recovery methods. Plan recovery methods are based on plan-space searches and are able to adapt the existent plan to the new state of the world. They can be divided into two stages:

The first stage, known as plan diagnosis, analyzes the effects of the updated platform status on the current mission. According with the new updated constraints received from the Mₛₘ, it identifies the failures and gaps existent in the current mission plan. These plan gaps are causing the inconsistency between the existent plan and the current status of the platform and the environment. They are, therefore, preventing the correct execution of the mission. The approach developed at this stage is based on unrefinement planning strategies and uses the potential of the ontology reasoning previously described in order to identify the resources that remain available.
The second stage is known as plan repair. The strategy during this stage is to repair with new partial plans the gaps or failures identified during the plan diagnosis stage. The plan repair stage is based on refinement planning strategies for plan recovery.

In simple terms, when changes on the ABox planning ontology are sensed (d) that affect the consistency of the current mission plan ppL, the plan adaptability process is initiated. Based on the outputs of the planning ontology reasoner, the plan diagnosis stage starts an unrefinement process that relaxes the constraints in the mission plan that are causing the mission plan to fail. The remaining temporal mission partial plan ppL is now relaxed to be able to cope with the new sensed constraints. This will be the simplest stage of recovery necessary to continue with the execution of the plan but it does not guarantee that all the mission goals will be achieved. The plan repair stage then executes a refinement process searching for a new mission plan ppL+1 that is consistent with the new world status D′ and P′. By doing this, it can be seen that the new mission plan mpL′ is not generated again from D′ and P′ (re-planned) but recycled from ppL (repaired). This allows re-use of the parts of the plan ppL that were still consistent with D′ and P′.

C. Application results

A set of synthetic simulated scenarios have been implemented in order to test the performance of the mission planner module (MMP). The tests are based on the mine counter measure (MCM) operation scenario using AU.Vs. In this scenario, AU.Vs support and provide solutions for mine-hunting and neutralisation. The operation involves high levels of uncertainty and risk of damage in the vehicle. Navigating in such a hazard environment is likely to compromise the vulnerability of the platform. Additionally, if a vehicle is damaged or some of its components fail, mission adaption will be required to cope with the new restricted capabilities.

A set of 15 selected MCM scenarios where simulated. For each scenario, the detection of a failure dL was triggered. The mission adaptation was performed using classic replanning methods and the mission plan repair approach.

Performance was compared by looking at different metrics. Fig. 9 shows the computation time required for adapting the mission to the new constraints. Fig. 10 displays the distance δ(ΠL, ΠL′) between the original plan ΠL and the adapted plan ΠL′ [29]. Fig. 11 shows the number of instances explored during the search of the new adapted plan. Fig. 12 represents the mission plan quality based on the number of goals achieved. It can be seen that, in general, mission repair compared with replanning improves performance and time response while maintaining mission quality (number of goals achieved) and reducing distance to the original mission plan.

V. Architecture

As described in Section II, the system implements the four stages of the OODA loop (see Fig. 13). The status monitor MSM reports changes in the environment and the internal status of the platform to the MME and MWIM modules. The world model MWIM stores the ontology-based knowledge provided by the a-priori expert orientation and the observation of events received from the status monitor. A mission planner module MMP generates and adapts mission plans based on the notifications and capabilities reported by the MME and MWIM. The mission executive module MME is in charge of executing the mission commands in the functional layer based on the actions received from MMP. In order to provide independency of the architecture with the vehicle’s functional layer, an Abstract Layer Interface (ALI) has been developed.

The translation from the MMP to the MME is done via a sequence of instances of action executions. An action execution α is described by the domain ontology TBox and gets instantiated by the action grounded on mp. The instance contains the script of commands required to perform the action (see Fig. 14). The action execution contains a timer, an execution counter, a timeout register and a register of the maximum number of executions. The success, failure or timeout outputs control the robust execution of the mission and the executive repair process.

Once mp is obtained and the list of αi is generated for mp, the mission plan gets transformed into a state machine of action execution instances. A action execution graph gets generated that contains all the possible options for the new plan. This deals with the robustness of the execution and the execution repair process. This minimises the number of calls to the MMP and therefore the response time for adaptation.
The performance of the system has been evaluated on a REMUS 100 AUV platform (see Fig. 15). A PC/104 1.4GHz payload running Linux has been installed in the vehicle. The system is capable of communicating with the vehicle’s control module via the manufacturer’s Remote Control protocol.

Fig. 16 shows the original mission plan described by the operator. The mission plan consists of a simple go-return pattern at an approximate constant latitude. Each of the two mission legs is approximate 500 meters. This original mission plan is transferred to the vehicle.

On the other hand, the adaptive system receives information about the area of operation and a list of goals. At this point, the seabed areas are provided based on automatic computer aided classification knowledge generated from previous existent data. These areas are shown in Fig. 17.

The planner generates a new mission plan based on the information provided and the list of goals received. In this case, the mission goals can be resumed as ‘survey all known areas’. This mission is passed to the executive that takes control of the vehicle for its execution.

While the mission is being executed the status monitor maintains an update of the knowledge stored in the world model. In this example, the status monitor reports status of hardware components, such as batteries and sensors and external parameters such as water currents. At this point, the incomes to the status monitor have all been modelled and simulated in order to be able to interact with its outcomes. The status of this elements over time during mission are displayed on Fig. 20, Fig. 21 and Fig. 18.

When the executive is about to execute the next action in the mission plan, it looks at the knowledge stored on the world model in order to generate the adequate list of commands to the vehicle. It can been seen that the list of waypoints generated for the lawnmower pattern survey of the areas is related to the measured water current at the moment of initiating the each of the surveys.

If a parameter in the world model changes affecting the execution of the current mission, the system generates a new set of commands to cope with the change. In this scenario, a change in the lawnmower pattern can be observed when one of the transducers of the sidescan sonar is reported as faulty. This adaptation allows maximizing sensor coverage for the survey while the transducer is down. It can be seen how the pattern change back to normal once the transducer is reported as fixed.

The symbols ♦, ○, ▽, and □ are use to describe the points where the events have occurred. ♦ represents the point when the first survey starts. At that point, the water current heading was 19.827 degrees (see Fig. 21). This is the heading taken by the lawnmower pattern in order to minimize drag and maximize efficiency. ○ represents the point when the diagnosis system
We implement a system capable of repairing and adapting the mission plan in real-time. The approach is capable of adapting to the critical status of certain components in mid-mission. It maximises the system performance and response time. The decision making process for mission adaption uses a combination of refined and unrefined phases of the constraints of a partial ordered mission plan. The system performance has been demonstrated in simulation. Additionally, the mission adaptation capability is shown during an in-water field trial demonstration.

In our fully integrated experiments we achieve the following:

- **Goal oriented planning vs waypoint based planning**: The system uses a goal oriented approach in which the mission is described in terms of 'what to do' instead of 'how to do' it.
- **Adaptation to environmental parameters**: The system shows adaptability to environmental elements, such as water current flows in order to improve mission performance.
- **Adaptation to internal issues**: The approach is capable of dealing with the critical status of certain components in mid-mission.

VII. Conclusion

The underwater domain is a challenging environment in which to maintain AUV’s operability. Operability can be improved with the embedded adaption of the mission plan.

We implement a system capable of repairing and adapting mission plans autonomously while during a mission. We do this by using combination of ontological representation of knowledge, system diagnosis and adaptive mission plan repair techniques. The advantage of this approach is that it maximises robustness, system performance and response time. The decision making process for mission adaption uses a combination of refinement and unrefinement phases of the constraints of a partial ordered mission plan. The system performance has been demonstrated in simulation. Additionally, the mission adaptation capability is shown during an in-water field trial demonstration.

In our fully integrated experiments we achieve the following:

- **Goal oriented planning vs waypoint based planning**: The system uses a goal oriented approach in which the mission is described in terms of ‘what to do’ instead of ‘how to do’ it;
- **Adaptation to environmental parameters**: The system shows adaptability to environmental elements, such as water current flows in order to improve mission performance.
- **Adaptation to internal issues**: The approach is capable of dealing with the critical status of certain components in mid-mission.
the platform and can react accordingly.

The proposed architecture is open to other sources of observation feeding the mission adaptation process. We are planning to extend the single vehicle mission plan recovery to a mission plan recovery for a group or team of vehicles performing a collaborative mission. We are currently working for an SA for a team of vehicles (SA\textsubscript{T}) to which every team member possesses the SA\textsubscript{S} required for its responsibilities. The main challenge for this will be to deal with the acoustic communication limitations associated to the underwater environment.

ACKNOWLEDGMENT

Our thanks to the members of the Ocean Systems Laboratory for their inputs and helpful critique. Thanks also to all at SeeByte Ltd, Edinburgh for providing the necessary AUV mobilisation, practical trials infrastructure and knowledge to run the experiments in the real world.

The work reported in this paper is partly funded by the Project SEAS-DTC-AA-012 from the Systems Engineering for Autonomous Systems Defence Technology Centre and by the Project RT/COM/5/059 from the Competition of Ideas, both established by the UK Ministry of Defence.

REFERENCES