Advanced Mission Management for Long-range Autonomous Underwater Vehicles

T. Pfuetzenreuter
Ilmenau Technical University
P.O. Box 100565
Ilmenau, 98684, Germany
torsten.pfuetzenreuter@tu-ilmenau.de

***Abstract*-Long-range autonomous underwater vehicles (AUVs) need a higher degree of autonomy than short-range vehicles. That results from the limited communication possibilities during the mission execution - AUVs are out of communication range of a supporting vessel almost the whole time. There is no way to transfer a new mission plan into to a vehicle or to command a surface maneuver.**

That's why the mission management system as the top level in the software architecture of an AUV is the key component for enabling complete autonomous operations. It has to observe the vehicle and environment conditions and to modify the mission plan given by an operator during mission preparation if the mission execution is endangered.

This paper presents a hybrid mission management system consisting of several modules responsible for different tasks. Mission Monitoring observes the execution of the mission plan and initiates a replanning if necessary. Mission Replanning is responsible for insertion, deletion and optimization of the involved maneuvers. As third module, the Chart Server checks the modified plan against a digital chart and modifies a route if a ground collision would occur.

Preliminary simulation results of the optimization and the other modification methods conclude the paper.

I. INTRODUCTION

Today AUVs are the most developed technology to explore the oceans. In contrast to aerial or space vehicles they need a very advanced control system since an intervention of a human operator is only possible during short contact phases. Most of the time of mission execution an AUV is out of communication range – the onboard software has to handle all unanticipated events for itself.

On the other side the exploration of the oceans, for instance for high resolution digital mapping of the sea floor in preparation for cable laying, requires deep-diving vehicles since the most interesting areas for this are found in greater depth. However, a diving depth of 2000 meters or more requires adapted hardware components (pressure hull, energy storage). Additionally the energy generation system should be able to produce enough energy for mission durations of 24 hours or more to increase the efficiency of mission execution.

To address these very specific technical challenges a German interdisciplinary consortium consisting of industrial and scientific partners has been established to develop the versatile AUV *DeepC* [1]. The team consists of STN ATLAS Elektronik GmbH (Bremen), OSAE-Offshore Survey and Engineering GmbH (Bremen), ATI Küste GmbH (Rostock), ELTA Engineering GmbH (Rostock), AIR Fertigungtechnologie GmbH (Rostock), Zentrum für Sonnenenergie- und Wasserstoff-Forschung (center for solar energy and hydrogen research, Ulm), the Ilmenau Technical University and the University of Karlsruhe.

Primary aim of the project, which is funded by the German Federal Ministry for Education and Research (BMBF), is to develop a deep-diving, long-range autonomous underwater vehicle with several outstanding features (Fig. 1):

- modular design with a flexible payload interface,
- two identical vehicle modules providing reliability by redundancy of important systems (e.g. power generation, computer system),
- two polymer electrolyte membrane (PEM) fuel cells generate energy for long-term missions up to 40 hours,
- an advanced software architecture allows autonomous operation of the vehicle (obstacle detection and avoidance, system diagnostics, mission monitoring and replanning).

The development process of the vehicle is attended by an abstract design description and overall simulation called Mission Level Design [2]. It combines generalized models of hard- and software components and therefore helps to find problems in early design stages of the vehicle.

II. SOFTWARE ARCHITECTURE

The software architecture of this vehicle orientates on the Rational Behavior Model developed by Byrnes et al. [3]. It represents a hierarchical, multi layer intelligent control architecture for autonomous vehicles consisting of three levels (strategic, tactical and execution, (Fig. 2)).

The execution level contains software modules directly connecting to vehicle hardware (sensors and actors) while the top layer (strategic) consists of software acting on the vehicle's mission plan. All other modules are located in the middle level (tactical).

In this paper only the tactical level of the software is discussed. It will be called Mission Management System (MMS). One important part of the MMS is the Mission Replanning System (MRS). Its primary task is to modify the vehicle's mission plan, e.g. as result of unanticipated events. The MRS uses a rule-based expert system for monitoring the mission execution and combines different techniques for mission plan modification.

0-933957-31-9

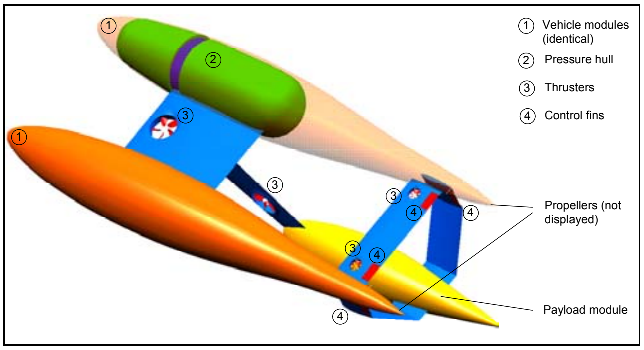

Fig. 1 Design of AUV DeepC

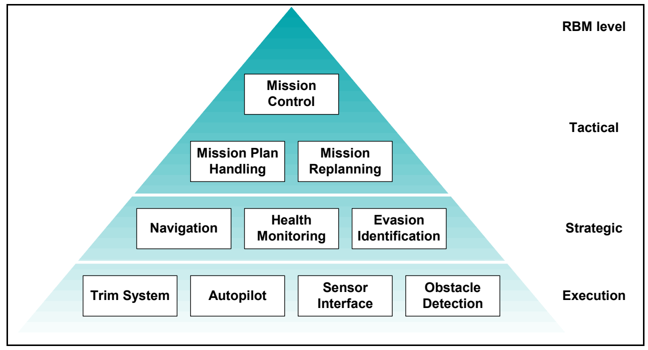

Fig. 2 Software architecture of DeepC with selected software modules

A. *Software architecture of the MMS*

The MMS consists of the following modules:

- Mission Control: the overall deciding instance of the vehicle,
- Mission Plan Handling: responsible for sequential realization of the maneuvers contained in the mission plan,
- Mission Monitoring: monitors the maneuver execution and requests a replanning if necessary,
- Mission Replanning: executes replanning procedures,
- Chart Server: checks modified mission plans against digital terrain data and proposes an alternative way if the planned one is nonnavigable.

The modules Mission Monitoring, Mission Replanning and Chart Server form the MRS. Fig. 3 shows the basic mode of operation of the MRS.

Mission Monitoring observes the mission execution by usage of a rule-based expert system. If a replanning is necessary, the monitoring module generates concrete replanning instructions out of the rules in the knowledge base and requests a replanning operation. Mission Control decides on realization of the replanning using the actual situation and information from the mission plan.

Mission Replanning receives the replanning instructions and executes them one by one. After modification the new mission plan is checked for correctness (readability by Mission Plan Handling) and violations of the digital terrain using the Chart Server.

The Chart Server splits the mission plan into basic maneuvers and determines potential collisions with the digital terrain. If possible, a proposal for a navigable route is generated in case of nonnavigable way segments.

B. *Structure of mission plans*

Mission plans are defined as a sequence of complex maneuvers from a maneuver catalog (Table I). Every maneuver is specialized for a specific task and has parameters describing the behavior of the AUV during maneuver execution. For the MRS the following are the most important parameters:

- start and end position,
- start and end speed,
- maneuver priority,
- admissible CEP (circular error probable),
- active components (payload, cameras, lights).

Track maneuvers without active components except of the vehicle's standard sensors and actuators are called transit maneuvers since they only connect their start and end points.

Using complex maneuvers for the composition of a mission plan simplifies the work of an operator during mission preparation, but it increases also the complexity of replanning algorithms (compared to straightforward plans). A mission management system using simpler mission points (one three-dimensional position, importance, procedure to be executed on arrival at the desired position) is described in [4].

III. MISSION MONITORING

Mission Monitoring is basically formed by a rule-based expert system. Such a system consists of a knowledge base (containing all needed data and rules) and an inference engine (working on the rules). An introductional overview of expert systems is available in [5].

The knowledge base contains the rules to be executed, object instances holding the actual vehicle state, environmental information, the mission plan as well as facts generated during rule firing. The facts represent the internal state of the expert system.

TABLE I
MANEUVER CATALOG OF AUV DEEPC

Name	Description
Track	Strait line between start and end point
Meander	Meandering search pattern
Q-Route	Spiral search pattern
GPS Update	Combination of different complex maneuvers (surface maneuver, GPS position fix, descent maneuver)
Surface	Two phase surface maneuver
Descent	Two phase descent maneuver
Circling	Circling maneuver around a center point for a definite time

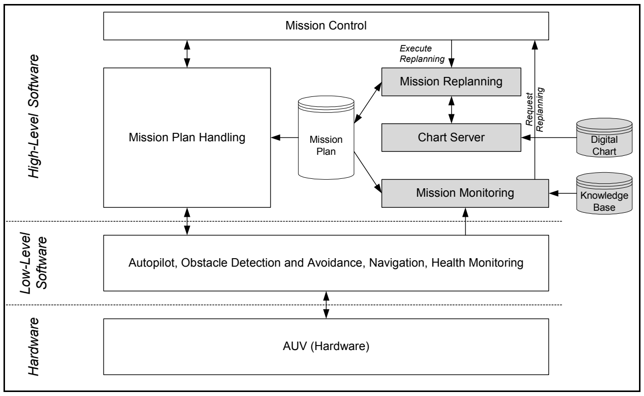

Fig. 3 Mode of operation of the MRS

During a monitoring cycle the inference engine determines the rules to be fired and executes them depending on their priority. As result one or more replanning instructions are created inside the rules. If, for instance, the payload of the vehicle is defective, the associated rule will create "Delete" instructions for all maneuvers needing the payload of the vehicle. Another example is the insertion of a GPS update in case of a too large position error estimated by the navigation module.

The rules in the knowledge base are arranged in several modules that are completely separated from each other. The mission-independent part contains rules dealing with general problems (energy and time restrictions, vehicle hardware); a mission-dependent module aggregates rules for special mission types.

Subdividing the rule set has several advantages. Due to the separation, adding or modifying rules of other modules of the knowledge base does not influence tested and accepted modules. Furthermore, future enhancements, for instance by introducing machine learning methods for creation of rules out of experienced situations, are supported by adding new modules. Errors in this rule sets do not endanger the vehicle since the high prior mission-independent rules guarantee a safe termination of the mission.

The mission-independent rules accomplish general tasks required in every mission, for instance:

- check the energy and time consumption to reach the surface and for the remaining maneuvers,
- observe vehicle sensors and actors,
- compare estimated navigation data with the values defined in the mission plan.

The mission-dependent module contains rules that may change from mission to mission. The first tasks of the AUV are route survey and cable inspection missions. Especially the cable tracking requires rules for a successful mission execution. If, for example, the tracking sensors lost the cable to inspect, a search pattern hast to be executed at the last valid position.

IV. MISSION REPLANNING

Mission Replanning uses the replanning instructions generated by Mission Monitoring for the modification of a mission plan. This section presents some definitions, the structure of the replanning instructions and the mission plan optimization.

A. Definitions

Most of the information about planning and replanning comes out of the artificial intelligence (AI) since planning problems play an important role in this science. Using the same terms facilitates understanding and propagation to other problem domains; this section attempts to create a mapping of the various names.

The mission plans of the AUV correspondent to total-ordered plans in AI. Plans consist of steps, these steps are the maneuvers contained in the mission plan. Every step has associated attributes; Table II shows the variables and descriptions of them.

A shift from one step to the next in the plan happens if the effect of the active step is fulfilled. For the AUV it is important that start and end positions of successive maneuvers are equal. This is called the continuity condition:

$$\mathrm{x}_s\left(i+1\right)=\mathrm{x}_e\left(i\right) \quad \forall i=1,2,\ldots,n-1 \tag{4.1}$$

with n as the total number of maneuvers.

The attributes cost and weight in Table II are important for the optimization of the mission plan. The weight is a measurement of the maneuver profit computed using different criteria (e.g. priority, relative distance) while the cost vector represents the resources needed by the maneuver.

B. Replanning instructions

Replanning instructions form the interface between Mission Monitoring and Mission Replanning. For modification of mission plans three basic commands are sufficient (Fig. 4):

- Insert: inserts a maneuver into the mission plan,
- Delete: deletes a maneuver from the mission plan,
- Modify: alters parameters of a maneuver.

Inserting a maneuver into the plan requires an additional flag distinguishing two insertion methods. Insert into

TABLE II
ATTRIBUTES OF MANEUVERS

Attributes	Variable	Description
Precondition	$\mathbf{x}_s$	start position
Action	t	depending on type of maneuver
Effect	$\mathbf{x}_e$	end position
Parameters	$\mathbf{p}$	vehicle speed, active components, etc.
Costs	$\mathbf{c} = f(\mathbf{x}_s, \mathbf{x}_e, \mathbf{p})$	energy consumption, maneuver duration
Weight	$\mathrm{w} = f(\mathbf{x}_s, \mathbf{x}_e, \mathbf{p})$	measure for maneuver profit

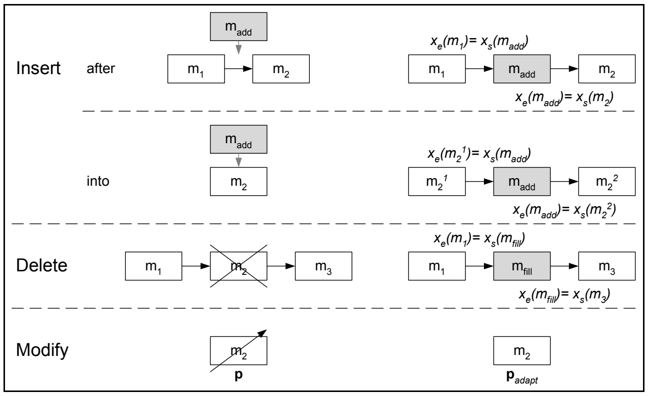

Fig. 4 Insert, Delete and Modify instructions

means that the actual maneuver will be interrupted and continued after execution of the inserted one. Insert after will integrate the new maneuver after finalization of the referenced maneuver.

The deletion of a maneuver requires a fill maneuver (a transit maneuver) since the continuity condition (4.1) has to be fulfilled. If this results in successive transit maneuvers these are recombined to one joined transit.

Beyond these instructions two others are defined:

- Abort: aborts the mission, a transit to a defined recovery position could be optionally planned,
- Optimize: optimizes the mission plan considering the actual situation.

Each instruction may have attributes defining its behavior, e.g. the name of the maneuver to be processed or new parameter values for Modify.

C. *Optimization of mission plans*

Mission Monitoring initiates the optimization of a mission plan if the energy or time resources are not sufficient to complete the whole plan. In this case a low-prior optimization task is started to modify the mission plan. If concurrently other replanning commands have to be realized the optimization will be aborted and restarted later since a (for the energy or time balance possibly positive) plan modification may change the initial situation of the optimization.

Main goal of the mission plan optimization is to create a plan meeting the actual resources. Beside this it should maximize the profit of the whole mission. The optimization uses as objective function the sum of all maneuver weights

$$f(\mathrm{x}) = W(\mathrm{x}) = \sum_{i=1}^{n} w_i \tag{4.2}$$

with $\mathbf{x}$ as the vector of optimization variables, possible candidates for the presented problem are:

- the vehicle speed – individually adjustable for every maneuver and
- the activity of maneuvers – allows the deactivation of maneuvers in the plan.

In the first version of the optimization method only the activity is used as optimization variable. This has several advantages.

- The activity of a maneuver is limited to the values {0, 1} so the optimization belongs to the class of combinatorial optimization problems [6]. These are solvable by robust methods.
- Using the speed of the AUV as second optimization variable results in a mixed integer programming problem. The time needed to solve such a task is not predictable - this is not acceptable with regard to the security of the vehicle.

Utilizing the activity of the maneuvers, i.e. $\mathbf{x} = \mathbf{act}$, creates a set of different mission plans called maneuver configurations. These configurations differ by one or more activity values and are characterized by the activity vector **act** containing the activities of all non-transit maneuvers:

$$\mathrm{act} = \left[act_1, act_2, \ldots, act_m\right] \quad act_i \in \left\{0,1\right\} \tag{4.3}$$

where m is the count of non-transit maneuvers. The transit maneuvers are ignored since the deactivation of such a maneuver results in replacing it with a transit (similar to deleting it). This would inflate the solution space without achieving a better solution.

If the deactivation of maneuvers results in successive transit maneuvers these are recombined to reduce energy and time consumption (Fig. 5).

The costs of a mission, i.e. the energy and time needed to perform the given tasks, act as constraints **g**(**act**) for the optimization problem

$$\mathrm{g}(\mathrm{act}) = \sum_{i=1}^{n(\mathrm{act})} \mathrm{c}_i - \mathrm{c}_{\max} \tag{4.4}$$

where $\mathbf{c}_{\max}$ is the maximum admissible resource consumption and $n(\mathbf{act})$ is the maneuver count of the actual configuration. Using the given definitions the optimization problem could be written as follows:

$$\max\left\{f(\mathrm{act}) : \mathrm{g}(\mathrm{act}) \leq 0\right\}. \tag{4.5}$$

For solving this problem a task-adapted, two-step optimization method was developed (Fig. 6). In the first step all possible maneuver configurations are enumerated. There exist 2^m different combinations of activity values,

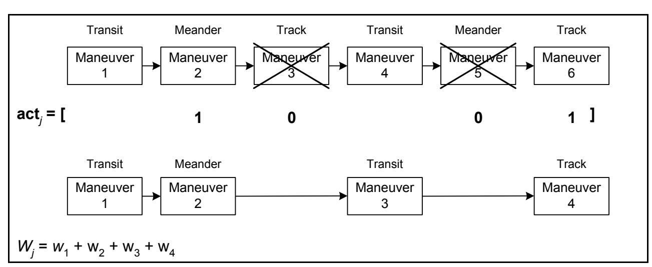

Fig. 5 Creation of a maneuver configuration

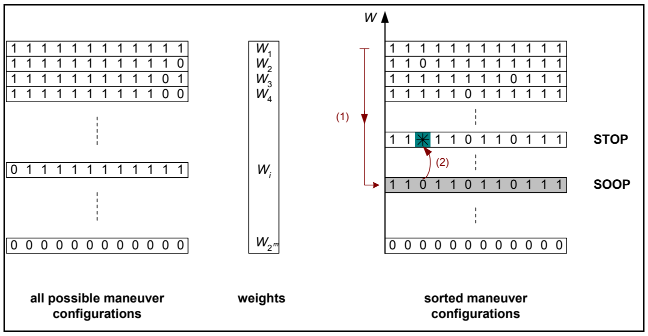

Fig. 6 Optimization of mission plans

but for a small amount of remaining non-transit maneuvers the solution will be found in acceptable time.

For every configuration the mission profit will be computed, for all configurations this forms the mission profit vector W. If this vector is sorted in descending order, the task reduces to check the constraints of the optimization problem sequentially. The first configuration fulfilling the constraints is the best solution of this step and will be called SOOP (Step One Optimal Plan).

If the maneuver activity is not fixed to integer values but is an element of the closed interval [0,1] the solution could be improved. A maneuver having such an activity value will be partially executed – from the start position to a termination point defined by the activity value.

The second step of the optimization examines the deactivated maneuvers of the SOOP. If it is possible to realize one (or more) of them partially with the constraints in mind, the mission profit can be increased. The originated plan is called STOP (Step Two Optimal Plan). Mission Monitoring is responsible for terminating the partial performed maneuver if its percental completion reaches the activity value. The mission will be continued with a transit to the subsequent maneuver.

Fig. 7 shows an example for the optimization of a mission plan. The maneuvers of the original plan are labeled with their types; Track maneuvers, which represent transits, are labeled accordingly. For this example the energy resources of the AUV are too low to realize the entire plan so the optimization has to reduce the covered distance. The first optimization step deletes three maneuvers from the mission plan (Track, GPS update, Track) and inserts a transit to the subsequent maneuver (labeled as SOOP).

Examining the weights of the deactivated maneuvers in the second step leads to the STOP, the first Track maneuver will be partial executed.

D. *Plan check*

After applying all changes the mission plan has to be checked for correctness and terrain violations (using the Chart Server).

The correctness check is necessary to ensure that Mission Plan Handling can load the modified mission plan. If the generated plan is error-free it will be delivered for execution, in the other case an inspection of the replanning reason decides on rejecting the changes of the mission plan or creating an emergency plan.

V. CHART SERVER

The Chart Server is responsible for checking modified mission plans against digital terrain data and, if necessary, for modifying maneuvers violating the terrain constraints.

A. *Mission plan check*

The check of a mission plan for touching or crossing the triangulated terrain requires a preprocessing of all maneuvers of this plan. They are disassembled into base maneuvers (strait lines and arcs), which could be directly tested against the sea floor.

The region covered by the mission has to be provided as set of terrain tiles. Every tile contains three-dimensional data points and their triangulation forming a Triangular Irregular Network (TIN) (Fig. 8). A fast computation of intersections between lines and covered triangles detects nonnavigable maneuvers. Result of the check is a list of intersection points.

B. *Mission plan modification*

If one ore more of the maneuvers of the plan violate terrain constraints, a mission plan correction using the information of the digital terrain is initiated. The first version of the Chart Server only supports the modification of Track maneuvers. This reduces the modification task to find a valid (and short) path from the start to the end position of the maneuver.

One of the most powerful methods for solving shortest path problems is the Dijkstra algorithm [7]. It uses a directed graph to find shortest path distances from a start node to all nodes in the graph; two nodes in the graph are joined if there exists a direct path between them.

For the Chart Server the nodes are the three-dimensional data points of the terrain tiles. All points, which are visible among each other, have a connection in the graph. The graph is implemented as a distance matrix containing horizontal distances for joined nodes and zero values for all other. For performance reasons this matrix is

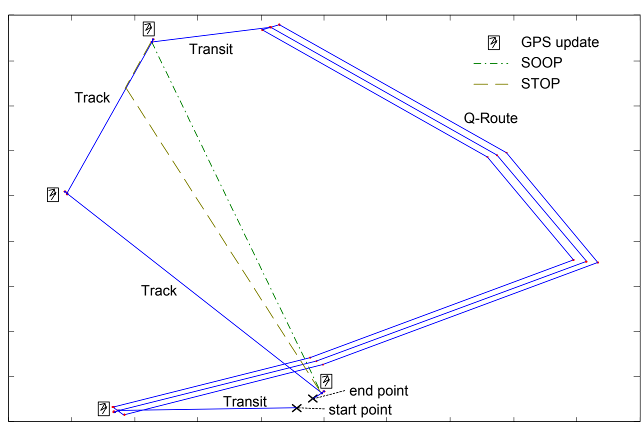

Fig. 7 Example for mission plan optimization

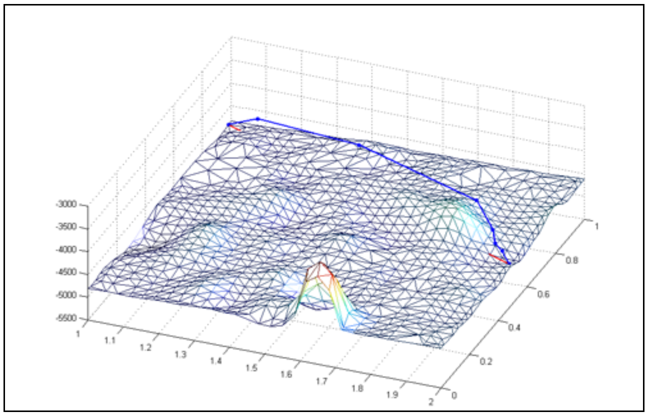

Fig. 8 Triangulated terrain with nonnavigable and modified route

precomputed and stored together with the triangulation. The vertical component of the distances is included during computation of the new route; this allows considering the height information of the two positions to be connected.

Result of the Dijkstra algorithm is a list of points from the start point to the destination position forming the shortest valid path between them. Final step of the plan modification is to smooth the height values of the found points (Fig. 8).

VI. CONCLUSIONS

A new architecture for advanced mission management, especially for long-range underwater vehicles has been presented. Using software modules for monitoring and modifying the mission plan of a vehicle during mission execution simplifies the work of the operator during mission preparation – there is no need to plan alternatives for a series of possible scenarios. First tests with the AUV simulator are planned for fall 2003 and with DeepC for spring 2004.

The system is developed to be universal as far as possible; most of the presented methods are transferable to other classes of autonomous vehicles using mission plans. However, testing it for instance with a land robot requires other monitoring rules and modifications due to mission plan differences (two-dimensional positions, other maneuvers). These adaptions for a class of small, wheeled robots are planned for the near future.

REFERENCES

[1] DeepC, Internet homepage, http://www.deepc-auv.de.

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

[3] R.B. Byrnes, A.J Healey, R.B. McGhee, M.L. Nelson and S.H. Kwak, "A Rational Behavior Model Software Architecture for Intelligent Ships: An Approach to Motion and Mission Control," Naval Engineers Journal, American Society of Naval Engineers, March 1996.

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

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

[6] K.G. Murty, "Operations research: deterministic optimization models," Prentice Hall, Englewood Cliffs, USA, 1995.

[7] T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, "Introduction to algorithms," MIT Press, Cambridge, USA, 2001.