

# Enabling Fault Recovery and Adaptation in Mine-Countermeasures Missions Using Ontologies

Georgios Papadimitriou, Zeyn Saigol, David M. Lane  
Ocean Systems Laboratory  
School of Engineering & Physical Sciences  
Heriot-Watt University, EH14 4AS, Edinburgh, UK  
Email: gp125@hw.ac.uk

**Abstract**—The work presented in this paper focuses on addressing vehicle hardware faults and changes in high level priorities during Mine Countermeasures (MCM) missions with Autonomous Underwater Vehicles (AUVs). The approach utilizes ontologies for representing knowledge about the vehicle and its operational environment and is based on the KnowRob system. Reasoning on the vehicle capabilities and consequently the actions it can execute is continuous and occurs in real time. Hardware faults are incorporated into the reasoning process as a means of driving adaptive planning and execution. The system utilizes the OPTIC Planning Domain Definition Language (PDDL) planner. Adaptive execution is prioritized over adaptive planning as mission planning can be very demanding in terms of computational resources. Changes in high level mission priorities are also addressed as part of the adaptive planning behaviour of the system. The main contribution of this paper is an ontological approach that drives an adaptive behaviour for increasing persistent autonomy of AUVs in unexpected situations. That is when hardware faults threaten to put the mission at risk and changes in high level mission priorities should be incorporated as part of decision making.

## I. INTRODUCTION

There are many benefits to performing underwater operations with Autonomous Underwater Vehicles (AUVs). These among others include risk reduction to human personnel in the field of operations, improved effectiveness through quality of sensor data and enhanced endurance. Carrying out successful autonomous missions though and consequently obtaining the aforementioned benefits comes with its own set of significant challenges.

AUV missions commonly use pre-scripted plans, but these do not scale well with partially known or unknown environments. Due to their nature, pre-scripted plans have limited capability of exploiting observations that can be used as a basis for decision making. Unexpected changes in the world or vehicle state, sensor limitations, and hardware faults often affect mission performance at best. In addition, high level mission priority changes can occur in real time and vehicles should be able to accommodate and act upon them. Addressing the aforementioned challenges is a non-trivial task. Real time fault detection, online mission planning and knowledge acquisition approaches are necessary but hardly sufficient. To promote persistent autonomy and successful mission execution these

approaches should be backed by fault recovery capabilities, adaptation both on the mission planning and execution levels and efficient knowledge representation and reasoning.

In this paper we present an ontology-based system that utilises a combination of efficient knowledge representation, knowledge evolution and reasoning to drive adaptive mission planning, execution and fault recovery. Our system can be used by AUVs to: i) Reason about their capabilities and actions, reassess them in real time and recover in response to hardware faults, ii) Respond to changes in high level mission priorities during mission execution.

The system is tested in the simulated Mine Countermeasures (MCM) scenario that was presented in [1], where the goal is to survey an area for possible mines, then classify targets as Mine Like Objects (MLOs) with associated coordinates, probability of being mines and entropy. Finally, the goal is also to reacquire each MLO and inspect it. The MCM mission is broken down into two passes consisting of two phases each using a single AUV. Surveying an area for mines (detection) and classification are the two phases of the first pass while reacquisition and inspection are performed as part of the second pass. We assume the operator is able to communicate changes in mission priorities to the vehicle along the low-bandwidth acoustic channel, for example in response to a changing situation on the surface that the vehicle cannot monitor. The AUV is also equipped with multiple sensors, and the failure of one of these may allow a mission to continue using others, subject to replanning.

The rest of the paper is organised as follows. Section II provides a brief survey on related work in the fields of ontology based knowledge representation and reasoning, fault recovery and adaptive planning and execution. Section III provides intuition on our ontological knowledge representation and reasoning approach for fault recovery and adaptation. Section IV describes our adaptive planning and execution approach. In section V we present our experimental setup. Finally, we present our findings and conclusions in sections VI and VII respectively.

## II. RELATED WORK

Ontology based knowledge representation and reasoning, fault recovery as well as adaptive planning and execution

have been gathering a lot of attention for many years now especially with the increase of using autonomous vehicles for real world robotic applications. Utilizing ontologies in robotics builds on a mature prior body of work on developing ontology based knowledge representation and reasoning systems for the maritime industry, the semantic web, biology, aerospace etc. [1].

Highly relevant to our work is the KnowRob system [2], [3] upon which our system is based. KnowRob is an ontological knowledge representation, processing and reasoning system that was built for household robots but can be easily extended to other domains. In terms of encoding knowledge about the components, capabilities and executable actions on a robot our system extends KnowRob so that it encodes information about faulty components which is incorporated in the reasoning process. Another ontology based robotic framework is the ORO system [4]. ORO focuses on building cognition in robots as a means of promoting Human Robot Interaction (HRI) by combining prior domain knowledge, sensor data and human input. In [5], [6] the authors present an ontology based framework for AUVs that is capable of fault recovery by devising mission plan and execution adaptation where appropriate. Similarly to our approach the system prioritizes execution adaptation over mission planning adaptation in order to save computational resources. Additionally, the system implements the OODA (Observe-Orient-Decide-Act) architecture that was initially proposed in [7].

Outwith the ontology-based approaches one of the most established adaptive systems is T-Rex [8]. T-Rex comprises control agents (Teleo-Reactors) each of which follows a Sense-Plan-Act (SPA) approach to control. Each control agent “has a different functional and temporal scope” [9]. The functional scope refers to the goals the agent: i) understands, ii) can plan for, iii) monitor the progress of. The temporal scope on the other hand refers to how much time in the ‘future’ the agent has to generate a plan for. For planning, T-Rex utilizes the EUROPA2 temporal planning library [10].

### III. ONTOLOGY BASED KNOWLEDGE REPRESENTATION AND REASONING

The use of ontologies in robotics is increasing, as evidenced by the mature KnowRob system upon which our approach is based. Capturing relational knowledge with ontologies provides a well defined and powerful knowledge representation mechanism, which allows for sophisticated reasoning about every aspect of the environment and the AUV subsystems. Our framework uses ontologies encoded in OWL DL to provide a modular and flexible system which is easily extensible. As well as defining concepts and relations, the ontology supports a Knowledge Base (KB) for storing run time data. We build on our previous work with this framework [1] by adding capabilities reasoning (described in Section III-A) and mission goal adaptation (Sections III-B and IV).

#### A. Components, Capabilities, & Actions Ontologies

Our system encodes knowledge about the status of the AUV in three different, interconnected ontologies. These ontologies describe the components, the capabilities and the actions that are available to the AUV. Modelling is performed in such a way that the AUV is aware of the dependency relations between them. The approach followed is bottom-up, that is, the status of the components determines the availability of capabilities which successively determine the actions that the AUV can execute (see Fig. 1). More specifically, the components

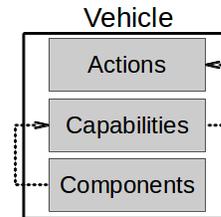


Fig. 1. Chain of interconnections between the Components, Capabilities and Actions ontologies.

ontology models both the hardware and software components of the vehicle and their health status (faulty/functional). This piece of information triggers a capability instantiation process inside the capabilities ontology. For example, the navigation capability is instantiated (becomes available) if the thrusters, the Doppler Velocity Log (DVL), the gyroscope and the navigation module are functional. In cases where there is component redundancy for a capability on the AUV the system instantiates a set of capabilities for these components in the form of primary, secondary, tertiary and so on. Regarding the navigation capability mentioned above, if the AUV has also a functional compass component then the system instantiates both a primary and a secondary navigation capability. Consequently, if the gyroscope breaks (the primary navigation capability is lost), the system will determine that it can still navigate since it can use the compass instead (secondary capability). We call the capabilities that achieve the same functionality *semantically equivalent*. Finally the information (instantiations) in the capabilities ontology is used to instantiate actions where the notion of *semantic equivalence* is also applicable. For instance, *semantically equivalent* inspection actions are instantiated if the inspection and navigation capabilities are available to the vehicle irrespective of whether these are primary, secondary or n-ary (precedence). The existence of precedence is intentional and higher precedence implies at least equal, if not better, outcomes compared to lower precedence (e.g. for navigation the gyroscope is more reliable since it is less prone to magnetic interferences).

The process of reasoning about (determining) the available capabilities and actions of the AUV by checking the components’ health status is taking place continuously and in real time during a mission. This enables the AUV to revert to using the higher precedence capabilities and actions should they become available.

## B. Planning & Execution Ontologies

One of the most critical requirements for successful missions is the efficient encoding and management of information from the planning-execution loops. In order to meet this requirement we developed two interconnected ontologies, namely a planning and an execution ontology.

The planning ontology that we use was first introduced in our previous work [1]. It encodes the syntax and the semantics of the MCM Planning Domain Definition Language (PDDL) domain and `problem` that the planner needs to generate mission plans. In this work we have extended the planning ontology by including three high level mission priorities (goals) that drive mission planning. The high level mission priorities that we have encoded are: i) MCM while minimizing the consumed energy of the vehicle, ii) MCM by reacquiring and inspecting the MLOs that are the most likely to be mines first while minimizing the consumed energy of the vehicle, iii) MCM by reacquiring and inspecting the MLOs that have the highest entropy first while minimizing the consumed energy of the vehicle. High level mission priorities are described in more detail in section IV-B.

On the other hand, the execution ontology stores archival information about planning and execution outcomes. These include but are not limited to: time spent planning, mission plans, time spent executing the mission, action execution status (executed, pending, successful, unsuccessful). While this piece of information is very important for post mission analysis it is also used by the planning ontology to update its status accordingly in real time (e.g. mines that have already been inspected do not need to be considered in the planning procedure any more).

## IV. ADAPTIVE MISSION PLANNING AND EXECUTION

Hardware faults and changes in high level mission priorities during mission execution require adaptive mission planning and execution approaches. These approaches firstly enable robustness under a variety of unexpected events, and secondly ensure that mission goals are satisfied to the maximum extent possible. In the event of hardware faults our approach prefers adaptive execution to adaptive planning, by initially searching for *semantically equivalent* actions to be executed. If none is executable then it resorts to replanning given the current state of both the vehicle and the world in conjunction with remaining mission goals (adaptive planning). Prioritizing adaptive execution over adaptive planning is important as mission planning can be very demanding in terms of computational resources [1]. However, when a request for changing high level mission priorities is issued to the vehicle, the framework performs replanning due to effectively invalidating the prior plan in the vast majority of cases. For planning we use the PDDL planner OPTIC [11].

### A. Generating Mission plans

Every PDDL planner requires as input a `domain` and a `problem` for generating mission plans. The `domain` encodes

action definitions that the planner can use to generate a plan given a problem that needs to solving.

Actions comprise parameters, preconditions and effects. In order for an action to be applicable and be considered as part of a plan its preconditions must hold before applying the action and its effects should be contributing towards solving the problem. Snippet 1 illustrates the PDDL MLO reacquisition (`do_reacquire`) action of our domain. The

---

### Snippet 1 PDDL `do_reacquire` action reconstructed from the planning ontology.

---

```
(:action do_reacquire
:parameters (?v - vehicle ?from ?to - Mlopoint)
:precondition (and (at_mlop ?v ?from)
                  (= (reacquired_mlop ?to) 0)
                  (>= (remaining_energy ?v)
                     (distance_mlop ?from ?to))
                )
:effect (and (not (at_mlop ?v ?from))
            (at_mlop ?v ?to)
            (decrease (remaining_energy ?v)
                     (distance_mlop ?from ?to))
            (increase (consumed_energy ?v)
                     (distance_mlop ?from ?to))
            (increase (reacquired_mlop ?to) 1)
            (increase (mission-time)
                     (distance_mlop ?from ?to))
            (increase (cnt_reacquired_mlop ?v) 1)
            (increase (prob_mlop_quotient_sum ?v)
                     (/ (prob_mlop ?to) (prob_mlop ?from)))
            (increase (ent_mlop_quotient_sum ?v)
                     (/ (ent_mlop ?to) (ent_mlop ?from)))
            )
)
```

---

action has two parameters: a vehicle `?v` and two mlopoints `?from` and `?to`. For the action to be applicable the vehicle needs to be at location `?from`, not have reacquired the MLO positioned at `mlopoint ?to` and have enough energy to travel from its location to the mlopoint location. The effect of executing the `do_reacquire` action would be that the vehicle has left its initial location (`not (at_mlop ?v ?from)`) and moved at the location of the mlopoint `?from`. Additionally, the vehicle's remaining and (total) consumed energy have changed relative to the distance between the locations `?from` and `?to`. The mlopoint at location `?to` has now been reacquired, the mission duration has increased and the (total) number of mlopoints that have been reacquired is increased by one. Also, `increase (prob_mlop_quotient_sum ?v) (/ (prob_mlop ?to) (prob_mlop ?from))` increases the sum of probability quotients by the quotient of dividing the probability of the MLO at location `?to` being a mine by the probability of the MLO at location `?from` being a mine. Finally, `increase (ent_mlop_quotient_sum ?v) (/ (ent_mlop ?to) (ent_mlop ?from))` achieves the same but with entropy instead of probability. Both these effects related to probability and entropy are related to high level mission priorities which are described in section IV-B. Due to space limitations, detection, classification and inspection actions are omitted from this paper but can be found in our previous work [1].

Regarding the problem that a PDDL planner needs to solve (i.e. generate a plan), it consists of an initial and a goal state. Starting from the initial state, the planner generates a sequence of actions (plan) that achieve the goal state. Snippet 2 illustrates a small PDDL reacquisition problem. Based on the problem in Snippet 2 the planner must generate

---

**Snippet 2** PDDL reacquisition problem reconstructed from the planning ontology.

---

```
(:objects mlopoint1 mlopoint2 mlopoint3 - Mlopoint
nessie - Vehicle)

(:goal
  (and
    (>= (remaining_energy nessie) 0)
    (= (cnt_reacquired_mlop nessie) 2)
  )
)

(:init
(at_mlop nessie mlopoint3)
(= (cnt_reacquired_mlop nessie) 0)
(= (prob_mlop_quotient_sum nessie) 0)
(= (ent_mlop_quotient_sum nessie) 0)
(= (distance_mlop mlopoint1 mlopoint2) 7.19)
(= (distance_mlop mlopoint2 mlopoint1) 7.19)
(= (distance_mlop mlopoint1 mlopoint3) 6.05)
(= (distance_mlop mlopoint3 mlopoint1) 6.05)
(= (distance_mlop mlopoint2 mlopoint3) 4.32)
(= (distance_mlop mlopoint3 mlopoint2) 4.32)
(= (reacquired_mlop mlopoint1) 0)
(= (reacquired_mlop mlopoint2) 0)
(= (reacquired_mlop mlopoint3) 1)
(= (prob_mlop mlopoint1) 0.606)
(= (ent_mlop mlopoint1) 0.967)
(= (prob_mlop mlopoint2) 0.755)
(= (ent_mlop mlopoint2) 0.803)
(= (prob_mlop mlopoint3) 1000.0)
(= (ent_mlop mlopoint3) 1000.0)
(= (remaining_energy nessie) 300.124)
(= (consumed_energy nessie) 0)
(= (mission-time) 0)
)
(:metric maximize (remaining_energy nessie))
)
```

---

a plan in which our vehicle (nessie) will reacquire two MLOs (mlopoints) without spending more energy than it has available (goal state). The initial state indicates that our vehicle starts at mlopoint3, the MLO at this location has been reacquired and the distances between mlopoints (MLOs) are known. Associated probabilities, entropy, remaining energy as well as consumed energy are also known. The metric maximize (remaining\_energy nessie) indicates that the high level priority of the mission is for the vehicle to reacquire the MLOs located at mlopoint1 and mlopoint2 in a way that the remaining energy on the vehicle is maximized (the shortest possible distance is travelled). The solution obtained is suboptimal since the planner utilizes Weighted A\* (WA\*) to traverse the planning graph to find a plan that satisfies the metric.

## B. High Level Mission Priorities

High level mission priorities are taken into consideration for the reacquisition and inspection of mines, and may be affected by changes in the situation on the surface. For example, a ship may be en-route to the area, and probability-efficient mine clearance given the AUV's remaining energy may be required. As was mentioned in section III-A we have considered three high level mission priorities in total.

The first is to generate a plan for MCM that minimizes the consumed energy of the AUV, that is, reacquire and inspect all MLOs in a manner that the smallest possible distance is travelled. We call this energy-efficient reacquisition and inspection. This effectively formulates a Open Loop Symmetric Travelling Salesman Problem (OSTSP) [12], [13] that the planner needs to solve. The TSP is open loop because the vehicle does not require to return to its initial position. In addition it is symmetric because we assume that for travelling from  $MLO_i$  to  $MLO_j$  the AUV requires the same amount of energy (distance) for travelling from  $MLO_j$  to  $MLO_i$ . For doing so we use the following metric within the problem definition: metric minimize (cosumed\_energy auv) or equivalently metric maximize (remaining\_energy auv).

The second high level mission priority is to generate a plan by reacquiring and inspecting all the MLOs that are the most likely to be mines first (we call this probability-efficient reacquisition) while concurrently minimizing the consumed energy of the vehicle. This formulates a multi-objective optimization problem. To solve this problem we employ the weighted sum scalarizing method that scalarizes a set of objectives into a single objective by multiplying each objective with a weight. The weighted sum method is given by the following:

$$\min F(x) = \sum_{i=1}^n w_i f_i(x) \quad (1)$$

$$s.t. \quad w_i \geq 0, \quad i \in 1, \dots, n \quad (2)$$

$$\sum_{i=1}^n w_i = 1 \quad (3)$$

where  $F(x)$  is the single objective,  $f_i(x)$  is the set of objectives and  $w_i$  the weights. The probability-efficient reacquisition objective in our case is formulated as minimizing the sum of probability quotients (prob\_mlop\_quotient\_sum auv) as shown in section IV-A. In order to apply the weighted sum to form a single objective we need to normalize each objective so that it is expressed in the same range. The linear normalization is given by:

$$f_N(x) = (f(x) - oldMin) \frac{newRange}{oldRange} + newMin \quad (4)$$

where,  $f_N(x)$  is the normalized objective,  $f(x)$  is the original one,  $oldMin$  is the minimum value of  $f(x)$ ,  $newRange$  ( $newMax - newMin$ ) is the range of  $f_N(x)$  that we desire,  $oldRange$  ( $oldMax - oldMin$ ) is the range of  $f(x)$  and  $newMin$  is the desired minimum value of  $f_N(x)$ .

Finally, the third high level mission priority is to generate a plan for MCM by reacquiring and inspecting the MLOs

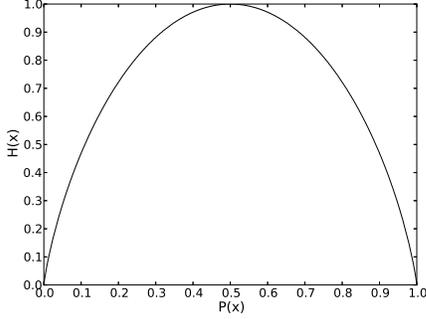


Fig. 2. Entropy versus Probability

that have the highest entropy first (we call this entropy-efficient reacquisition) while concurrently minimizing the consumed energy of the vehicle. Again this formulates a multi-objective optimization problem. The entropy-efficient reacquisition objective is formulated as minimizing the sum of entropy quotients (`ent_mlop_quotient_sum` `auv`) as shown in section IV-A. In our case entropy  $H$  is an indication of the uncertainty level about an MLO being a mine ( $H(MLO_i)$ ) and is related to the probability of an MLO being a mine ( $P(MLO_i)$ ) as follows:

$$H(MLO_i) = -P(MLO_i)\log_2 P(MLO_i) - (1 - P(MLO_i))\log_2(1 - P(MLO_i)) \quad (5)$$

This third high level mission priority is ideal for cases where we want to prioritize high uncertainty reduction and high information gain first in an energy-efficient manner. Fig. 2 illustrates the relationship between entropy and probability.

## V. EXPERIMENTAL SETUP

Experiments were conducted in a simulated MCM environment consisting of spherical objects that represent mines. The vehicle used was a simulated version of the Nessie V AUV [14]. The task for the vehicle was to plan and execute MCM missions that are composed of four phases: detection, classification, reacquisition and inspection of mines using appropriate sensors (e.g. sonar, cameras, DVL, etc.). For the detection phase, the AUV is given a square area to survey, and it generates a lawnmower trajectory which is asserted into the KB along with associated semantic information (e.g. the lawnmowerpoints of the trajectory are unvisited). The vehicle then generates a plan and surveys the area for mines. While surveying, detections are asserted into the KB and are used as input for classification. Classification generates MLOs with associated coordinates, probability and entropy, and asserts these into the KB. This information along with the reacquisition status and inspection status of the MLOs forms the basis for planning the reacquisition and inspection phases. For a detailed description of how the vehicle proceeds with detection, classification, reacquisition and inspection the interested reader should refer to [1]. Fig. 3 illustrates a

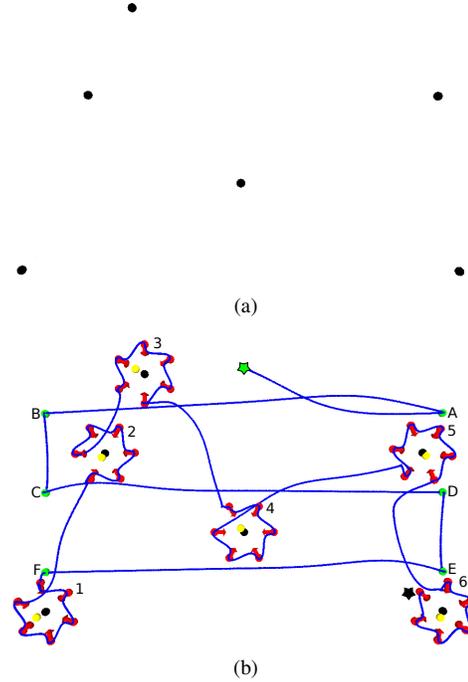


Fig. 3. (a) Geometrical setting of mines in simulation. (b) MCM mission. The mission start point is represented by the green star while the lawnmower points by green spheres (A-F). Areas 1-6 represent reacquisition and inspection areas. Red spheres with attached arrows represent inspection points, yellow spheres are classified MLOs while black spheres are the positions where the MLOs are estimated to be after reacquisition. The black star represents the mission end point.

TABLE I  
AVERAGE MISSION STATISTICS.

High level priority	Energy	Energy-Prob	Energy-Ent
Planning time (secs)	165	236	235
Execution time (secs)	676	693	698
Total mission time (secs)	841	929	933
Total distance travelled (m)	173.29	183.68	188.32

geometrical setting of mines in our simulation environment (Fig. 3a) as well as an MCM mission based on that setting (Fig. 3b).

For testing our system's fault recovery we simulate hardware faults and check our system's ability to overcome them and carry out its mission to the maximum extent possible. In addition, we test our system's behaviour under different high level mission priorities and its ability to adapt when changes in priorities are requested.

## VI. RESULTS

This section presents the results of the experiments that were carried out in simulation. For all the multi-objective optimizations the weight used for each objective was 0.5 ( $w = 0.5$ ). The first set of results comprises the time the vehicle spent planning, executing the mission as well as the total mission time and total distance travelled for the three different high level priorities (see Table I).

In Table I the average mission statistics are generated from running missions on a total of 10 random geometrical

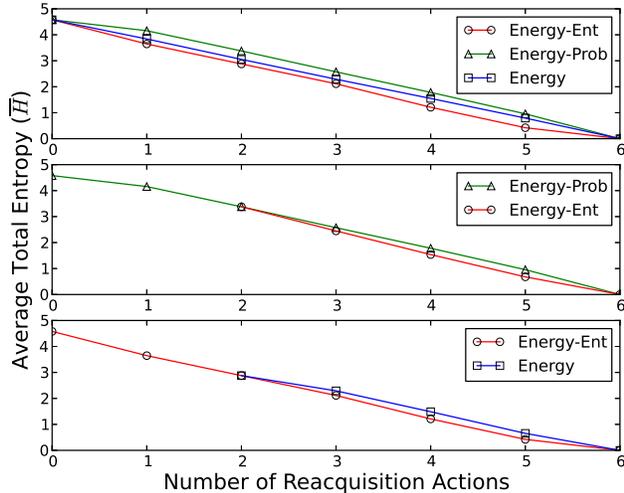


Fig. 4. Average total entropy reduction over the number of MLO reacquisition actions taken by the vehicle: under the three different high level mission priorities (upper graph), when adapting from the Energy-Prob to the Energy-Ent priority (middle graph), when adapting from the Energy-Ent to the Energy priority (bottom graph).

configurations of 6 targets within an area of  $22 \times 12$  meters under the three different high level mission priorities. For each geometrical configuration, after classification, every MLO is associated with a random probability within the range of  $0.52 - 0.99$ . It can be seen that the average total distance travelled for the energy-efficient (Energy) reacquisition and inspection is smaller than the ones combining energy efficiency with probability efficiency (Energy-Prob) and energy efficiency with entropy efficiency (Energy-Ent). This is expected since in the latter two high level priorities we perform multi-objective optimization. This means that we are trading total distance (energy) for visiting MLOs that are most probable to be mines or yield the highest information gain first. Additionally, it takes more time to plan a mission when performing multi-objective optimization because the system needs to calculate the maximum and the minimum of each objective first for using them in the weighted sum approach (see Equation 4).

Despite the fact that the system consumes more energy to execute its mission (distance travelled is increased) it compensates in terms of information gain (uncertainty reduction) and high certainty exploitation as shown in Fig. 4. The upper graph in Fig. 4 shows the average total entropy reduction under the three high level mission priorities. For the Energy-Ent reacquisitions the graph line is lower compared to the Energy and Energy-Prob reacquisitions. Additionally, for Energy-Prob the graph line is higher than the rest. This means that for Energy-Ent the uncertainty about the environment is always lower after every reacquisition compared to the rest while for Energy-Prob it is always higher. For the Energy reacquisition it is somewhere in-between.

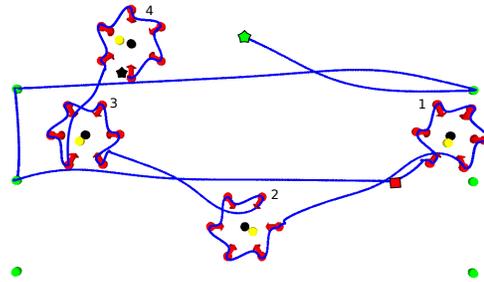


Fig. 6. Adaptation due to sonar fault (red square). The mission start point is represented by the green star while the mission end point is represented by the black one.

Given that the probabilities for MLOs are in the range of  $0.52 - 0.99$  means that during Energy-Prob reacquisitions the AUV tends to visit MLOs with higher probability first (see also Fig. 2). Moreover, these findings are supported by the middle and bottom graphs in Fig. 4 which show how the average total entropy reduction is affected when adapting from one high level priority to another.

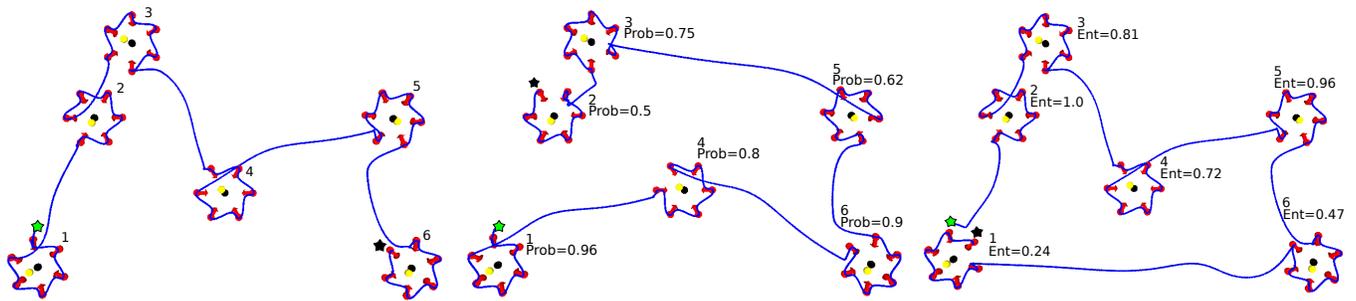
Figs. 5a-5c illustrate the reacquisition and inspection trajectories followed by the simulated AUV due to planning based on different high level mission priorities. In the mission illustrated in Fig. 5d, the initial plan for the vehicle was to reacquire and inspect MLOs with the Energy-Prob high level mission priority as shown in Fig. 5b. After visiting areas 1 and 4 a request for changing the high level mission priority into Energy-Ent took place and the vehicle adapted accordingly. A similar adaptation is shown in Fig. 5e but for switching from the Energy-Ent high level priority (Fig. 5c) to the Energy one.

Finally, Fig. 6 shows a simulation of a sonar fault and the recovery of the system. The lawnmower survey is interrupted due to the sonar being a critical component for the survey. That is, there is no component that will provide a semantically equivalent survey capability and consequently a semantically equivalent survey action. As a result, the vehicle adapts and proceeds with classification, reacquisition and inspection (using a camera) of the already detected targets. This allows the vehicle to satisfy its mission goals to the maximum extent possible since it is unable to continue surveying.

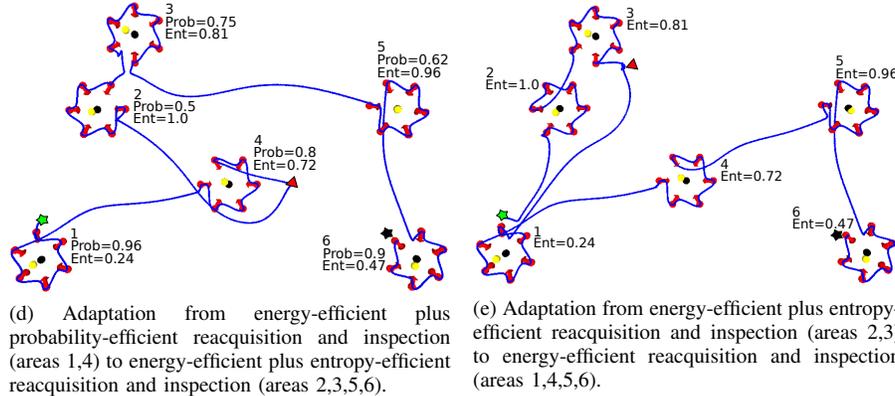
## VII. CONCLUSIONS

In this paper we have presented a framework that enables adaptation and recovery due to hardware faults and changes in high level mission priorities. The results demonstrate the power of an ontology-based knowledge representation and reasoning approach in driving adaptation.

Adaptation due to high level mission priority changes is an important feature because it facilitates the need for adaptation due to real time mission requirement changes. The experimental results demonstrate the robustness of the framework in that respect. Moving from the standard MCM missions where energy consumption and/or execution time govern the mission, we have provided an alternative perspective in which



(a) Energy-efficient reacquisition and inspection. The AUV proceeds by reacquiring and inspecting MLOs in all the areas in the following order: 1,2,3,4,5,6. (b) Energy-efficient plus probability-efficient reacquisition and inspection. The AUV proceeds by reacquiring and inspecting MLOs in all the areas in the following order: 1,4,6,5,3,2. (c) Energy-efficient plus entropy-efficient reacquisition and inspection. The AUV proceeds by reacquiring and inspecting MLOs in all the areas in the following order: 2,3,4,5,6,1.



(d) Adaptation from energy-efficient plus probability-efficient reacquisition and inspection (areas 1,4) to energy-efficient plus entropy-efficient reacquisition and inspection (areas 2,3,5,6). (e) Adaptation from energy-efficient plus entropy-efficient reacquisition and inspection (areas 2,3) to energy-efficient reacquisition and inspection (areas 1,4,5,6).

criteria such as probability and entropy are also considered with promising results. Finally, the fault recovery capabilities of our framework provide the means for minimizing the risk faults pose for underwater missions.

#### ACKNOWLEDGMENT

The authors would like to thank the Ocean Systems Laboratory members at Heriot-Watt University for their input and support.

#### REFERENCES

- [1] G. Papadimitriou and D. Lane, "Semantic Based Knowledge Representation and Adaptive Mission Planning for MCM Missions using AUVs," in *Proceedings of the OCEANS'14 MTS/IEEE Conference*, Taipei, April 2014.
- [2] M. Tenorth, "Knowledge processing for autonomous robots," Ph.D. dissertation, Technische Universität München, 2011.
- [3] M. Tenorth and M. Beetz, "KnowRob – A Knowledge Processing Infrastructure for Cognition-enabled Robots," *International Journal of Robotics Research (IJRR)*, vol. 32, no. 5, pp. 566 – 590, April 2013.
- [4] S. Lemaignan, R. Ros, L. Mösenlechner, R. Alami, and M. Beetz, "ORO, a knowledge management platform for cognitive architectures in robotics," in *International Conference on Intelligent Robots and Systems*, 2010.
- [5] P. Patrón, E. Miguelanez, Y. R. Petillot, D. M. Lane, and J. Salvi, "Adaptive mission plan diagnosis and repair for fault recovery in autonomous underwater vehicles," in *OCEANS 2008*. IEEE, 2008, pp. 1–9.
- [6] P. Patrón, "Semantic-based adaptive mission planning for unmanned underwater vehicles," Ph.D. dissertation, Heriot-Watt University, Edinburgh, 2010.
- [7] J. R. Boyd, *A discourse on winning and losing*. Unpublished set of briefing slides available at Air University Library, Maxwell AFB, Alabama, 1987.
- [8] C. McGann, F. Py, K. Rajan, H. Thomas, R. Henthorn, and R. McEwen, "T-rex: A deliberative system for auv control," *ICAPS WS on Planning and Plan Execution for Real-World Systems*, Providence, RI, USA, 2007.
- [9] M. P. Brito, N. Bose, R. Lewis, P. Alexander, G. Griffiths, and J. Ferguson, "The role of adaptive mission planning and control in persistent autonomous underwater vehicles presence," in *Autonomous Underwater Vehicles (AUV), 2012 IEEE/OES*. IEEE, 2012, pp. 1–9.
- [10] J. Frank and A. Jónsson, "Constraint-based attribute and interval planning," *Constraints*, vol. 8, no. 4, pp. 339–364, 2003.
- [11] J. Benton, A. J. Coles, and A. I. Coles, "Temporal planning with preferences and time-dependent continuous costs," in *Proceedings of the Twenty Second International Conference on Automated Planning and Scheduling (ICAPS-12)*, June 2012.
- [12] D. L. Applegate, R. E. Bixby, V. Chvatal, and W. J. Cook, *The traveling salesman problem: a computational study*. Princeton University Press, 2011.
- [13] A. N. Letchford, S. D. Nasiri, and D. O. Theis, "Compact formulations of the steiner traveling salesman problem and related problems," *European Journal of Operational Research*, vol. 228, no. 1, pp. 83–92, 2013.
- [14] R. Baxter, J. Cartwright, J. Clay, O. Clert, B. Davis, J. Lopez, F. Maurelli, Y. Petillot, P. Patrón, and N. Valeyrie, "Nessie v autonomous underwater vehicle," *Student Autonomous Underwater Competition-Europe (SAUC-E)*, 2010.