An Onboard Mission Replanning System for Autonomous Underwater Vehicles

Torsten Pfuetzenreuter

Abstract— **This paper presents the architecture of a hybrid mission replanning system for an autonomous underwater vehicle (AUV). Limited communication capabilities prevent a deep diving vehicle from asking a human operator for new commands in case of unexpected events. That's why mission replanning is an important part of the software architecture of an AUV. It reacts on those events with a modification of the vehicle's mission plan. The system consists of the functional modules Mission Monitoring and Mission Replanning. While Mission Monitoring observes of the actual conditions and initiates a plan adaption Mission Replanning as the central part of the system realizes the necessary modifications and is able to optimize a mission plan regarding objective criteria. The additional module Chart Server checks a replanned mission against a digital chart and proposes adaptions if there are violations of terrain constraints. The mission replanning system will be used aboard a German AUV called *DeepC*. However, its usability is not limited to AUVs, other autonomous vehicles with predefined mission plans are also possible candidates.**

Keywords— **Autonomous Underwater Vehicle, Mission Replanning, Mission Management.**

I. Introduction

For many years AUV technology was intensively studied all over the world. Many scientific vehicles as well as various software architectures and algorithms were developed to proof the technical feasibility of unmanned underwater vehicles. In contrast to land, air or space vehicles they only have limited communication possibilities and act in a very hostile environment. This requires reliable hardware and intelligent software algorithms for autonomous operation of AUVs.

AUVs have a set of advantages opposite to remotely operated vehicles (ROVs) used so far. A support vessel is only needed for launch and recovery, or if an AUV is launched by helicopter or from shore, no vessel is needed at all. Tasks like route surveys and inspections can then be accomplished with lower costs. Using multiple vehicles one vessel may explore a greater area of the oceans by launching them successively at different positions. During mission duration the vessel could accomplish other tasks. This list of advantages could be continued still further.

Observing the evolution from proof of concept to the first commercial vehicles in the last few years the industry shows increasing interest in development of those vehicles. The predicted demand for AUVs in the next years generates manifold activities to participate in the expected success of AUVs all over the world. In Germany an interdisciplinary consortium consisting of industrial and scientific partners has been established to develop the versatile AUV *DeepC*

T. Pfuetzenreuter is with the Institute of Automation and System Engineering, Ilmenau Technical University, Ilmenau, Germany. E-mail: torsten.pfuetzenreuter@tu-ilmenau.de .

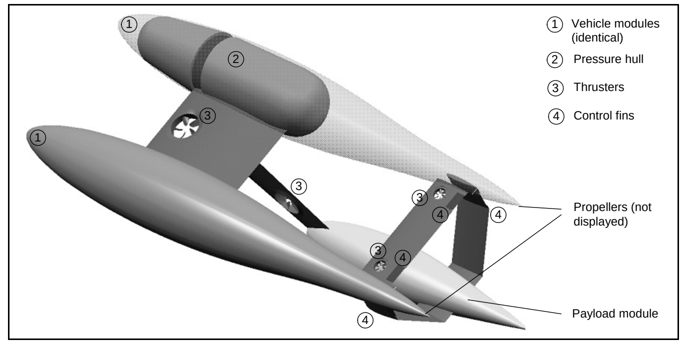

Fig. 1. Design of AUV *DeepC*

[1]. The project is funded by the German Federal Ministry for Education and Research (BMBF). *DeepC* represents a deep diving, long-range vehicle with several outstanding properties (Figure 1). Exemplary only some of them should be mentioned:

- the modular three body design allows a flexible payload interface,
- the maximum diving depth of 4000 meters allows explorations of wide parts of the oceans,
- two identical vehicle modules guarantee reliability by redundancy of all important systems (e.g. power generation, computer system),
- a polymer electrolyte membrane (PEM) fuel cell generates enough energy for long-term missions up to 40 hours,
- different software modules facilitate the autonomous operation of the vehicle (e.g. highly accurate long-term navigation, autonomous obstacle recognition and avoidance, system diagnostics system, case-sensitive track control, situation-adaptive vehicle controller, high-level mission management system with replanning capabilities).

The whole development process of the vehicle is attended by an abstract design description and overall simulation called Mission Level Design [2].

This paper deals with an important part of the software architecture: Mission replanning is an essential property of an AUV to react reasonable on modified conditions. An adequate reaction is one of the fundamental tasks of the AUV which are characterized by [3] in the following manner:

- The AUV *must* complete its mission successfully.
- It *must* cope with its environment without failure.
- It *must* maintain reliable on-board system performance.

These three assignments have to be considered in the corresponding software modules to be prepared for unanticipated situations. In the past with its short-term scientific vehicles their significance was not very great, today deep-diving long-term AUVs require paying more attention to

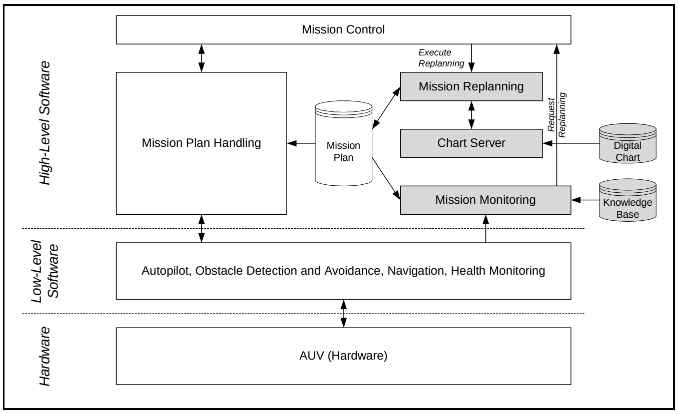

Fig. 2. Software architecture of the AUV, only the high-level modules are displayed separately

this tasks. Worldwide different institutions are involved in refinement of these autonomous features. For instance, US Navy's Autonomous Operations Future Naval Capabilities effort is an actual project to increase the autonomous capabilities of unmanned vehicles [4]. One of the first priority tasks is the 'Autonomous path and mission replanning for Unmanned Underwater Vehicles'.

In the following sections the Mission Management System (MMS) of an AUV and especially the Mission Replanning System (MRS) as an important component for handling unanticipated situations will be presented. MRS uses a rule-based expert system for monitoring the mission execution and combines different techniques for mission plan modification.

II. Software architecture of the Mission Management System

The MMS is the high-level part of the software architecture of the AUV (Figure 2). It contains the following modules:

- **Mission Control:** acts as the 'Captain' of the vehicle and must be asked before any action of the MMS takes place,
- **Mission Plan Handling:** realizes a mission plan (a list of maneuvers),
- **Mission Monitoring:** observes the mission plan execution, the actual vehicle condition as well as the environment situation and, if necessary, recommends a mission replanning,
- **Mission Replanning:** replans the mission plan considering all available information.

MMS sends base maneuvers to the low-level software modules that are responsible for the realization.

The MRS consists of the modules *Mission Monitoring* and *Mission Replanning*. Additionally a module called *Chart Server* will be implemented. The Chart Server is responsible for the final check of a replanned mission plan against terrain data and a possibly necessary modification of particular maneuvers.

Mission Monitoring acts as observer of mission execution. The associated knowledge base contains a set of rules whose actions generate concrete replanning instructions. These instructions are the input of Mission Replanning and have a general format which allows the use of different knowledge bases for different missions without modification of the Mission Replanning module.

Mission Replanning executes the instructions one by one and checks the resulting plan for energy and time restrictions as well as terrain violations with help of the Chart Server.

The mission plan consists of a series of complex maneuvers (Meander, GPS-Update, Track, Descent, Surface, etc.). This increases the complexity of replanning algorithms compared to simpler plans. A mission management system for a mission plan consisting of simpler mission points is described in [5].

III. Mission Monitoring

As stated in the previous section the Mission Monitoring module consists of a rule-based expert system. Basic components of the expert system are a knowledge base containing the rules, a fact list with the actual mission and vehicle data as well as a inference engine for executing activated rules. Further information about expert systems could be found in [6].

The activation of one or more rules results in a plan modification. For instance, if the payload sensor of the vehicle is defective, all maneuvers that use this sensor are deleted from the mission plan. And a failure of important hard- or software modules creates a surfacing maneuver, which replaces all remaining maneuvers.

The knowledge base is subdivided into several parts that are completely separated from each other: mission-independent rules handling general problems and mission-dependent rules for special missions. This subdivision has several advantages. First, due to the separation, tested parts of the knowledge base are not influenced by changing one or more rules of other parts. Second, future enhancements of the MMS, for instance by including machine learning methods creating rules for special situations, are supported by constructing a new rule set. Errors and wrong decisions made out of these rules do not endanger the vehicle since the high prior mission-independent rules guarantee a safe termination of the mission.

The mission-independent rules are among others responsible for the following tasks:

- check energy and time consumption (comparison with the mission plan),
- observe general vehicle sensors and actors (aggregated output of the module Health Monitoring),
- compare the navigation data with the plan-defined tolerance limits.

The mission-dependent rule set contains the rules that may change from mission to mission. At first the AUV will work on survey and cable tracking missions. Particularly cable tracking requires a series of rules. For example, if the tracking sensor detects a cable lost, a search pattern must be executed at the last valid position.

IV. Mission Replanning

Mission Replanning modifies the actual mission plan by executing the replanning instructions from Mission Monitoring. In this section some basic definitions and the methods used for modifying a mission plan are presented.

A. Definitions

A mission plan is - in generalized sense of the Artificial Intelligence - a total-ordered plan. The steps of a plan here are called maneuvers, every maneuver contains a start position $\mathbf{x}_s$ as precondition, an action t as well as an end position $\mathbf{x}_e$ as effect. A shift from one maneuver to the next in the plan happens if the precondition of the new maneuver is fulfilled. Naturally the effect of a plan step is the same as the precondition of its successor:

$$x_s(i+1) = x_e(i) \qquad \forall i = 1, 2 \ldots, n-1 \tag{1}$$

with n as the total number of maneuvers. Some additional properties are assigned to each maneuver (see Table I). These properties are needed for optimization of the mission plan due to energy or time problems.

An important attribute is the weight of a maneuver as profit measurement, which is computed using universal, objective criteria. And this is important for domain-independent application of the developed methods.

Property	Variable	Description
Parameters	$\mathbf{p}(t)$	vehicle speed, active sensor components, etc.
Costs	$c = f(\mathbf{x}_s, \mathbf{x}_e, \mathbf{p})$	energy consumption, maneuver duration, etc.
Weight	$w = f(\mathbf{x}_s, \mathbf{x}_e, \mathbf{p})$	measure for contribution to mission profit

TABLE I

Properties of maneuvers needed for replanning

B. Replanning instructions

To modify a mission plan five basic instructions are necessary (shown in Figure 3):

- **Insert:** inserts a new maneuver into the mission plan; two different insertion methods are to distinguish: *Insert into* a existing plan step and *Insert after* a denominated plan step,
- **Delete:** deletes a maneuver from a mission plan,
- **Modify:** modifies an existing maneuver of a mission plan by tuning the parameters $\mathbf{p}(t)$,
- **Optimize:** optimizes a mission plan in consideration of the actual conditions.
- **Abort:** aborts the mission.

This instruction set is sufficient to perform all needed plan modifications. Additional attributes are assigned to each command, e.g. the maneuver name for Insert, Delete and Modify or new parameter values for Modify. *Insert into* means that the original maneuver will be interrupted for execution and continued after completion of the inserted plan step. Erasing a maneuver from a plan requires a filling maneuver (a simple transit maneuver) to fulfill the continuity condition (equation (1)). If this results in successive transit maneuvers these are recombined to one joined transit.

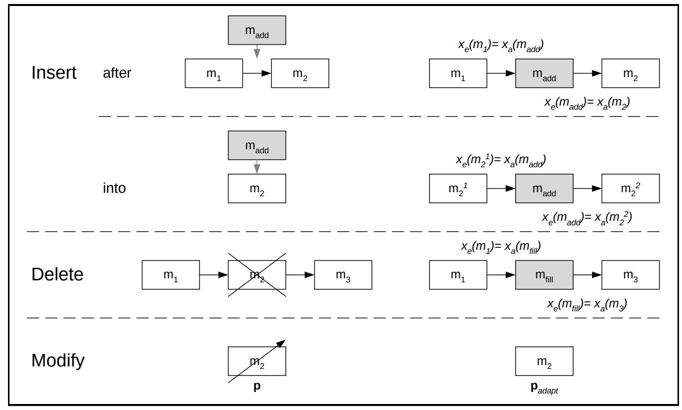

Fig. 3. Insert, Delete and Modify instructions for mission replanning

C. Mission plan optimization

Mission plan optimization takes place if the energy (or time) resources are not sufficient for the remaining maneuvers. Main goal is to generate secure conditions - the modified plan has to meet the actual resources. Beside this the method should try to maximize the mission profit. Therewith the objective function for the optimization is defined to be:

$$f(\mathbf{x}) = W(\mathbf{x}) = \sum_{i=1}^{n} w_i$$

where $\mathbf{x}$ is the vector of the optimization variables. Potential parameters for this problem are

- the vehicle speed - separately changeable for each maneuver - and
- the maneuver activity - a variable to disable a maneuver by setting its value to zero.

In the first approach only the activity is used as parameter. By changing the activity of one or more maneuvers a maneuver configuration is created out of the original mission plan. Every configuration is identified by its activity vector **act** containing the activities of all non-transit[1] maneuvers:

$$\mathbf{act} = [act_1, act_2 \ldots, act_m], \quad act_i \in \{0, 1\}$$

where m is the count of non-transit maneuvers. Analogous to the Delete replanning instruction resulting successive transit maneuvers are recombined to one joined transit. This may contribute to save energy by reducing the total distance covered during the mission.

[1] Changing the activity of a transit maneuver from 1 to 0 results in replacing this maneuver with a transit. This inflates the solution space without achieving a better solution.

The costs of a mission (the time and energy resources) act as constraints $\mathbf{g}(\mathbf{x})$ for the optimization:

$$\mathbf{g}(\mathbf{x}) = \sum_{i=1}^{n(\mathbf{X})} \mathbf{c}_i \; - \; \mathbf{c}_{max}$$

where $\mathbf{c}_{max}$ is the maximum permissible consumption and $n(\mathbf{x})$ the maneuver count of the actual configuration. With this definitions and using **act** as vector of variables the optimization problem could be written as follows:

$$\max \{ f(\mathbf{act}) : \mathbf{g}(\mathbf{act}) \leq 0 \}$$

Because of the nonlinear behavior of both cost and weight function the problem belongs to the class of constrained nonlinear optimization problems with integer variables. Due to the computation time constraints to find a solution classical and non-classical optimization methods don't seem to be suitable. That's why a task-adapted directed search method solving the problem in determined time was developed. This method computes a mission profit vector $\mathbf{W}$ using possible maneuver configurations, sorts it in descending order and determines the violation of the constraints. The first valid solution is called SOOP (**S**tep **O**ne **O**ptimal **P**lan) because the described algorithm is the first step of the optimization. This is the best result if only binary activity values are allowed ($act \in \{0, 1\}$).

If the maneuver activity is not fixed to this set but is an element of the closed interval $[0, 1]$ an improvement of the SOOP can be achieved. Having such an activity value a maneuver will be partial executed - from the beginning to a break point defined by the activity. In the second step of the optimization the disabled maneuvers of the SOOP are considered: A further increase of the mission profit is reachable if one or more of them could be partial executed (with the constraints in mind). This plan derived from the SOOP is called STOP (**S**tep **T**wo **O**ptimal **P**lan). Mission Monitoring is responsible for canceling the partial performed maneuver if the percental completion reaches the predefined activity value. The plan execution will be continued with a transit to the subsequent maneuver.

The figure 4 shows an example of the optimization. The original mission plan consists of a set of maneuvers labeled with their types. In this case the energy resources of the AUV are too low to realize the complete plan. The first optimization step creating the SOOP deletes two maneuvers from the mission plan (Track and GPS-Update) and replaces them with a Transit. Then the resulting two successive Transit maneuvers are recombined to one single Transit (dash-dotted line).

In the second optimization step the weights of the two deactivated maneuvers are examined. As consequence the Track maneuver will be partial executed (dashed line).

D. Plan check

After all modifications have taken place, the new mission plan will be verified for correctness and violations of terrain constraints. The correctness check ensures that the Mission Plan Handling module can load the adapted mission plan.

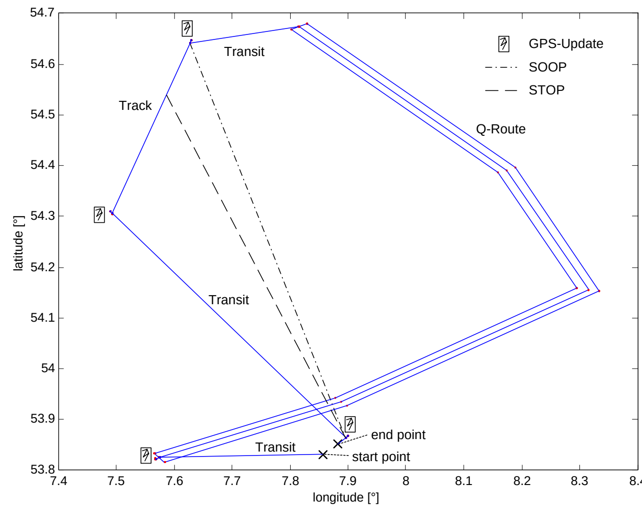

Fig. 4. Plan optimization example

The Chart Server determines potential collisions with the terrain and known obstacles utilizing line-of-sight computations for every maneuver. This module uses digital charts containing regular or irregular spaced data points and creates a corrected mission plan as proposal for Mission Replanning if necessary.

V. Conclusions

A new architecture for a mission replanning system was presented. This system is developed to be as far as possible universal regarding different vehicles; most of the statements and methods are valid for the whole class of autonomous systems. However, a test for instance with a wheeled robot requires modifications due to an other mission plan (no descent and surface maneuvers, two dimensional way points) as well as different sensors and actors. Adaptions for a class of land robots are planned for the near future.

First tests with a complete AUV simulator will be done in summer 2003. The goal is to prepare *DeepC* for first in-water experiments in fall 2003.

References

[1] DeepC, "Homepage," http://www.deepc-auv.de, jul 2002.

[2] T. Liebezeit and V. Zerbe, "Mission level design of autonomous underwater vehicles," in *First International ICSC Congress on Autonomous Intelligent Systems*, Deakin University, Geelong, Australia, feb 2002.

[3] A. S. Westneat and W. L. Clearwaters, "A generalized contingency alternative matrix for unmanned underwater vehicles," in *Proceedings of the IEEE Symposium on Autonomous Underwater Vehicle Technology*, Washington, DC, jun 1990, pp. 29–33.

[4] Office of Naval Research US Navy, "A future naval capability: Autonomous operations," http://www.onr.navy.mil/media/, jul 2002.

[5] C. Barrouil and J. Lemaire, "Advanced real-time mission management for an AUV," in *NATO Symposium on Advanced Mission Management and System Integration Technologies for Improved Tactical Operations*, Florence, Italy, sep 1999.

[6] J. Giarratano and G. Riley, *Expert Systems - Principles and Programming*, PWS Publishing Company, Boston, USA, 1998.