DEVELOPMENT OF CENTRALIZED CONTROL SYSTEM FOR AUV GROUP

Igor Tuphanov[1,2]
ituphanov@gmail.com

Alexander Scherbatyuk[1,2]
scherba@marine.febras.ru

Yury Vaulin[1]
vaulin@marine.febras.ru

[1]Institute for Marine Technology Problems FEB RAS
Vladivostok, Russia

[2]Far Eastern Federal University
Vladivostok, Russia

Abstract— **The problem of AUV group control is considered in the paper. A group of AUVs could give a number of advantages in search operations or environment monitoring, especially if the group is coordinated according to an optimal plan. Some of those advantages are high performance and reliability during the mission. In order to arrange a group of AUVs it demands to develop and implement appropriate algorithms for optimal plan design. A group control system should monitor the mission execution and switch plan according to changing conditions, for example, if one of the vehicles leaves the mission due to an accident or for another task. We consider these problems within a specific control system for MARC AUV.**

Index Terms—**AUV, group operation, control system**

1. Introduction

One of the tools for solving survey tasks in the sea is AUV. There are problems with speed and reliability in case of using a single vehicle. One of the ways to overcome these problems is group AUV usage. It needs in specialized control systems for effective AUV group control. Such systems are being developed in various research institutions and described in papers.

In Underwater Systems and Technology Laboratory (LSTS) of University of Porto (Portugal) the work on software for collaborative control of heterogeneous autonomous vehicles is under way. Not only underwater, but also surface and aerial vehicles are considered [1]. A unified controlled system named DUNE [2] uses application message transmission protocol IMC [3] in communication between modules of the system in a single vehicle, as well as between vehicles. An operator console for simultaneous management of several vehicles is also developed. Each vehicle in DUNE system has a plan, which is either a motion trajectory or a command to follow another vehicle. The plan is determined by operator, so current version of the system doesn't support automatic plan construction and task allocation.

An approach with task allocation before mission start is used in [4]. Area coverage is considered as a single task. Ant colony optimization is used to build optimal plan.

One of open-source AUV control systems is MOOS-IvP. Several processes exchanging their data via shared memory are running simultaneously on the vehicle's onboard computer. A set of all possible vehicle actions is considered and one of them is chosen in each step, depending on joint work of several behaviors. Each behavior has its own utility function. Linear combination of utility functions for different behaviors is taken as common utility function. The action which gives maximum to this function is performed (on details for utility function representation and optimization techniques see [5]). Algorithms for group control are implemented as additional modules in MOOS-IvP. Collaborative work is suggested according to the following scheme. A central node is the source for all tasks (this could be operator console or one of AUVs, having the whole mission plan). "Point visiting" tasks are considered. As new tasks appear, the central node publishes them and takes bets from vehicles. A vehicle with minimal bet gets the task. This is an instance of auction planning. It can be implemented in MOOS-IvP as two modules: central node which publishes tasks and auctioneer on each vehicle, making bets. If an auctioneer won a bet, this new task influences on utility function of its AUV.

Hybrid architecture is used in many of current AUV control systems. The whole control is separated in to three layers:

- low level of control performs elementary actions; this level is the most close to onboard devices and works in a reactive way (responds local reactions on changing conditions);
- middle level breaks the plan in elementary actions and accumulates sensor data to provide it to upper level;
- upper level of control builds action plan according to mission and environment model.

Hybrid systems are, for example, DUNE [2], DSAAV [6] (National University of Singapore), control system used in IMTP FEB RAS [7]. Some control systems are based on other principles. An example of behavior-based control system is MOOS-IvP [8]. TRex system [9] provides control based on hierarchical structure on perception–model–plan–act scheme.

This paper is devoted to the development of hybrid control system, supporting group AUV operations on centralized control scheme. This scheme is used on two reasons. First, it does not demands any negotiation procedures among AUVs, as do action planning methods in distributed systems. This reduces total number of transmitted messages in very noisy and slow marine acoustical environment. Second, distribution of control gets effective for big number of vehicles but we use a small AUV group.

2. Some Algorithms of AUV Centralized Planning

Generally the planning process consists of some parts. First of all, the goal should be decomposed, i.e. broken on several tasks. Then a sequence of tasks should be assigned to each robot. We will use the principle of tasks indivisibility: each task should be executed continuously, and each AUV can't start a new task if the previous one isn't finished.

Among all of the plans, the one should be chosen which minimizes some functional (for example, total run of all vehicles). While an optimization problem formulation, we should also consider the fact, that each task can be accomplished by one of several ways. Each task is characterized with estimated execution time and with several ways to accomplish it, which can have different starting and ending points. Finally, the plan execution should be controlled and corrected in case of emergency.

Suppose we have m vehicles and n tasks. Initially, q-th vehicle is located at point $\mathbf{s_q}$ of 3D-space. It is known v_i ways to perform the task number i. Further, the j-th way to perform the task number i is characterized by start point $\mathbf{a_{ij}}$, end point $\mathbf{b_{ij}}$, and the execution time l_{ij}.

Let's denote as a plan of single vehicle such cortege of pairs

$p = (i_1, j_1), (i_2, j_2), ..., (i_{|p|}, j_{|p|})$, that $i_k \in 1..n$, $j_k \in 1..v_{i_k}$ for all $k \in 1..|p|$. The execution of the plan for q-th vehicle starts at the point $\mathbf{s_q}$. The vehicle at first goes to the point $\mathbf{a_{i_1 j_1}}$ and executes the j_1-th variant of i_1-th task. Then the vehicle goes to the point $\mathbf{a_{i_2 j_2}}$ and so on. The execution finishes at the point $\mathbf{b_{i_{|p|} j_{|p|}}}$. We assume, that transitions between points in space while moving between tasks or from the initial point to a task are performed along straight line segment (i.e. we use Euclidian distance). So, the execution time of plan p by the q-th vehicle is:

$$t_q(p) = \sum_{k=1}^{|p|} l_{i_k j_k} + \frac{1}{V}(\rho(\mathbf{s_q}, \mathbf{a_{i_1 j_1}}) + \sum_{k=2}^{|p|} \rho(\mathbf{b_{i_{k-1} j_{k-1}}}, \mathbf{a_{i_k j_k}})),$$

where V is the speed of vehicle.

We denote as common plan $P = (p_1, p_2, ..., p_m)$ such cortege of single vehicle plans that each task exists in it just once and in one way of execution. Total execution time can be denoted as $t(P) = \max_{q \in 1..m} t_q(p_q)$, since a mission can be considered accomplished only when all vehicles are done.

The optimization problem is concluded in finding a common plan P, minimizing $t(P)$ for given initial positions of all vehicles, given tasks and ways of their execution. The problem of generating a plan is close to multiple traveling salesman problem /MTSP/. There are well-known methods to solve it, both exact and approximate, described in [10]. Let's consider two algorithms, solving this problem.

The first algorithm gives the exact solution. It's based on the idea of dynamic programming and it's a modification of Held-Karp algorithm [11], solving TSP problem. In Held-Karp algorithm the state of system is described by the last visited vertex of the graph (TSP problem is stated on a full graph with different costs on edges) and the subset of vertices, which were already visited.

Using a binary search, our problem can be reduced to the following: find a common plan P with execution time $t(P) \le T$ or state, that it's impossible. We will apply a dynamic programming to solve the last problem. Consider sequential generating of a plan for each AUV, with only those transitions between states, which makes execution time for current vehicle not more than T. The state of our system is described by the triple $(q, S, \mathbf{u})$, where q is the number of current vehicle, S is the subset of tasks, and $\mathbf{u}$ is the number of vertex, in which the q-th vehicle will locate by the end of current plan (this could be any vertex of $\sum_{i \in 1..n} v_i + m$ vertices from vertex sets, given by b and s).

Further, denote as $t_T(q, S, \mathbf{u})$ the minimal execution time for q-th vehicle in the state $(q, S, \mathbf{u})$ for given T. A transition to a new state is either choice of the next task, which is not in S, or switch to generating a plan for the vehicle number $q+1$. So, transitions are described as follows:

$$t_T(q, S, \mathbf{s_q}) = \begin{cases} 0, & \text{if } \min_{\mathbf{u}} t_T(q-1, S, \mathbf{u}) \neq \infty, \\ \infty, & \text{if the state is not reachable,} \end{cases};$$

$$t_T(q, S, \mathbf{b_{ij}}) = l_{ij} + \min(t_T(q, S \setminus \{i\}, \mathbf{s_q}) + \frac{1}{V}|\mathbf{s_q} - \mathbf{a_{ij}}|,$$

$$\min_{i_1 \in S \setminus \{i\}} \min_{j_1 \in 1..v_{i1}} (t_T(q, S \setminus \{i\}, b_{i_1 j_1}) + \frac{1}{V}|\mathbf{b_{i_1 j_1}} - \mathbf{a_{ij}}|)),$$

where $i \in S$. Transitions are performed only if gained $t_T(q, S, \mathbf{u})$ is not more than T.

The structure of the algorithm for the fixed value of T is as follows. First, the value $t_T(1, \{\}, \mathbf{s_1}) = 0$ is initialized. Then all other t_T are calculated in the order of non-decreasing q and non-decreasing $|S|$ for the fixed q.

The second algorithm is a modification of auction method applied to a centralized planning. In this solution all tasks are sequenced. Each task in all ways of execution is added to the already generated partial plan of each single vehicle. The vehicle, which has the minimal cost of partial plan after inserting the task, is declared as a winner and the task is assigned to it. So, for each task we choose:

- a vehicle to assign the task;
- a position of the task in partial plan for the vehicle;
- a way of execution of task, which can be applied.

In the real conditions there is a possibility of emergency situations. An emergency could be more time execution needed for some task or an AUV failure. According to the principle of indivisibility of tasks, it should be taken into account, that some vehicles could be available not at once as emergency detected, but only after finishing their current tasks. This leads to some modifications in algorithms described above. Generally, the central control node should continuously monitor the execution of current plan and make a re-planning in case of a strong deviation from the plan.

In the paper [12] the application and modeling results of designed algorithms to a problem of search and estimation of local heterogeneities is discussed. The problem of adaptive data sampling by group of AUVs for further high-precision mapping of environment parameters under restriction of whole given area covering based on mentioned above algorithms is considered in [13].

3. Control System of MARC AUV for Group Operations

Small AUV named MARC /Marine Autonomous Robotic Complex/ (fig. 1) was designed in 2011 in Scientifical-Educational Center "Underwater robotic" that was organized based on Far Eastern Federal University /FEFU/ and Institute for Marine Technology Problems FEB RAS. The AUV has weight of about 50 kg and length of 1800 mm (depending of sensors configuration). Maximum diameter of the vehicle is 250 mm.

Control software of MARC AUV is a hybrid system. It consists of several executable modules running simultaneously. Some of the modules are intended to run on the ordinary AUV onboard computer, while others are designated to run on AUV central node or operator console. Modules are distributed by levels in the following way.

- Low level consists of onboard device drivers, navigation module, motion regulators, and control-alarm system. Frequent message transmission (tens of messages per second) is typical for modules of this layer.
- Middle layer is implemented in group mission manager. This module performs plan, which is build by central node. The plan is separated on elementary actions from task library of AUV.
- Upper layer is represented by operator software and planner. The latest implements functions of central control node. In can run on AUV central node or operator console.

For communication between modules the messages are used, which lets to generalize communication between modules to communication between AUVs in a natural way. For messages transmitting through network sockets the IPC library [14] is used. It lets to run control system modules in various configurations with several onboard computers.

Fig. 1. AUV MARC is intended for group operations.

The part of control system within a single AUV of a group is depicted on fig. 2. The common structure of the system is depicted on fig. 3 with suggestion, that central node is being run on operator console. The following scheme is used to exchange IPC-messages between one of AUVs and central node (a message in hydroacoustic channel from group mission manager to central node is given as an example):

- group mission manager publishes IPC-message for hydroacoustic modem driver in local AUV network;
- hydroacoustic modem driver receives the message and transmits a hydroacoustic package;
- operator console modem receives the package and modem driver publishes IPC-message in operator console network.

Modem driver also responsible for reliable connection (channel level). It's suggested that module, which sends the message, receives either delivery confirmation or deny.

A. Group part of control system

Group work is implemented by interaction of two programs: planner on central node and group mission manager on each AUV.

There are several types of messages, which are exchanged by the planner and group mission manager. These messages are listed in table 1. It is suggested that each AUV is provided with a unique identifier – its address. Each message contains address field.

Let us consider a planner algorithm in details. It takes mission description as its input and works until all tasks are finished.

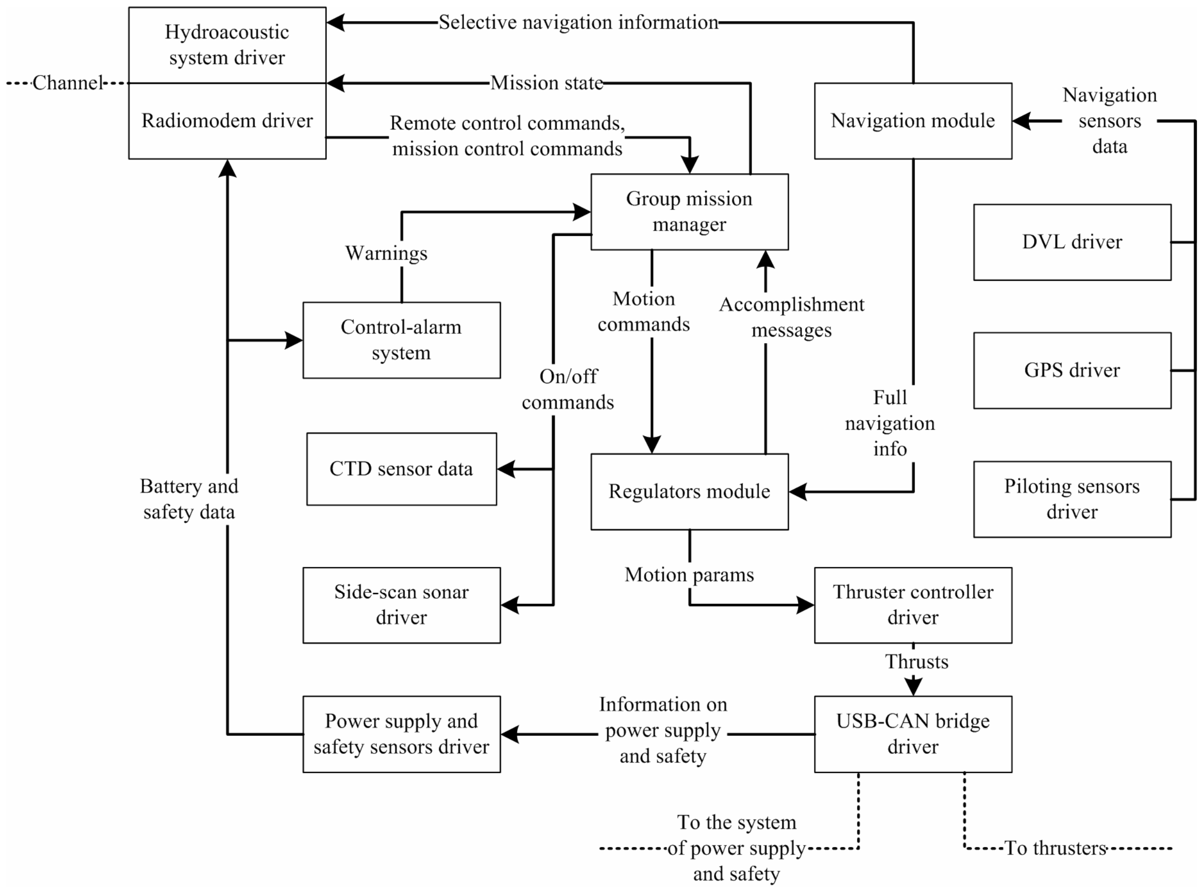

Fig. 2. Structure of control system within a single AUV.

The planner keeps information about all AUVs, which are taking (or took) part in the mission and about each task. For each task its status (done, running, pending for run) and AUV identifier are stored. Plan and current state of this plan for each AUV is stored. Moreover, a list of AUVs with plan failed to submit is stored. These plans are periodically resubmitted.

The base of planner is message processing cycle. This cycle consists of the following steps.

- Check for new messages.
- Update of tasks state if there are new MSG_TASK_DONE or MSG_TASK_START messages.
- Check for updates in group composition: new message MSG_VEHICLE_AWAY from a vehicle in the group, or new message MSG_VEHICLE_CAPS from a vehicle not in the group, or there was no MSG_VEHICLE_ALIVE message for certain time from one of vehicles in the group.
- If group composition is changed or one of AUVs has fulfilled its plan, replaning and sending of new plans to each vehicle with MSG_PLAN message are executed.
- If the list of not submitted plans is not empty then plans for which resubmit time is out are sent.

Now let us consider the group mission manager algorithm. It runs on each AUV in the group and stores information on current plan and its state. All the time while mission manager is running it whether performs group mission or tries to join it. The list of messages, which had unsuccessful submit, is

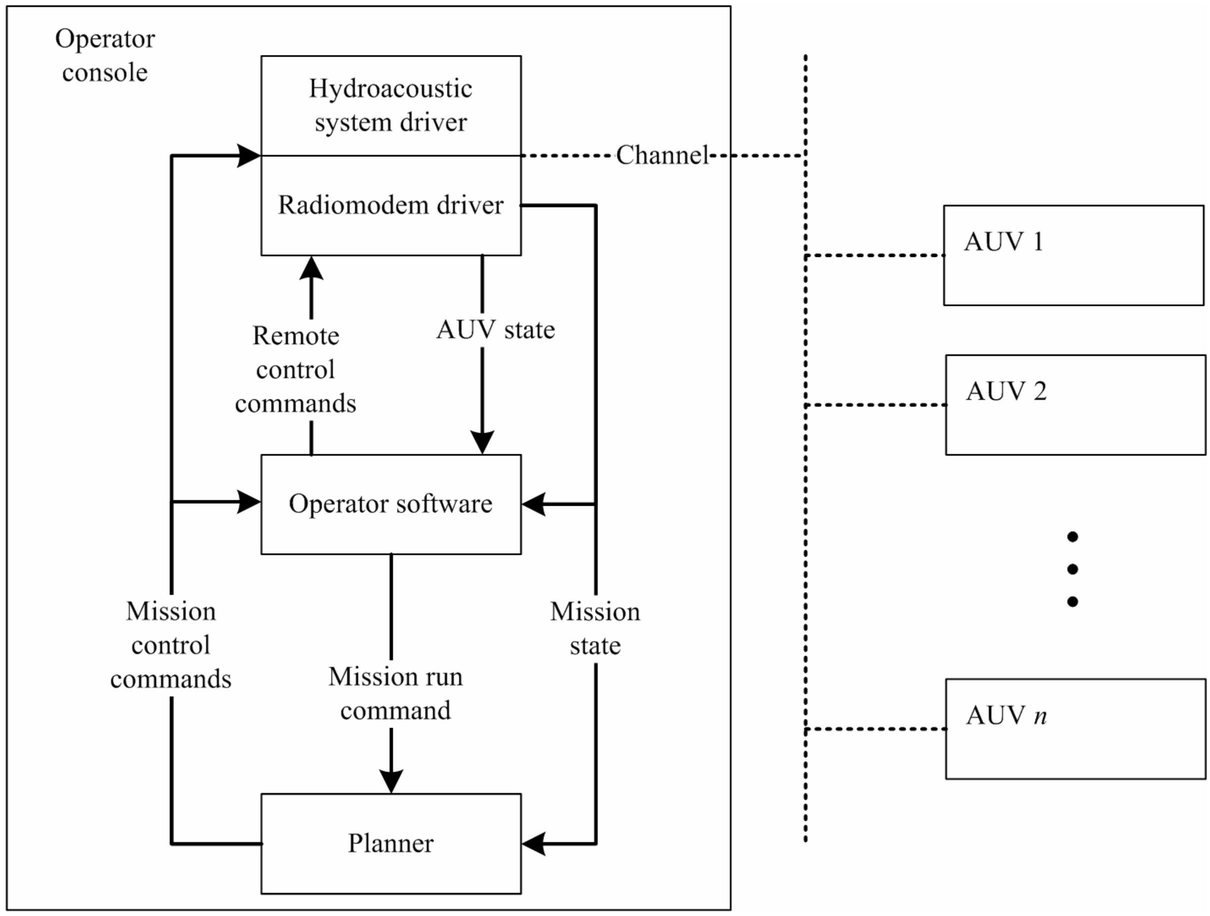

Fig. 3. Common structure of the control system – arrows denotes IPC messages of the same type.

supported like in the planner. These messages are periodically resubmitted. The list always contains whether MSG_VEHICLE_STATE message (if the vehicle is in the group), or MSG_VEHICLE_CAPS (if the vehicle is not in the group). The message processing cycle for group mission manager contains the following steps.

1. Check for new messages.
2. If there is a message from operator console on leaving the group, then send MSG_VEHICLE_AWAY end exit the cycle.
3. If the vehicle is not in a group then go to step 8.
4. If there is a new MSG_PLAN message then renew the plan. It is considered that AUV in the group.
5. If there is a new message from regulators module on accomplishment of current task, then send MSG_TASK_DONE message. If the task was last in the current plan then it is considered that AUV as not in the group.
6. If there is a new message form regulators module or plan is renewed then it is started a transition to new task start point.
7. If there is a new message from regulators module on arrival to the

Table 1. Messages between planner and AUV mission manager.

Identifier	Direction	Description
MSG_VEHICLE_CAPS	to the planner	A report about AUV readiness to join the mission. A message contains information about the AUV motion parameters and onboard devices.
MSG_VEHICLE_AWAY	to the planner	A report that the AUV quits mission by remote control command or by message from control-alarm system.
MSG_VEHICLE_STATE	to the planner	A periodical message, saying that the AUV is performing its plan. It contains the information about vehicle state.
MSG_TASK_DONE	to the planner	A report that the AUV has performed its current task. It contains an identifier of accomplished task.
MSG_TASK_START	to the planner	A report that AUV has started to run its next task. It contains the task identifier.
MSG_ PLAN	from the planner	A message contains task enumeration in the order of execution. Each task is characterized by its type, parameters and identifier.

start point of current task then MSG_TASK_START message is sent.

8. Messages are sent if resend time is out. If MSG_VEHICLE_CAPS or MSG_VEHICLE_STATE were sent, then return one of them to the queue depending on whether the AUV is in a group.

4. Some Marine Trial Results

The designed control system was tested onboard of single MARC AUV. Trials were conducted in the Great Peter Bay near Vladivostok. Scientific research ship "Yuri Molokov" (fig. 4) took part in the operations.

The tested control system now includes about 20 modules except testing and diagnostics subroutines. On this stage of trials a searching task with meander like trajectory was used as group mission manager imitator. The results of task execution is shown on fig. 5.

The marine trials shown that control system operates properly on board of MARC AUV in real conditions.

Fig. 4. MARC AUV fulfils the mission program.

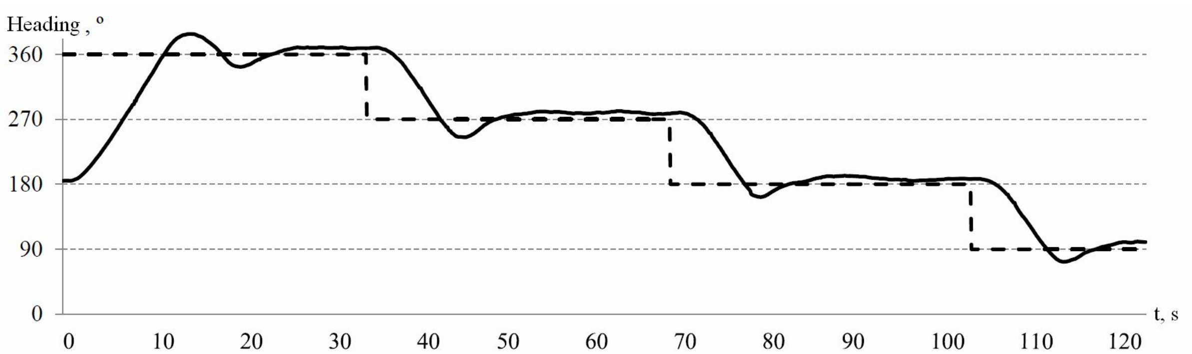

Fig.5. The programming (a) and real (b) MARC AUV heading during of one testing mission execution.

Acknowledgments

This work was fulfilled in frame of contract №14.A18.21.0283 with Ministry of Education and was supported by the Russian Fund of Fundamental Investigation under grant N100800249a.

References

[1] DIAS H., CALADO P., BENCATEL R., GOMES R., FERREIRA S., SOUSA J. Operations with multiple unmanned systems // 2012 IEEE/RSJ

International Conference on Intelligent Robots and Systems (IROS), October 7-12, 2012, Vilamoura, Portugal.

[2] PINTO J., CALADO P., BRAGA J., DIAS P., MARTINS R., MARQUES E., SOUSA J. Implementation of a control architecture for networked vehicle systems // IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles (NGCUV 2012), April 10–12, 2012, Porto, Portugal.

[3] MARTINS R., DIAS P. S., MARQUES E. R. B., PINTO J., SOUSA J. B., PEREIRA F. L. IMC: A communication protocol for networked vehicles and sensors // Proceedings of the OCEANS 2009 MTS/IEEE Conference, May 11–14, 2009, Bremen, Germany.

[4] WANG H., WANG L. L., LI J., BIAN X., ZHANG W. Autonomous Team Mission Planning for AUVs // Proceedings of the OCEANS 2011 MTS/IEEE Conference, September 19-22, 2011, Kona, USA.

[5] BENJAMIN M. R. Interval programming: a multi-objective optimization model for autonomous vehicle control: Doctoral Dissertation. – Brown University Providence, RI, USA. – 2002. – 130 p.

[6] CHITRE M. DSAAV – A distributed software architecture for autonomous vehicles // Proceedings of the OCEANS 2008 MTS/IEEE Conference, September 15–18, 2008, Quebec City, Canada.

[7] M.D.Ageev, D. R Blidberg, V.E.Gornak, D.B.Khmelkov, A.Ph. Scherbatyuk, Ju.V.Vaulin. Mission Control System for Solar AUV and Results of the Vehicle Long Time Operation Trials. Proc. of the 11th Int. Symp. on Unmanned Untethered Submersible Technology, New Hampshire, 2001.

[8] http://moos-ivp.org

[9] RAJAN K., PY F., BARREIRO J. Towards Deliberative Control in Marine Robotics // Marine Robot Autonomy. – Springer, 2013. – P. 91–176.

[10] G. Gutin, A. Punnen (eds.), The Travelling Salesman Problem and Its Variations, Kluwer Academic Publishers, 2002.

[11] M. Held, R. Karp, A Dynamic Programming Approach to Sequencing Problems, Journal of the Society for Industrial and Applied Mathematics, vol. 10, no. 1. pp. 196-210, 1962.

[12] Scherbatyuk A.Ph., Tuphanov I.E. Algorithms for Underwater Local Heterogeneity Survey Based on AUV Group Usage. Proceedings of the OCEANS 2012 MTS/IEEE Conference, May 21-24, 2012, Yeosu, Korea, ISBN CD-ROM: 978-1-4577-2090-1.

[13] Tuphanov I.E., Scherbatyuk A.Ph. Adaptive Algorithm of AUV Meander Pattern Trajectory Planning for Underwater Sampling. Proceedings of the Tenth (2012) ISOPE Pacific/Asia Offshore Mechanics Symposium. Vladivostok, Russia, October 3-5, 2012, ISBN 978–1 880653 93-7: ISSN 1946-004X.

[14] http://www.cs.cmu.edu/~IPC