A Reconfigurable Web-based Simulation Environment for AUV

*Alexander Pavin
Laboratory of Underwater Robots
Far Eastern Federal University
Vladivostok, Russia
pavin@bk.ru

Alexander Inzartsev
Laboratory of Underwater Robots
Far Eastern Federal University
Vladivostok, Russia
inzartsev@yahoo.com

Grigoriy Eliseenko
Control Systems Laboratory
Institute of Marine Technology Problems
Vladivostok, Russia
eliseenko_g@mail.ru

Oleg Lebedko, Mikhail Panin
Information Systems and Technology Specialty
Far Eastern Federal University
Vladivostok, Russia
loa92@mail.ru, kars25rus@mail.ru

***Abstract*—This paper is related to solving the autonomous underwater vehicle simulation problem. The main goal of the simulation is to debug AUV control algorithms. Simulator architecture, user interface, usage experience and applied techniques are described in this paper.**

***Keywords*—robot simulator; user interface; control system; web-interface; AUV; ROV; UUV.**

I. Introduction

The development of reliable autonomous underwater vehicle (AUV) software is impossible without a thorough simulation of the control system and navigation. The need for an easily scalable (for different vehicles) simulator caused growing numbers of underwater vehicles and their devices. This work is part of integrated project "Modern technologies and technical means of monitoring the state of marine ecosystems and marine biological resources" [1-4]. The main goals of the newly developed simulator are:

- Simple **reconfiguration** for existing and new AUVs.
- Operability in the most popular **operating systems** (Linux, QNX, Mac-OS and Windows).
- A remote user interface supporting **mobile devices**.
- Tight integration with any AUV **control system** and existing devices.
- The ability to **simulate emergencies**.
- A performance **faster than a real-time** mode for long-term mission simulation.
- **Simple and stable architecture** that allows fast integration of new modules into the simulation environment (e.g. a new module of a multi-beam echo-sounder).

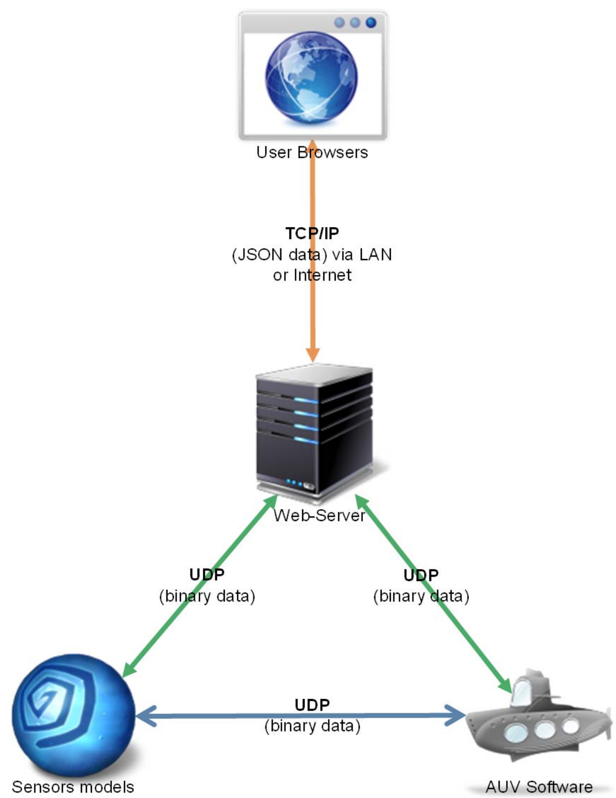

Fig. 1. Simulation environment architecture.

Many years of experience with AUV development in two different organizations (Institute of Marine Technology Problems, Far Eastern Branch of the Russian Academy of Sciences and Far Eastern Federal University) allows us to offer the following solutions to achieve the mentioned goals (see fig. 1):

*This work was supported by Russian Science Foundation, (project No. 14-50-00034)

- Decentralized distributed architecture without any central module for inter-process communication.
- A web-based interface for interaction simulator software with users.
- JSON data exchange between web-client and simulation web-server on the base of TCP/IP protocol.
- Binary (row) data communication using UDP-multicast datagrams between simulator and AUV software.

Decentralized distributed architecture is very suitable for the reconfigurable simulation environment. We used the asynchronous interaction of separate executable modules often located in different network segments (see fig. 2). This solution allows using more computational resources when solving problems of photo image rendering and acoustic echogram modeling (e.g. side scan sonar or multi-beam sonar modeling).

Fig. 2 represents the scheme of simulator using depending on the problems to be solved. First option is using for simulation without AUV. In this case all required simulation modules, some AUV modules (non-required of specific devices), web-server of simulating environment and browser with graphical user interface launched at user's computer. Second option is useful for prestarting mission simulation of the vehicle. All simulating modules and web-server is located on AUV in this case. User access to a robot is made by a local area network or Internet via any available device with browser (HTML 5 support is needed). Last variant is necessary for resource-intensive model experiments that require separate computers for solving the rendering tasks of photo images, acoustic echograms and etc.

II. GRAPHIC USER INTERFACE

A Web-based interface was used for operator interaction with the simulation environment and the AUV control system. We applied the bootstrap technology to achieve compatibility with mobile devices. Such an approach allows remote simulation using an existing AUV control system from any computer or mobile device without installing additional software.

The main problem of graphical user interface (GUI) development is placing the AUV indicators, simulation data and user control elements in a browser window (fig. 3). It should be noted that the structure of information, the number of elements and indicators may change when adding/editing software modules. The following approach is proposed to solve this problem:

- The user interaction with modeling system is performed only through the web-server, and web-socket technology uses for sending and receiving data. The data format is described in Chapter 3 - Reconfiguration.
- All system information divided into input and output data in relation to the AUV & simulator. There are two types of input data: commands which are sent once by request (for example, the command of "on" or "off" to some device) and control values which are sent periodically (e.g., velocity correction of the robot).

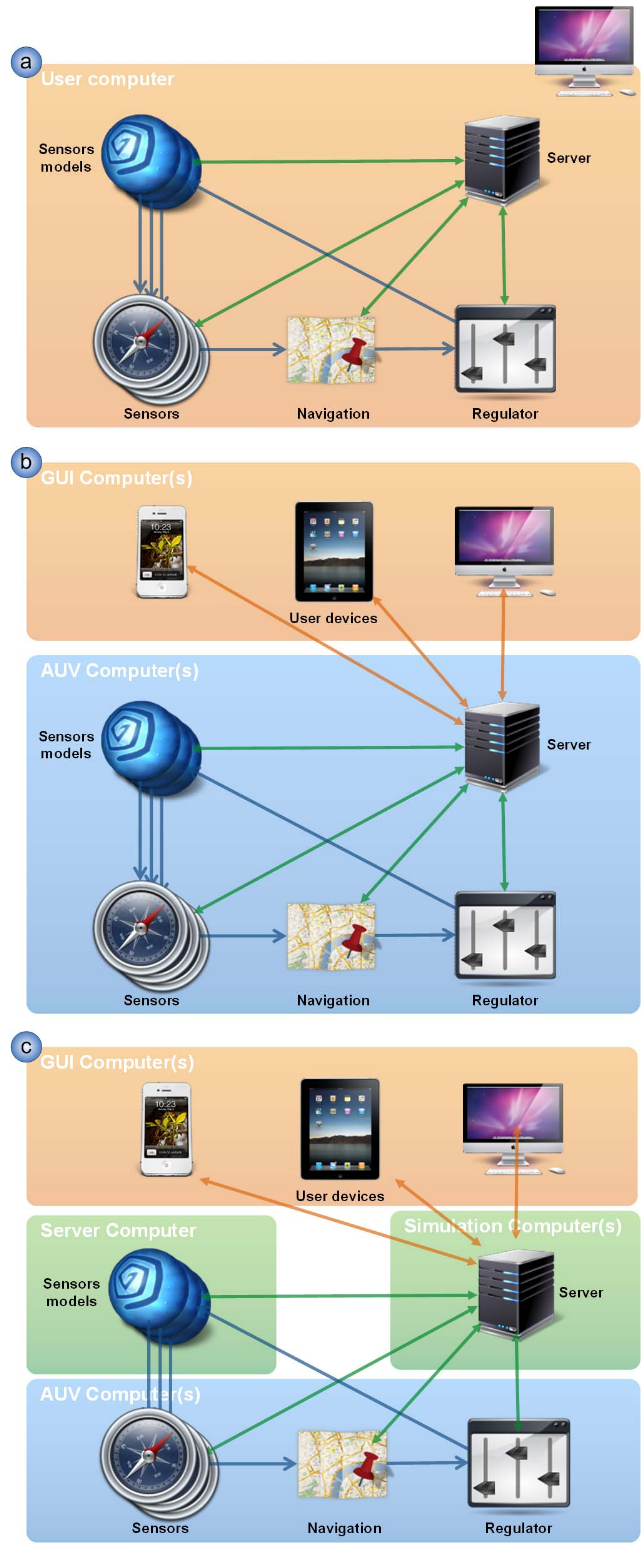

Fig.2. Software modules location depends on different tasks: a) simulation tasks without AUV; b) mission simulation on AUV; c) high computational tasks with AUV and many simulation computers.

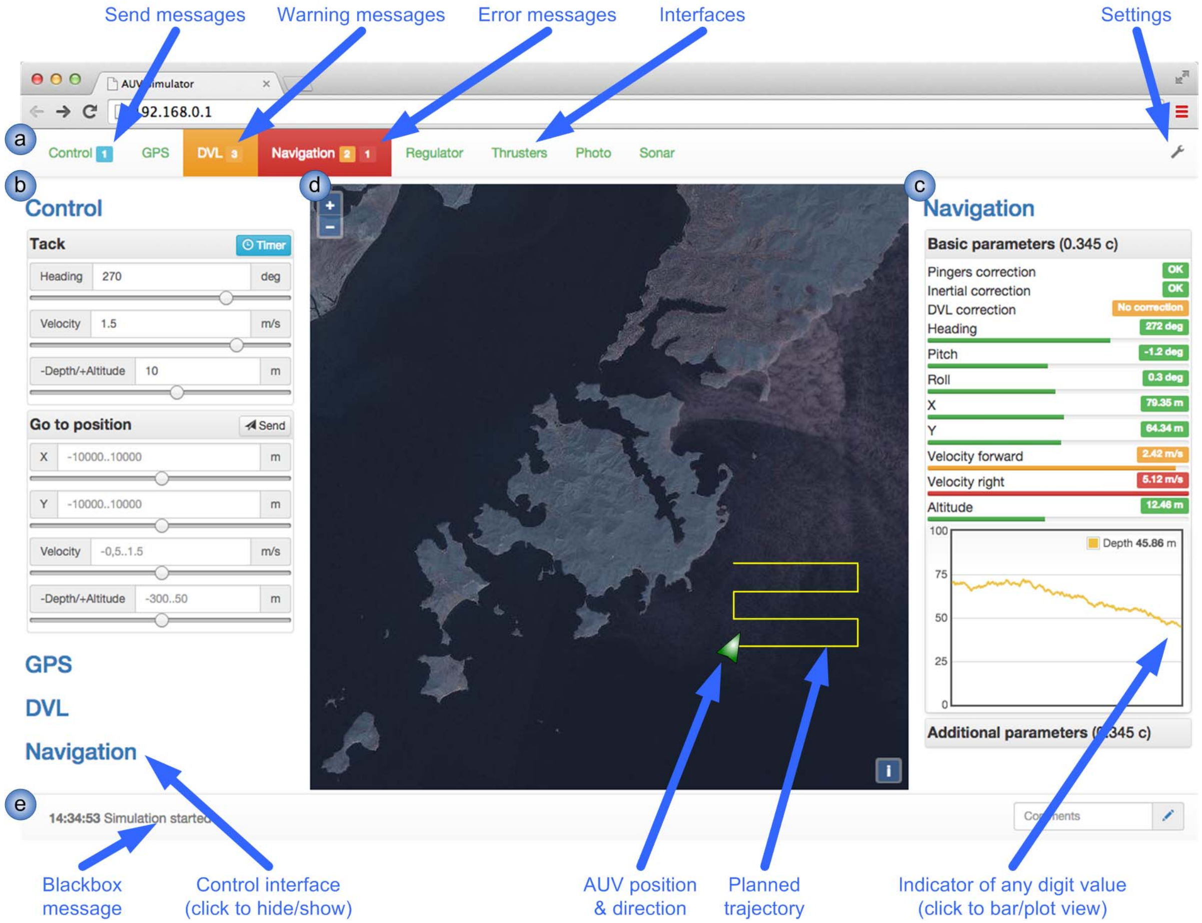

Fig. 3. Simulator web-GUI.

Outgoing data are called messages and published by AUV control system & simulation modules at any given time.

- Input and output data are combined into "interfaces" according to their logical identity. For example, the interface "Navigation" includes one input command to the navigation system ("on" or "off" correction of coordinates using acoustic pingers) and many output messages: AUV heading, velocity, coordinates, depth and others.

Thus, user interaction with the simulator and AUV software reduced to receiving and sending messages that appear in the browser. The browser window is divided into the following parts:

- **Top menu** consists of interfaces list (fig. 3-a). Each of this interface is highlighted according to its state. Click on the corresponding interface changes the contents of the control elements panel (fig. 3-b) and the indicators panel (fig. 3-c).
- **Control panel** contains control widgets of each subsystems of the robot (fig. 3-b). These elements grouped into commands (which can be pub by pressing the "Send" button) or control values (which can be pub by timer - button "Timer"). The number of commands, control values and widgets depends on the particular configuration of AUV.
- **Indicators panel** (fig. 3-c) includes the messages generated by the simulator and AUV subsystems. Each message is a set of fields of a certain type: boolean, integer, number, enum or string. This fields are displayed by default in the short form: name, value, unit of measure and bars for number values. By clicking on the appropriate field the graph of this value is appear (for boolean, integer, number and enum).
- **Map section** (fig. 3-d) displays navigation information with geo-referenced: AUV coordinates, its trajectory, allowed and prohibited area, etc. This part of the interface implementation was made using tiles technic and based on the component OpenLayers. The future direction of work is seen in adding image stitching functionality of photos and acoustic echograms with geo-referenced.
- **Black-box** footer displays text messages and debugging information that appears in the modeling’s process (fig. 3-e). Here you can add a custom message as a comment (fig. 3-e - right side).

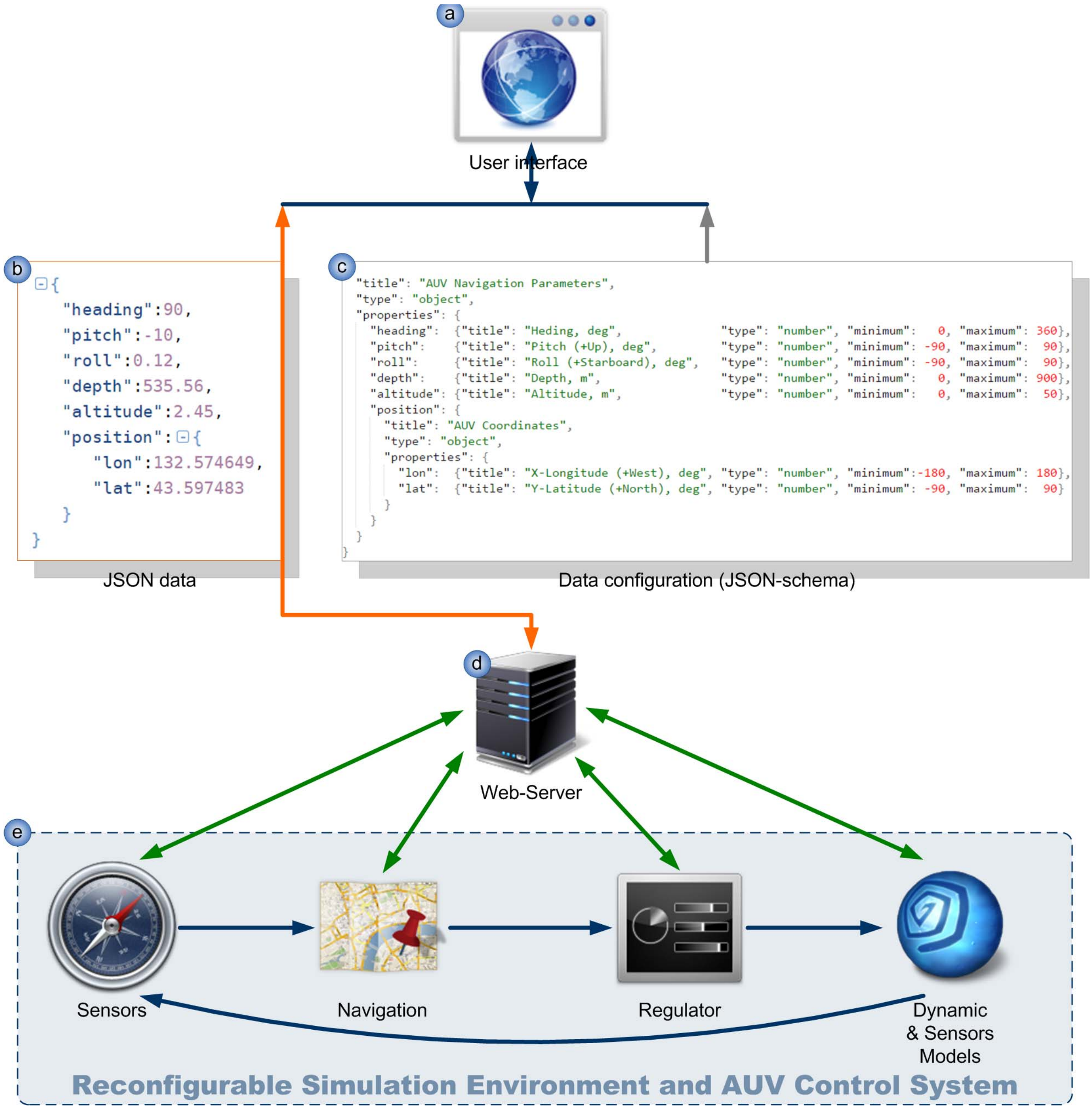

Fig. 4. Simulation environment architecture: a) simulator user interface; b) data transmitted between GUI and a web-server; c) part of the simulator configuration file; d) a web-server communicator between GUI and the simulation environment; e) the simulation environment with an AUV control system.

III. Reconfiguration

A simple reconfiguration of the system is carried out using the JSON (JavaScript Object Notation) and JSON-Schema formats for data exchange between AUV modules and a graphic user interface. All you need to add a new module – just describe your new data in the JSON schema form (see fig. 4-c). After the first browser-request to the server, all described data will automatically be added to the graphical user interface, which reconfigures in accordance with the data description. GUI allows display of all basic data types of the JSON schema standard in the form of numbers and charts: integer, number, boolean, string, enum, array and object.

The key element to build the reconfigurable system is a server. Server's main task is to receive raw data from simulator modules (and AUV control system modules), convert it to JSON format and transmit to client web-page. Reverse mechanizm is used to control the simulation process: client-to-server data converted from JSON format to the AUV/simulator format and server publish UDP-datagrams in AUV network. In this way, when any simulator or AUV is changed it is enough to describe the new data in JSON format.

IV. Inter-Process Communication

An asynchronous UDP-datagram mechanism was used for data exchange between the server and the running simulation programs. It allows not maintaining the connection between system components and added modules as needed even during a simulation.

TABLE I. Parameters of the numerical experiments for inter-process communication.

No	Parameters	QNX	Windows	Units
1	Message size	1	1	Kb
2	Message period	100	100	ms
3	Processor frequency	3000	3000	MHz
4	Computer memory size	2048	3250	Mb
5	LAN	100	100	Mb/s

A series of numerical experiments with multiple messaging systems were produced when choosing an optimal mechanism for inter-process communication. Fig. 5 shows the results of time delays measurements and the CPU load for two systems: IPC [5] and UDP-multicast datagrams. IPC system has a central server and provides data transmitting via TCP/IP protocol. UDP-multicast protocol provides decentralized asynchronous messaging. We used two computers with two operating systems QNX 6.5.0 and Windows XP Service Pack 2 for these experiments. Technical characteristics of each of this computers is listed in Table 1. The experiment is consist in increase of number of modules (involved in the messages exchange) and measurement of the performance characteristics. The measurements were performed for two modes: for local host and for local area network (LAN). Every newly added module did one message publication in working period (see in tab. 1), but one message took every other system modules, i.e. connections was "every with every" (fully connected graph).

Experimental results (Fig. 5) show that the IPC library with the central server can not handle with the messages transfer with 15 fully connected modules (the time delay between messages exceeds the period of publication). While decentralized UDP-multicast messaging exceeds this threshold of publication on 40 modules.

V. Simulation Environment

Simulation environment model represent a digital spatial model of marine hydrology and dynamic model of the AUV disturbances. This model determined by the speed and direction of the underwater currents, and irregularities of the marine environment. Simulation software of an environment generates and outputs the data as a set of characteristics for a particular point in space.

The simulator includes one basic module (dynamic module) and many other modules for simulation different AUV systems, like DVL, photo, sonars etc. Each module is a separate executable program, which can be run on one or many computers (in one local area network). The dynamic module calculates AUV movement in various modes: at cruising speed, at very low speed and along complex spatial trajectories. The

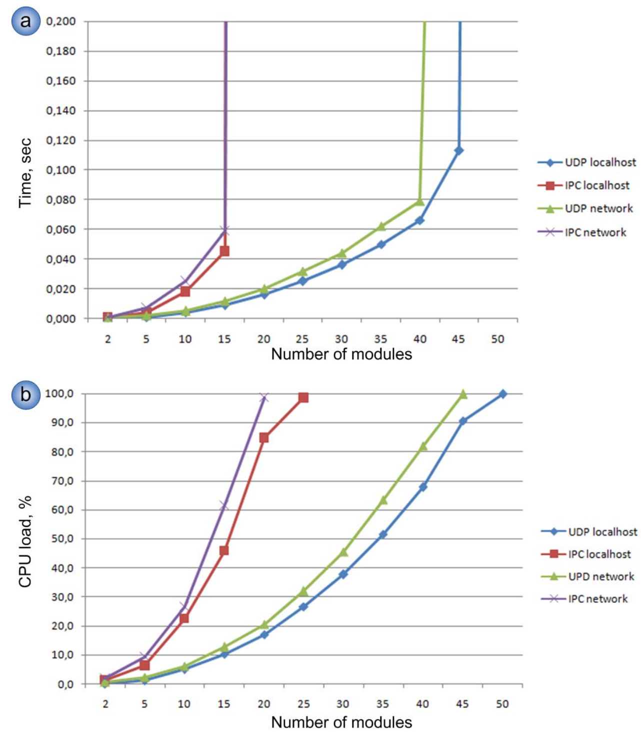

Fig. 5. Comparison of two inter-process communication models (IPC [5] and UDP-multicast): a) message delays vs number of modules; b) CPU loading vs number of modules.

AUV's trajectory is calculated in the connected coordinate system (coordinate axes are directed along the vehicle axes of symmetry). For calculation the position and the velocity in the connected system of coordinates the following equation is used:

$$\boldsymbol{W}\, dV/dt = T(P) + R(V) + G(\varphi,\psi,\theta) + A(\varphi,\psi,\theta) \qquad (1)$$

where: t - time; V - velocity vector $[V_x, V_y, V_z, V_\varphi, V_\psi, V_\theta]^T$; $\boldsymbol{W}$ - weight matrix of the vehicle with the attached weights of a liquid (matrix 6*6); $T(P)$ - thruster force vector $[T_x, T_y, T_z, T_\varphi, T_\psi, T_\theta]^T$ dependent on control parameters; $R(V)$ - resistance force matrix $[R_x, R_y, R_z, R_\varphi, R_\psi, R_\theta]^T$ dependent on vehicle velocity and current velocity; $G(\varphi, \psi, \theta)$ - gravity force vector dependent on angular position of the vehicle; $A(\varphi, \psi, \theta)$ - Archimedean force vector (dependent on angular position of the vehicle); P - control parameters vector; φ - heading; ψ - pitch; θ - careen. More info about used math model see [6].

Photo image simulation module generates two-dimensional images, using three-dimensional computer graphic based on OpenGL standard. This module as well as side scan sonar module provides seabed and underwater objects rendering. Fig. 6 shows screenshots of the underwater scenes during simulation experiments.

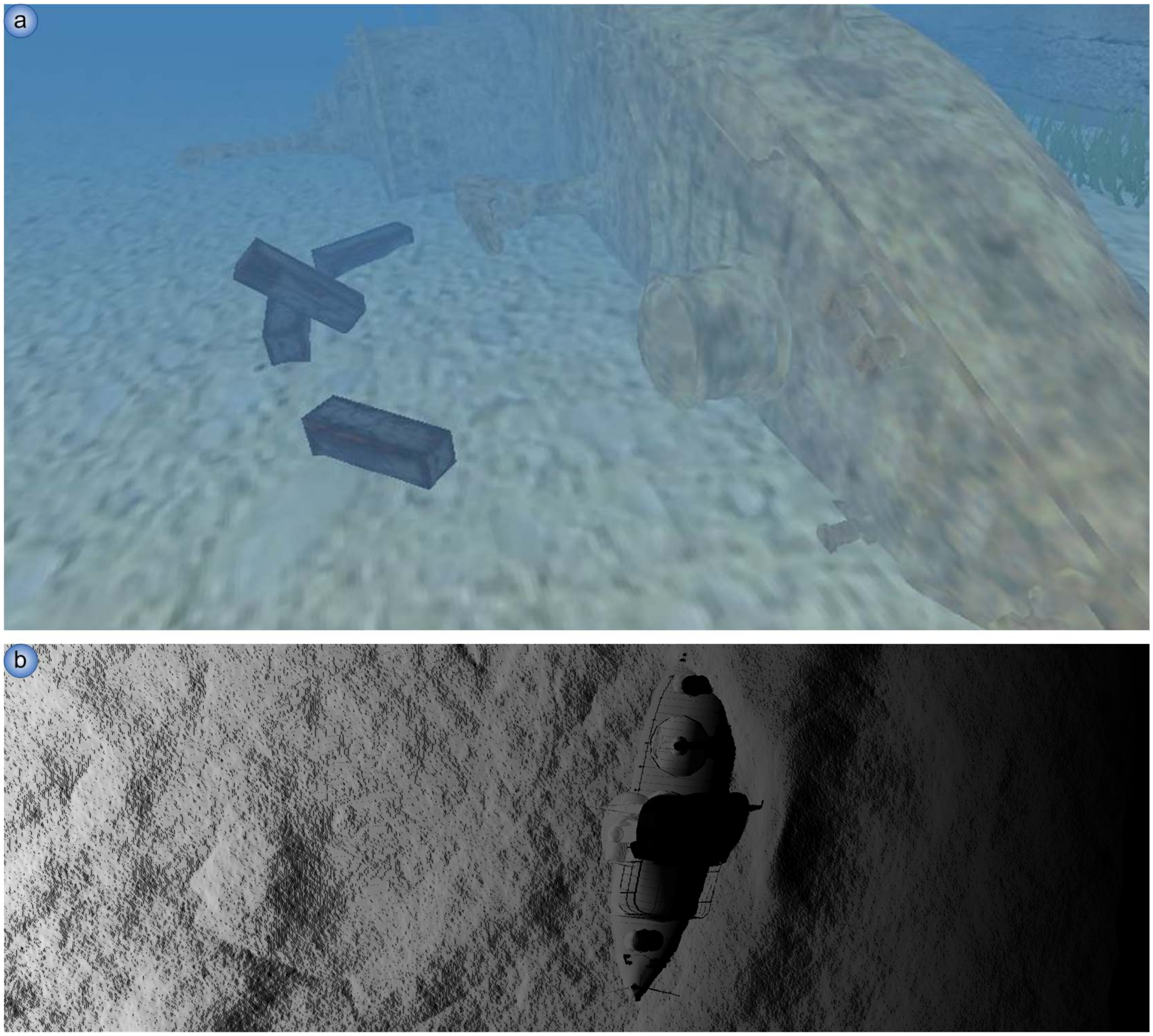

Fig. 6. Simulated images of the seabottom and objects on it: a) photo image; b) side scan sonar image.

VI. CONCLUSION

The developed simulation software includes a model of navigation, control system and information-measuring system of the AUV and the environment (to perform computational experiments). This simulator was used to debug algorithms of several AUVs designed in FEFU & IMTP FEB RAS. The experience showed the high effectiveness of the selected approaches and techniques, especially when solving complex tasks that require large computing resources (simulation faster than a real-time mode and photo/acoustic image rendering). A comparison of modeling experiments with field tests allows speaking of the high quality of the simulation data.

The future direction of research is seen in the finalization of the simulator GUI, as well as its integration with AUV control interface. It is expected that in the end of the study we will get a single user interface, that allow to control of any underwater vehicle. This interface intended for AUV/ROV movement modeling, simulation prelaunch, diagnostics and debugging of control algorithms. We suppose to use this system for single vehicle as well as for group of robots.

ACKNOWLEDGMENT

The authors acknowledge Russian Science Foundation for the financial support of this study (Grant No. 14-50-00034). Also authors thanks colleagues from IMTP FEB RAS and FEFU, who were involved in this and other projects concerning AUV design.

REFERENCES

[1] Yu. N. Kulchin, A. Yu. Mayor, D. Yu. Proschenko, A. Yu. Zhizhchenko, S. S. Golik, M. Yu. Babiy, A. G. Mirochnik, Modification of a new polymer photorecording material based on PMMA doped with 2,2-difluoro-4-(9-antracyl)-6-methyl-1,3,2-dioxaborine by ultrashort

pulses Quantum Electronics, 2015, 45 (5), 477–481 DOI: 10.1070/QE2015v045n05ABEH015764.

[2] Kameneva P.A., Imbs A.B., Orlova T.Yu.. 2015. Distribution of DTX-3 in edible and non-edible parts of Crenomytilus grayanus from the Sea of Japan, Toxicon 98, pp. 1-3.

[3] Ryazanova T.V., Eliseikina M.G., Kalabekov I.M., Odintsova N.A. A herpes-like virus in king crabs: Characterization and transmission under laboratory conditions// Journal of Invertebrate Pathology 127 (2015) 21–31.

[4] Beleneva I.A., Shamshurina E.V., Eliseikina M.G. Assessment of the toxic effect exerted by fluorescent pseudomonads on embryos and larvae of the sea urchin strongylocentrotus nudus //Ecotoxicol Environ Saf. 2015;115:263-71.

[5] Inter Process Communication (IPC) [Internet resource] / IPC. URL: http://www.cs.cmu.edu/~ipc/

[6] Alexander Pavin "Simulation Environment for AUV Control Algorithms Development" // Proceedings of 14th International Symposium on Unmanned Untethered Submersible Technology (UUST05), August 21-24, 2005, Durham, New Hampshire, USA.

[7] Andrey Bagnitsky, Alexander Inzartsev, Alexander Pavin, Sergey Melman, Mikhail Morozov. Side Scan Sonar using for Underwater Cables & Pipelines Tracking by Means of AUV // Proceedings of Symposium on Underwater Technology 2011, 5-8 April, 2011, Komaba Research Campus, The University of Tokyo, Tokyo, Japan, ISBN: 978-1-4577-0163-4.