Contingency Planning for Long-Duration AUV Missions

Catherine Harris
School of Computer Science
University of Birmingham
Birmingham, United Kingdom B15 2TT
Email: C.A.Harris.1@cs.bham.ac.uk

Richard Dearden
School of Computer Science
University of Birmingham
Birmingham, United Kingdom B15 2TT
Email: R.W.Dearden@cs.bham.ac.uk

***Abstract*—In recent years, the use of autonomous underwater vehicles has become increasingly popular for a wide variety of applications. As the cost of deploying a vehicle and the risk of loss or damage are often high, AUV missions typically consist of simple pre-scripted behaviours. Designed to minimise risk to the vehicle and its scientific cargo, these behaviours are inevitably overly-conservative, reserving a significant proportion of battery as a contingency should usage be higher than expected. Consequently, in the average case, the vehicle is not used to its full potential.**

As environments in which AUVs operate are dynamic and their effect on the vehicle is often uncertain, it is difficult to accurately predict the resource cost of a mission or individual task in advance. By modelling this uncertainty and allowing the vehicle to observe both the progress of the mission and the surrounding environment, the mission plan may be autonomously refined during operation. For example, in the event that resource usage, such as battery power, is observed to be lower than expected, the vehicle can schedule additional data collection tasks. Conversely, if the resource usage is higher than expected, the vehicle can remove lower priority tasks from the mission plan in order to increase the probability of successful recovery without the need to abort the mission.

Such planning becomes increasingly beneficial when performing longer duration missions comprised of many tasks.

This paper discusses the development of a new autonomous planning algorithm which models the uncertainty in the AUV domain and attempts to maximise the collection of scientific data without compromising the safety of the vehicle. It includes a technical overview, recent results and a discussion of the research in the context of potential applications, focusing on long-range and low-cost vehicles.

I. Introduction and motivation

Until recently, AUVs typically had very little on-board intelligence, instead favouring 'lawnmower' surveys and other simple behaviours [1]. By analysing data from the vehicle between deployments, operators could tailor subsequent missions to further investigate areas of interest. The recent development of long-range vehicles, such as Autosub Long-Range [2], designed to perform missions lasting many days or weeks, has the potential to revolutionise the collection of oceanographic data. However, as the progress of long-duration missions cannot be periodically reviewed by human operators, long-range vehicles will require an increased level of autonomy to fully capitalise on their increased capabilities.

Long-duration missions can be thought of as over-subscribed planning problems, where finite amounts of battery power and data storage space limit the number and duration of data-collection tasks achievable by the vehicle. Designed to safeguard the vehicle whilst operating in inherently uncertain environments, pre-scripted missions are inevitably over-conservative, assuming worst-case battery and memory usage. For large-scale planning domains, such as AUV missions, it is infeasible to compute the optimal course of action for every possible eventuality in advance. Instead of relying on prior estimates, by observing resource usage during plan execution the behaviour of the vehicle can be optimised. If the current resources are sufficiently high, additional data-collection tasks can be added to the plan. Conversely, if resources are consumed faster than expected, contingencies, such as removing low-priority data-collection tasks from the plan, can be implemented.

By representing the AUV domain as an over-subscribed planning problem, we show that enabling a vehicle to autonomously refine the mission plan during plan execution facilitates additional data-collection without compromising the safety of the vehicle. Although our investigation to date has been performed purely in simulation, the results have encouraged further development of a contingency planning algorithm for long-range AUVs.

II. AUV problem domain

A. Scenario

Our 'AUV problem' scenario envisages a vehicle with limited battery power and on-board memory conducting a long-duration mission, collecting scientific data from specific pre-defined survey areas. The location of these survey areas along with an estimate of the value of each data set are assumed to be provided in advance by a scientist. The value of the data-sets does not need to be precise, but should reflect any preference the scientist may have for one data-set over another. At each survey area, if the vehicle has sufficient unused memory it can perform a data-collection behaviour, such as mapping the extent of a chemical plume. The vehicle is able to travel between survey areas using a pre-defined network of waypoints and may surface at any point during the mission. Whilst on the surface, the vehicle may attempt to

transmit data-sets back to base via a satellite link. Data-sets are only of value to scientists if they are successfully recovered from the vehicle, either by being present on the hard-drive at the point of vehicle collection or by being transmitted by the vehicle mid-mission. Without this stipulation, the potential value of the data and the cost of losing it (such as through the corruption of onboard storage or the total loss of the vehicle) would not be represented within the problem.

All actions will consume battery power; however, the exact amount consumed is not known in advance but modelled with a probability distribution specific to each action. For example, there is much less uncertainty in the power used while surfacing than in collecting data from a survey area. Similarly, the amount of memory a data-set will consume once compressed is also uncertain.

During the execution of each action we have assumed there is a small chance of mission failure. This failure rate represents the inherent risks to both the success of the mission and the safety of the vehicle itself. Once the vehicle has either collected data from all of the pre-defined survey areas or the remaining battery power is low, the vehicle should attempt to move to the predefined end location and surface ready for recovery by a support vessel (modelled by the *EndMission* action below). In the event of an unexpected situation, the vehicle always has the option to abort the mission, contact the base and await recovery.

B. Modelling the Problem

The AUV problem can be modelled as a Markov Decision Process (MDP) [3]. Formally, an MDP is a tuple $\langle S, A, T, R \rangle$ where:

- S is the set of states the system can be in. Each state $s \in S$ is defined by the following set of *state variables*:
  - Location of the vehicle: $\{\text{start}, \text{end}, l_1, l_2, \ldots\}$.
  - Depth of the vehicle: $\{\text{surface}, \text{depth}\}$.
  - Data-sets which have been collected (one boolean variable per set).
  - The size of each data-set (a positive real number per set, zero for uncollected sets).
  - Data-sets which have been transmitted (one boolean per set).
  - Status of the mission: $\{\text{active}, \text{complete}\}$.
  - Battery power (positive real).
  - Available memory (positive real).

  S is a mixture of discrete and continuous variables, which, as we will see, makes computing a plan particularly difficult.
- A is the set of discrete actions available to the vehicle. As the action space is large, we use the following six parameterised *action schemas* to concisely represent the full action space:
  - *CollectData*(d): Collect data-set d if memory and battery power are sufficient and the vehicle is in the correct location.
  - *Move*(l_1, l_2): If the vehicle is at location l_1, move to l_2.
  - *Dive()* - Move from the surface to depth at the current location.
  - *Surface()* - Move from depth to the surface.
  - *TransmitData*(d): Attempt data transmission provided the vehicle is at the surface and has data-set d. The *TransmitData* action is stochastic as transmissions may fail.
  - *EndMission()* - Wait for recovery. Upon performing this action, no further actions may be performed.

  All actions reduce the battery by a quantity modelled by a Gaussian distribution. Additionally, *CollectData* similarly reduces the amount of available memory, while *TransmitData* increases it by the size of the transmitted data-set.
- T - the transition function $T(s, a, s')$. This represents the probability that performing action a in state s will leave the system in state s'. This allows stochastic action effects, such as occasional action failure, to be represented. Since our states are a mixture of discrete and continuous variables, the transition function is also a mixed distribution.
- R is the reward function $R(s, a, s')$ which specifies the immediate reward for performing action a in state s and transitioning to s'. Positive rewards are given when the vehicle successfully delivers a data-set. For example, successfully performing *TransmitData*(d) in a state where data-set d has been collected but not transmitted would result in a reward. There are also large negative rewards for failing to perform *EndMission* before running out of battery, or for performing an action that is invalid in the current state. Smaller negative rewards are given for not finishing the mission on the surface at the end location.

When computing the value of a plan, we use a discount factor to represent the small probability that the vehicle could be lost or suffer damage at each step as a result of environmental factors outside of the control of the planner. The discount factor slightly reduces the value of future rewards based on the number of actions the vehicle would need to perform to reach the rewarding state. This means that the sooner a reward can be reached, the higher its value.

The combined use of penalties and rewards favours plans which prioritise the collection of high value data-sets without compromising the overall safety of the vehicle. This means that during a mission, should the vehicle find it has used significantly more battery power than expected to complete a series of actions, the option to travel to the recovery location and end the mission becomes more attractive than continuing to collect additional data-sets.

III. Validation of approach

A. Over-subscription planning

Benton *et al* [4] state that the inclusion of optional, numeric goals allows the user more freedom in expressing what they desire from the final plan. In classical planning, a plan is considered successful upon the completion of a set of goals.

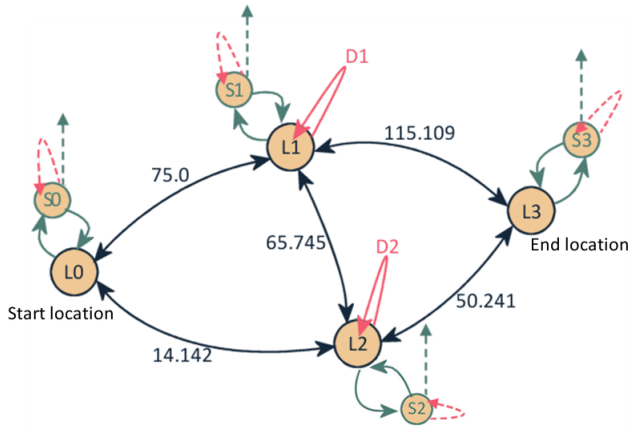

Figure 1. Schematic illustrating the small AUV problem. Circles represent locations that the vehicle can visit, including those on the surface (prefixed S). The specified end location is L3. Black arrows represent possible *Move* actions and are labelled with the distance between locations. Solid green lines represent the *Surface* and *Dive* actions. Solid pink arrows represent *CollectData* actions. Dashed pink arrows represent *TransmitData* actions. The dashed green arrow represents the *EndMission* action.

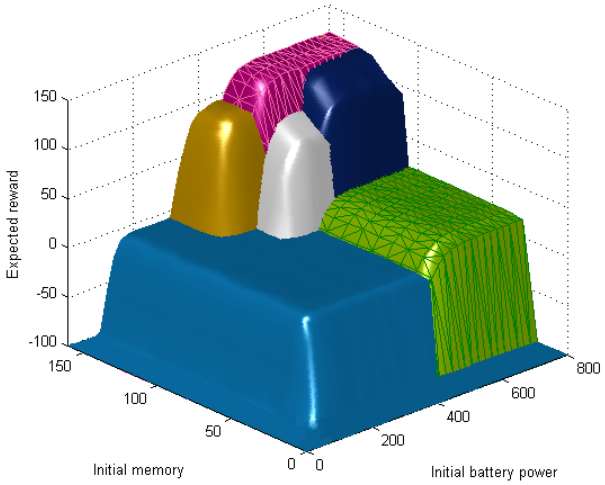

Figure 2. The optimal value function produced by performing Monte Carlo simulation on the set of straight-line plans from the small AUV problem. Each colour/pattern represents a different plan.

Instead, over-subscription or partial-satisfaction planning with numeric goals allows the planner to select a subset of goals to achieve. Goal selection primarily depends on the amount of resources available to the vehicle and the rewards achievable for completing these goals. For a long-range mission, such as completing a transect of an ocean ridge or passage, we envisage that the collection of high resolution data at some locations during the transect will be of higher scientific value than at others. The use of numeric goals allows the user to specify the relative importance of each data-set or data collection behaviour to the planner.

B. Switching between plans

As the actions in the AUV problem use an uncertain amount of resources, committing to a single straight-line plan is likely to be sub-optimal. We hypothesized that by allowing the vehicle to change its planned behaviour during execution in response to unexpected situations, the expected reward for this 'switching-plan' would increase above and beyond that of any single straight-line plan. Following on from work by Bresina *et al* [5], in which the authors computed the optimal value function associated with the Mars rover planning problem as a function of continuous resource variables, we performed an investigation to compare the optimal value functions for both straight-line and switching plans within the simplified AUV domain. For any combination of initial resources the optimal value function represents the expected reward obtained by following a particular plan.

To perform this investigation, we hand-generated a varied subset of viable plans from the small version of the AUV problem shown in figure 1. The problem comprised of four locations with two data-sets for the vehicle to collect. When calculating the value function for each straight-line plan, we considered each of the plans in the subset independently.

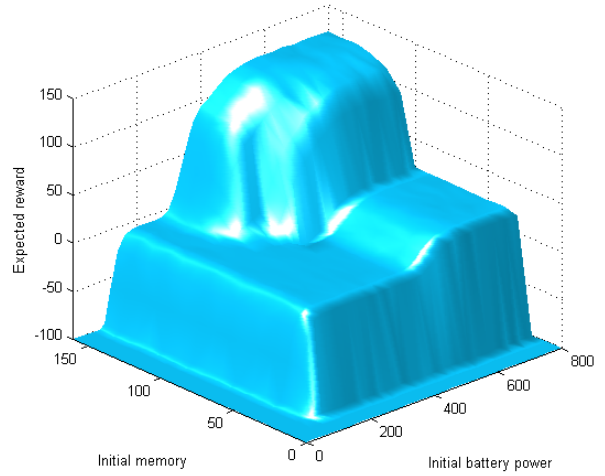

Figure 3. The optimal value function produced by simulating a switching planner, showing the expected reward of switching between plan fragments at run-time. When combined with the equivalent function for straight-line plans (shown in figure 2), the former is completely dominated by this function.

As the resource usage of each action is uncertain, Monte Carlo simulation was performed by running each plan 5000 times for every combination of initial resource values. The resulting reward values from each resource combination were then averaged and combined with the data from every other straight-line plan in the subset to produce the optimal value function, as shown in figure 2. Each colour represents a different plan, with higher expected rewards indicating the most successful plans. Negative reward indicates that the plan received more penalties than rewards. At the start of each mission, the reward is zero. To compute the equivalent value function for switching between plans during execution, the

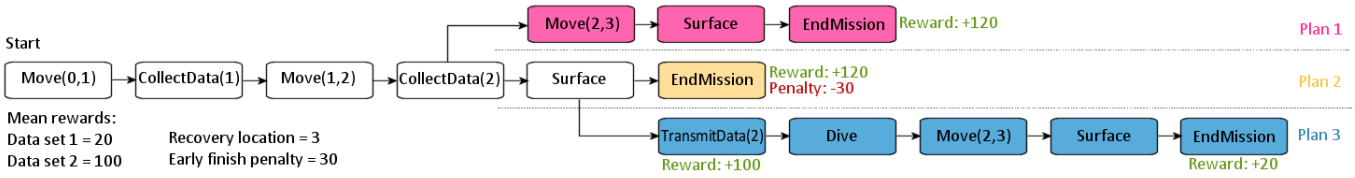

Figure 4. Block diagram of three plans from the small AUV problem. The initial actions (starting from the left) are common to multiple plans and shown in white, while action sequences unique to each plan are coloured separately. Each of the three straight-line plans is a single route from left to right. The switching planner can switch between plans at the branch points.

plans were examined and points common to multiple plans identified. From each of these common points, referred to as 'branch points', Monte Carlo simulation was again performed for every combination of resource values to find the associated branch condition, i.e. the point at which a change in resources causes one plan to become better than another. As shown in figure 3, the optimal value function completely dominated that of the straight-line plans, exceeding the expected reward for all resource combinations. This shows that branching is at least as good as the best straight-line plan. In fact, at multiple locations the switching plan significantly outperforms the best straight-line plan. To illustrate this finding, we will discuss the value functions of multiple plans in more detail. Figure 4 shows the three plans chosen for this example. Plan 1 collects both data-sets before moving to location 3 and ending the mission. Plan 2 also collects both data-sets but instead surfaces and ends the mission in location 2. Plan 3 collects both data-sets and attempts to transmit data-set 2 via a satellite before moving to location 3 and ending the mission. Temporarily ignoring the shaded switching plan, in figure 5 we can see that plan 1 is optimal for initial battery values greater than 480 units (marked with a dashed black line). Below this point, plan 2 becomes optimal. This is due to the additional move action in plan 1 which causes the average resource requirement for this plan to be higher than that of plan 2. However, as plan 2 does not result in the vehicle ending the mission at the specified recovery location, the total reward for this plan is subjected to a 30 point penalty, reducing the maximum reward achievable by following this plan. The point in the value function where the resource requirements exceed the amount of resources available to the vehicle is not a sharp threshold but a curve. This is due to the uncertainty associated with resource usage. At the higher end of the curve representing plan 1 in figure 5, the plateau in the function indicates that the increasing majority of runs had sufficient battery to complete the plan and earn the full reward. Conversely, as the available battery decreases the gradient of the function becomes much steeper. This indicates that an increasing majority of runs failed to complete the plan before exhausting the battery and suffered a 100 point penalty. Within this interval (400-530 battery), the value function for the switching plan outperforms both plan 1 and plan 2. This is because for some runs of plan 1, where battery usage was higher than expected, the switching plan was able to switch to a simpler plan, namely plan 2, and complete the plan successfully. Although the reward achievable by

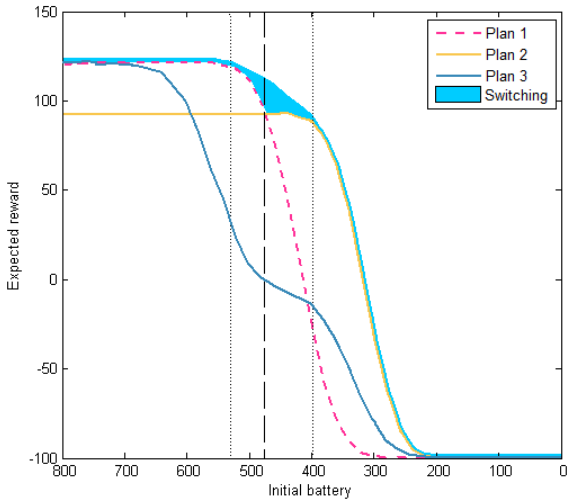

Figure 5. Cross-section of the optimal value function for the three plans outlined in figure 4. The switching plan is shown to be at least as good as the straight-line plans, significantly outperforming all plans between 400 and 530 battery. The dashed black line indicates the point at which plan 1 outperforms plan 2.

following plan 2 is lower than that of plan 1, it is significantly higher than the penalised value received for failure to complete plan 1. In addition to this, the switching planner is able to capitalise on additional rewards when the resource usage of a run is lower than expected. For example, the initial resources might suggest that following plan 2 is optimal. However, if during plan execution the resource usage during the first few actions is lower than expected, the switching planner is able to switch to plan 1 instead. Switching to plan 1 avoids the 30 point penalty suffered by plan 2 and increases the expected reward at the end of the mission.

Although imposing an upper bound on resource usage may prevent mission failure, such as the vehicle running out of battery, committing to a single straight-line plan in advance may cause the vehicle to miss valuable data-collection opportunities. By allowing the vehicle to change plan during execution, contingency plans can be found by the planner to handle either scenario, should they arise. This is especially useful during long-duration missions where it is significantly harder to accurately predict the resource usage of a vehicle as plans may be many days or weeks in length.

IV. Plan generation

In order to benefit from dynamic switching between plans at run-time, we first need an algorithm capable of generating an initial plan. Plan generation is not trivial for the AUV problem due to uncertain resource usage, resource constraints and numeric goals. Traditional goal-directed planners are not suitable for the AUV domain as there is no clearly defined goal state; the goal of the AUV domain is to maximise the reward given the resources available to the vehicle. The planning algorithm must be able to reason about uncertainty and continuous resource usage as well as stochastic action effects, such as that produced by the *TransmitData* action. All plans must satisfy the constraints of the planning domain (i.e. they must not let the vehicle run out of battery) whilst seeking to maximise the reward achieved by successfully collecting and delivering scientific data-sets.

A. Related work

Although there are algorithms in the planning literature which can generate plans to solve problems with similar characteristics to the AUV problem, the authors have yet to encounter a solution which satisfies all of the problem requirements. Many classical planning algorithms, including IPP [6] and MADbot [7] represent the resource usage of an action as the addition or subtraction of a discrete value from a resource variable. For realistic planning domains, such as that of the AUV problem, it is overly simplistic to represent the resource usage in this way. In reality, the level of resources such as battery power will vary constantly whilst the vehicle is in operation. Although MADbot is capable of changing its plan during execution in order to add replenishment goals when the amount of a resource falls below a threshold, the system generates these replenishment goals based on its own estimations of usage, rather than by using values observed during execution. This can lead to scenarios where the planner believes it has more resources than are actually available and consequently delays generating a replenishment goal. Although this may not be an issue in practice for some robot applications, in the AUV domain it is critical that the planning system is able to observe and react to the actual resource usage of the vehicle during a mission, preventing it from exhausting the battery.

Unlike MADbot [7], Bresina *et al*'s [5] action schema does not require action effects to alter the value of a resource by a fixed discrete amount. Instead, they explicitly represent uncertain and continuous resource consumption in the Mars rover domain using a probability distribution with a mean equal to the expected usage for that action. By formulating a simplified version of the Mars rover domain as a Markov Decision Process (MDP) with continuous state variables, Bresina *et al* [5] computed the optimal value function associated with the rover problem. By evaluating the transition and reward functions, the optimal action to perform in any given state can be found. This is known as the 'optimal policy'. In theory, by finding the optimal policy for a full representation of the AUV domain, the problem would be solved. The vehicle would be able to look-up the best course of action given its current state and available resources. However, Bresina *et al* [5] state that in practice, standard methods of solving MDPs, such as dynamic programming become computationally infeasible when the search space is large. In a follow-up paper, Feng *et al* [8] address this computational complexity by using the optimal value function to classify regions of the continuous state space into discrete states according to the value of their utility. They vary the level of discretisation across the search space, performing a much finer discretisation over areas where the expected utility changes most rapidly (indicated by curves in the value function) whilst grouping states with a similar expected utility (plateaus in the value function) into a single state. By reducing the size of the search space, Feng *et al* [8] significantly reduce the computational complexity of generating a policy for large-scale problems.

B. Methodology

Inspired by the work of Feng *et al* [8], we are investigating the use of an MDP for plan generation. However, instead of performing a finer discretisation across sections of the state space where the value function changes rapidly, we are investigating the effectiveness of varying the level of discretisation in response to observations made during plan execution.

By dividing both battery and memory into several discrete 'sections', we drastically reduce the size of the state space in the AUV problem. To illustrate this concept, suppose we decide to divide the total battery into five sections, representing *very high, high, medium, low* and *very low* battery power remaining. Instead of having two continuous state variables, we now have five times as many discrete states which approximate the continuous variables. As this significantly reduces the complexity of calculating the transition function, it is now possible to generate a policy for the AUV problem. Although this policy dictates the optimal action to take in each of the states represented, by performing such a coarse level of discretisation as in this example, the resulting states may no longer be an accurate representation of the true state of the vehicle. A fine degree of discretisation will cause plan generation to take significantly longer. Although this is less of an issue when generating the initial plan (as this is expected to occur prior to vehicle launch), minimising the time and computation required to replan during execution is a key concern.

As missions progress, areas of the state space will become unreachable or irrelevant. For example, once the vehicle has collected a particular data-set it will not be able to return to a state where this is not the case (N.B. although transmitting a data-set may remove it from memory, a record of the collection is kept to prevent the vehicle immediately re-collecting data from the same survey area). During replanning, such areas of the state space can now be ignored by the planner, reducing the computation required for searching for new contingency plans. At this point, the remaining state space may be discretised to a finer degree, increasing the quality of the resulting policy.

Early results for initial plan generation have been encouraging. The generated policy dictates separate strategies depending on the resources currently available to the vehicle. The plans generated by these scenarios are comparable to the hand-built examples in section III-B. For example, when battery is low, the plan is shortened to prioritise reaching the recovery location over trying to collect additional data-sets. Conversely, when battery is high, the vehicle collects as many data-sets as possible before heading to the recovery location, capitalising on the opportunity to maximise its eventual reward.

V. Further work

In order to fully implement the planning algorithm described in this paper, there are several research questions which we intend to address. The most important of these being *how do we decide when to replan?* The choice of when to replan is important because planning is an expensive task that is not guaranteed to result in an improvement to the current solution. Russell and Wefald [9] state that 'typically there will be some time at which the optimal agent should stop deliberating and carry out an action' but this time is difficult to predict and the benefits of deliberation hard to quantify in advance. It is entirely possible that replanning may result in the generation of a plan which is identical to the previous one. If the vehicle replans too rarely it may miss valuable opportunities or contingencies, whilst replanning too frequently is likely to be inefficient and wasteful. Having shown that changing the plan during execution is beneficial, we now intend to experiment further with the AUV domain, attempting to represent the costs of replanning in order to optimise the cost-benefit ratio. One way of doing this might be to consider replanning as an action in its own right. This would allow the agent to reason about the value of the decision making process and to decide whether the benefit-cost ratio of further deliberation outweighs that of immediate action.

In addition we wish to investigate *what can be precomputed, prior to the start of the mission, to aid the replanning process*. It may be possible to reduce the cost of replanning by using information from the current plan to refine future plans or by reusing key elements from previous policies; for example by pruning sequences of actions which result in no reward (such as repeatedly surfacing and diving) or by repeating behaviours which are likely to end in a reward (such as transmitting data which is currently onboard the vehicle). By reducing the cost of replanning, the vehicle will be able to change plan more often, reacting quickly to unexpected situations or changes in the environment.

VI. Conclusions

As AUV missions increase to many days or weeks in length, the level of uncertainty surrounding vehicle state and resource usage will also increase. Although current fixed-plan approaches are designed to minimise risk to the vehicle, these behaviours are inevitably based on overly-conservative estimates of resource usage. By following such a plan for a long period of time through an area about which we have little prior knowledge, the vehicle may potentially miss valuable data collection opportunities. Although simply reducing these resource usage estimates may facilitate additional data collection, doing so would greatly increase the risk of losing or damaging the vehicle. In order to prioritise the safety of the vehicle, whilst maximising the collection of scientific data, we are developing a dynamic planning algorithm which monitors and refines the plan during the mission, adding contingencies to handle unexpected situations. In this paper, we have defined the AUV problem domain, shown the benefits of changing the plan during mission execution, presented an overview of the planning algorithm and discussed our ongoing research.

References

[1] M. Pebody, "The Contribution of Scripted Command Sequences and Low Level Control Behaviours to Autonomous Underwater Vehicle Control Systems and TheirIimpact on Reliability and Mission Success," in *Proceedings of OCEANS – Europe*, June 2007, pp. 1–5.

[2] G. Griffiths, "Autosub-Long Range - A Deep-Diving Long-Range Autonomous Underwater Vehicle," 2009.

[3] S. Russell and P. Norvig, *Artificial Intelligence, A Modern Approach*, 2nd ed. Prentice Hall, 2003.

[4] J. Benton, M. B. Do, and S. Kambhampati, "Over-Subscription Planning with Numeric Goals," in *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, ser. IJCAI'05. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005, pp. 1207–1213.

[5] J. Bresina, R. Dearden, N. Meuleau, S. Ramakrishnan, D. Smith, and R. Washington, "Planning Under Continuous Time and Resource Uncertainty: A Challenge for AI," in *In Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 2002, pp. 77–84.

[6] J. Koehler, "Planning under Resource Constraints," in *Proceedings of the Thirteenth European Conference on Artificial Intelligence ECAI98*, vol. 65. John Wiley & Sons, December 1998, pp. 489–493.

[7] A. M. Coddington, "Motivations for MADbot: a Motivated and Goal Directed Robot," *Proceedings of the 25th Workshop of the UK Planning and Scheduling Special Interest Group (PlanSIG 2006)*, pp. 39–46, 2006.

[8] Z. Feng, R. Dearden, N. Meuleau, and R. Washington, "Dynamic Programming for Structured Continuous Markov Decision Problems," in *Proc. of the 20th Conference on Uncertainty in Artificial Intelligence*. Arlington, Virginia, United States: AUAI Press, 2004, pp. 154–161.

[9] S. Russell and E. Wefald, "Principles of Metareasoning," in *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1989, pp. 400–411.