Available online at www.sciencedirect.com

ScienceDirect

Procedia Engineering 100 (2015) 269 – 277

Procedia Engineering

www.elsevier.com/locate/procedia

25th DAAAM International Symposium on Intelligent Manufacturing and Automation, 2014

Autonomous Mobile University Robots AMUR: Technology and Applications to Extreme Robotics

Vladimir F. Filaretov[a], Valentin E. Pryanichnikov[b,c]*

[a]*Far-East Federal university, Vladivostok, Russia*
[b]*Moscow State University "Stankin"and KIAM Russian Academy of Sciences, Miusskay sq.4, Moscow, 125047 Russia*
[c]*University of Zadar, MihovilaPavlinovica bb, Zadar, 23000 Croatia*

Abstract

This paper addresses the problems of creating an innovative technology for the education of engineers on the basis of the so-called "recurrent" method by forming scientific and R&D technical tasks for students as well as their engagement in the implementation of software and hardware for mobile robots, including special robotics. One of the big problems in education is to avoid the gap between theory and practice, interest and formal studying. Here, we mainly consider the scientific and technical support of this education process. It should be emphasized, that the key scientific and technical problems and its' decision, presented in the paper, were to find optimal decomposition and standardization of software/hardware and construction of the required software ("middleware"), implementation of mobile robots. The suggested solution in didactics was to transform the theoretical courses for students into creative educational tasks in the field of mobile robotics. This was performed by creating proper methodological materials tested in 5 universities and required to solve of engineering tasks: construction of efficient mobile robots (MRs) and their group control in the virtual laboratory; elaboration of scientific and educational tasks for students, taken from our practice in extreme robotics. We consider the scientific/technical support of this education process.

Peer-review under responsibility of DAAAM International Vienna

Keywords: Mobile autonomous technological robots AMUR; new educational technologies; group robots' control with delays in internet channels

* Corresponding author. Tel.: +7-903-769-7288.
E-mail address: v.e.pr@yandex.ru, vepr@keldysh.ru

Peer-review under responsibility of DAAAM International Vienna
doi:10.1016/j.proeng.2015.01.367

1. Main problems

Initially we were looking for a proper way of distant reprogramming/tele-reprogramming of control and computer vision systems (CCVS) installed by us into Brokk-110D and Brokk-330 mobile robots. These robots were redesigned for a group operations in dangerous zones with restricted human access. Since our customers wanted to have the opportunity to change CCVS during missions without stopping the robots, new mechatronic devices/sensors (made by third-party producers) were installed into the control system. We had to be able to conduct installations, improvements, and training of these specialists even from remote locations [1-7].

Our findings were used in tele-rehabilitation control [8] and in universities for educational purposes. Of course, we've analyzed works in tele-rehabilitation [9] and the most popular software for education in robotics such as [12, 13] and a dozen of other works. Our educational programs were designed for both mechatronic and IT specialists ("knowledge engineers"). This additionally motivated us to develop our own technology (in this paper, we discuss the decomposition of processes in CCVS) in the form of a "virtual laboratory": hundreds of mechatronic devices with hierarchically distributed program links are combined and any device and can be accessed from different points. This software is of the "middleware" type [14] and provides a dynamic reconfiguration of system components without stopping, automatic fail-back, including auto-configuration. The software supports x86\Windows\Linux\Android OS platforms with minimized size for onboard industrial solutions and easy-to-implement/easy-to-understand additions, drivers, and devices. This architecture provides high adaptability, capacity, and frequency of transmission of control commands and data (see also another DAAAM-2014 presentation by K.Kirsanov *et al*.).

Our experience gives good grounds to expect that the problems faced by students can be avoided. Their main problems in some fields are associated with the use of easily accessible data from the Internet without any comprehension and attempt to compose knowledge of the subject under consideration. This "tape recorder" method of learning for students and schoolchildren becomes dominant and completely inhibits their smart growth as individuals capable of independently exploring the world and creating something new. This leads to a kind of humanitarian disaster that considerably hinders the modernization of education. To avoid it, different models of student–professor interaction are used [15, 16]. The problem of creating an innovative technology is to provide conditions for students as "knowledge engineers" capable to formalize tasks and formulate easy-to-use algorithms for their work. We believe that the algorithms of mobile robots (MR) behavior are the most suitable for this type of IT and robotics education. Thus, we came to the task of designing a proper group of "university" robots that interact with each other and with students.

In parallel, we were engaged in developing software and hardware for mobile robots operating in extreme conditions. The problem was to find instruments for fast prototyping and distant access to executing control software without stopping MRs because the list of sensors and drives varies continuously in the field work of our customers [1, 3].

These two main lines bring us to a combined solution of the problem: implementation of some integrating software [2–4, 7] and making software/equipment for sensor-flow processing [6]. We discuss these problems in the present paper. This approach was used in our suggestions to industrial applications [5, 6].

2. Experience in education

On the basis of our experience and educational experiments conducted in the Russian State University for the Humanities (IT department) since 1992, we proposed a way to solve our current research problems (jointly with students) that are of significant interest for students as well. We had to identify an appropriate "subject area" and a mechanism for involving students into research projects and avoiding their fears of the unknown problems (blank sheet) by offering prototype solutions. These prototypes can be altered during the execution of students' tasks.

Students feel themselves most confident as experts in coding path-finding algorithms for mobile robots, moving in rooms or on the ground. This is due to the fact that the "robot–man" projection is quite natural and, therefore, the introspection of their own behavior can be easily transferred into mobile robots, learning the geometry of the environment through simple sensors. For mobile robots, it is relevant and meaningful to consider the problem of local navigation and terrain mapping. We consider SLAM-type problems and construction of algorithms, building the return path for the robots, that loss the Wi-Fi connection during supervised control.

Therefore, this subject area and this type of tasks are best suited for the development of creative teaching technologies for students. Thus, the problems of mechatronics and sensorics become efficient tools for training a wide range of IT-specialists and computer scientists not only in IT-technology but also in the humanities such as the construction of models of social interaction, communications, group behavior, and brain–computer interfaces, as well as mechanisms of interaction between robots and patients, and between robots and children. This also brings new people working for us in serious R&D or scientific projects and in which we are involved in the "Sensorika"laboratory.

It seems trivial to give students creative tasks; however, if the results of these nonstop renewing tasks are unknown, one has to find proper arguments to use these didactic instruments in universities. One of these instruments is the percentage of students who have a will to transform their laboratory work on mobile robotics (during 1–2 semesters) into the subject of their following diploma and then Ph.D. dissertation. This constitutes 5–15% of a group of 15–25 students (with different basic specializations such as informatics, electronics, and mechatronics). This exceeds the respective level in other universities. This instrument was tested in several technical universities in Russia and is currently being used in the University of Zadar (Croatia).

It should be noted that our first experiments started long ago. We constructed two PC-controlled mobile robots on two platforms: 3 wheels (3x2 DOF) and 4 wheels (4+2 DOF), both with ultrasonic locators (panoramic in 72 directions) and capable of carting a useful additional load of 15–25 kg. The education methodology on the basis of mobile robots have successfully been implemented as early as in 1985–1987 [10 (pp. 56–172, photo on p. 62), 11] (Fig. 1).

Fig. 1. One of our first (1985) educational mobile robots with scanning ultrasonic locator and controlled by PC with microprocessor (left photo). Configuration of modern educational mobile robots: Robotino (designed by "Festo") and AMUR (right, designed by "Sensorika" Laboratory, KIAM RAS) within the virtual laboratory "Intelligent Robotronics".

3. Solution method for education problems

The education problems mentioned above were solved by creating methodical systems implementing the "recurrent" approach in laboratory and seminar and lecture tasks to students. Here, the tasks for laboratory studies (12–19 topics for 25–50 students) were suggested for one year as a small research project, and the results obtained for each subject become the basis for new research projects to be carried out by the next group of students. This also requires the development of appropriate technical means including brain–computer interfaces (BCi) with our designed exoskeleton. We conducted our own experiments [8] related mainly to medical training, but we also used it as the subject/source for new tasks for students, involving them into the R&D or research processes.

The universities mentioned above used the specified method to organize the education for graduate and undergraduate students. Within these efforts, AMUR-5, AMUR-6, AMUR-7, AMUR-105, and AMUR-107 mobile robots were designed and manufactured to be integrated into the virtual distributed laboratory at the Research and Education Center "Intelligent Robotronics".

The main parameters of AMUR are as follows. The weight is 17–23 kg, the dimensions are L580xB510xH340 mm, the velocity is 10–500 mm/s, and the autonomous operation time is 1–2 h. AMUR consists of a caterpillar or two wheels (differential DC-drives), a TV (1, up to 6), video servers, 50–3000 mm ultrasonic locators, odometers, onboard battery control and charging devices, four analogue and 16 in/out–on/off channels, power keys PWM/PID regulators 2x10A/12-32V, mPC (form-factor PC104), on-board NB and remote PC with Wi-Fi for a maximum distance of 50 m or 1 km (with additional antennas), a side Wi-Fi TV-module ("satellite") with PTZ (optical 10x). AMUR operates under Linux and has Python, C++, and a distributed technology for programming all video data processing and sensors integration with logical filters on the remote PC and/or onboard. Real time data train-files (DTF) reflect all the work details during weeks/months of the functioning of mobile robots. The DTF is not a simple writing of data into files; it provides a synchronization of data, avoiding the problems of data transfer delays and gaps in the data flow by logical filters.

4. The technique for the implementation of control algorithms

In addition to educational features, the mobile robots make it possible to formulate tasks for creating SLAM algorithms for Robotino and AMUR robots and then transform for extreme robotics. Specifically, the following new algorithms were proposed:

- Supervised control of mobile robots by a human operator to compensate the effects of network delays by a continuous transformation of the control program, rather than using instruction passing;
- Finding the coordinates of the Robotino mobile robot by comparing sensory data from odometers and interpreting the indications (logical filters) from infrared and ultrasonic sensors;
- Compilation of the results of sensory data processing and data from a constantly adjustable terrain map;
- Algorithms for group communication and activity synchronization in a group of operating mobile technological robots on the basis of modified Petri net mechanism and expert rules;
- Identification of markers for navigation by the real-time computer vision system (CVS).

These algorithms were also used for robots operating in extreme conditions. Fig.2 and Fig.3 illustrates the operation of the computer vision system on such a robot. The software implementations can be found in [2–4].

Below, we present the results of our investigations—the basic principles for software and algorithms: (a) the results of comparative tests of programming languages allowed us to use interpreting Python in real-time systems; (b) we came to Turing complete protocols for robot control instead of specialized protocols or graphical interfaces (GUI, like MS Robotic Studio, etc.);

Fig. 2. The class-room of AMUR mobile robots in the "Stankin" University, Moscow.

Fig. 3. Record of logs in the machine vision system designed for a real-time system with 6 video-data streams on the adaptive (signal-controlled) network with mobile nodes and distances of up to 1 km.

(c) all the data flows are organized in train-type data (DTF) normalized by fast recurrent logical filters (<10 arithmetical operations per 1 record), which makes interpolation/extrapolation and filtering up to 60% of the non-correct data from remote sensors and control signals with net delays [4, 6]; (d) a friendly interface for non-qualified users (students) with only two operations: read/write into array/name (special train-data structure) from any program to get any data about sensors or control signals in any robot in the AMUR network (Fig. 1) any time without renewing problems or net delays and identification processes; (e) decomposition of control tasks into processes (separated algorithms) as the key for success in collective work on programming or fast prototyping (10 times faster than creating software through the standard technology, see also [3–7]), or generation of new tasks for the education of students.

5. Decomposition of the control task

Now, we present the scheme of interactions of main control processes (on the upper level) from the functional point of view and determine the interrelations of algorithms/processes and data flows between them. We do not see a logical alternative to such formalization on this level of abstraction in the software description.

Assume that *X(t)* is the operating space of a mobile robot and the data set completely controls the functioning of a particular robot with remote sensors. We call the operating environment one-, two-, or three-dimensional, depending on what kind of robot motions prevail: linear motions, or motions on a plane adequately represented in the form of a flat map, or motions in 3D terrain (for example, in the case of manipulators or underwater robots). It should be noted, that the dimension of the space of the generalized coordinates *X(t)*, defining the motion of robots as electromechanical systems, may be higher than 1, 2 or 3, depending on the design of these robots. However, in the coordinate system *X(t)* of the mobile robot, the locomotion problems are posed on a certain subspace with 1, 2 or 3 dimensions (we call it the subspace of the locomotion task or the navigation subspace).

Let *Au* be the algorithm of forming control commands *U(t)* on the basis of internal representation *Mr* (model of dynamic capabilities of the robot and its parameters). The dimension of the control vector *U(t)* corresponds to the dimension of the vector of the generalized coordinates *X(t)*, i.e., it is supposed that a part of the generalized coordinates of robot is controlled and the remaining coordinates are either calculated in the virtual model of robot *Vr* (using the solution of differential equations of the MR motion) or changed during the motion of the real mobile robot in accordance with its kinematic constraints/ties.

Let *As* be the algorithm of processing the measurements and formatting the results of determination of *Xs(t)* from primary measurements *Ls* (accumulated into data train-files DTF or images), which has been filtered by using the model of sensors *Ms*.

By *An*, we denote the navigation algorithm for determining the position *Xm(t)* and formation of the target function *Xp(t)* on the basis of *Xs(t)* and the map of external environment *Me* (stored in the robot memory, reflecting real environment *Ve*). *Xp(t)* can be input data for algorithm *Au* through the interface *Vi* from the operator *Mu*.

Let *Vs, Vr,* and *Ve* be the virtual representation of sensors, robot, and environment, respectively, for the substitution of real devices, when the robot algorithms are tested with the help of mathematical simulation or training activities. In the case of real-world activities, *Vs* and *Vr* are the drivers of remote sensors and robot, respectively, and *Ve* is directly the real environment. The real location *X* of the robot in the space of its generalized coordinates is not directly accessible to the control system (only through the sensor system).

Let *Xp* be a purposeful program trajectory either formed through the interface of integrating software *Vi* with the operator of supervised control *Mu* or calculated by the algorithm *An* according to the data *Xm* (for example, we used a rule-based formation of this trajectory [1]).

Assume that *t* is the current time and *dt* is the time step: *tk* = *t–k dt*, where *k*=0, 1,…, correspond to the times of observations and transfer of digital control commands. Let *Ms, Mr*, and *Me* be data (structures of data, models, and parameters) providing the input of corresponding algorithms.

The algorithms/programs and data flows are related in the following way:

X(t) = Vr(Ve, U(t))
Ls(t) = Vs(Ve, X(t))
Xs(t) = As(Ms, Ls(t, t-dt, ... ,t-k dt))
Xm(t) = An(Me, Xs(t)), Xp(t) = Vi(Mu, Xm)
U(t) = Au (Mr, Xm(t) - Xp(t)), GOTO *X(t).*

This diagram contains 3 cycles: the main cycle (*X,…, U, X,…*) and two asynchronous cycles in *Vs, Vi*—the robot interaction with the sensor system and with the operator (or with software, automatically changing the targets or mission plans).

The AMUR mobile autonomous robots make it possible to efficiently simulate the behavior and integration problems, develop suitable software, and formulate and solve optimization problems. The construction of algorithms for mobile transport robots (MTR) is based on the technology of expert rules, which converts the structure of algorithms, maximizes the quality (by proposed criteria), and compensates the omitted measurements due to the work of logical filters converting asynchronous data into conditionally single-moment measurements. This allows MTR to keep their functionality with any distant data and provide at least an assessment of their locations (or even accurate trajectory measurements), ensuring that the data supply is sufficient for solving locomotion problems. We propose the following approach to constructing expert control systems of MTR.

Let $\Psi m(.)$ be the system of rules for threshold discretization of variable (.) over several intervals 0,...,*m* and $Res_m(.)$ be the distance to the nearest value in accordance with the lower threshold. We define a discrete operational environment $Q = Me(i,j)$ as a vector field, where $i = \Psi m\ (Xn|x),\ j=\Psi m\ (Yn|y),\ x,\ y,\ Xn,\ Yn$ are the horizontal coordinates of MTR and their navigational estimates, $Q = <z, \Psi m\ (Xs), \Psi m\ (U)>(i,j)$ is the two-dimensional chain with vector values. Its local map containing the list of data (decisions of target/mission changes based on the current navigation data), controls signals *U* and conducted measurements *Xs*. This is the basis for the method of solving SLAM-type problems. The "recursive" use of rules $\Psi m(Res_m(.))$ makes it possible to obtain multi-level threshold schemes.

Then, the expert scheme for making the control decision is constructed as follows:

1. Threshold discretization of sensor readings $\Psi m(Xs)$ is constructed; for example, for *m*=2 the readings (estimates) are classified in terms of "under-", "norm-", and "over-", or in a similar semantic interpretation;
2. Estimation of possible actions (*B, S*) is constructed on the basis of production rules:
   (*B, S*): Logical expression {$\Psi m\ (Xs)$ = ..., neighborhoods *Me (i,j)=...}* => estimation;
3. Sets of chains-actions are generated (i.e., movements *B* and control measurements *S* for given positions of MTR < (*B, S*), *Me(i,j)*> with highest estimates at each step);
4. Production rules are applied to estimate the efficiency of neighboring successive actions. The sets of actions are ordered with respect to *Max* (estimates) on the basis of rules for convolution of estimates;
5. The first (best) action from the set < (*B, S*), *Me (i,j)*> is executed and then the above-listed rules are repeated for the new position of MTR.

6. "Intelligent robotronics"

Now we'll consider our didactic approaches on the example of a recent experiment with three groups of students from two technical universities. These students had a minimum experience in programming and never used Python, but in two months they built practically new mathematical models for simulation of mobile robots in an unknown environment, using different combinations of remote sensors (ultrasonic and IR locators or TV-cameras) and the decomposition, described above. The functions implemented by them simulate the SLAM algorithms and data flows between sensors and making decision modules.

Our main didactic tasks were: (1) Involve the students into the world of scientific and practical disciplines named "Intelligent robotronics" (this method is described below); (2) Give them an experience of "knowledge engineers"; (3) Write and verify simulation models of mobile robots (final programs) on Python. The Python interpreter has a lot of advantages for our approaches, as described before.

What do we call the "Intelligent robotronics" (IAR)? IAR denotes the shortlist of methods related to the integrated use of robotics, sensorics, mechatronics, electronics (oriented to mechatronics), and methods of intelligent systems or artificial intelligence, applied to data processing in the feedback loops with delays (due to remote sensors or network specific effects) and to planning/navigation systems in robotic devices. Thus, for education purposes, we collected the description of these methods into lectures and laboratory practice: two kinds of training activities for students and PhD students (a total of 7 ECTS). The size of the course can be corrected in either side ± 4 credits. The course is based on the Russian scientific school IAR.

The objective of the course is to teach students to use the main concepts, approaches, and models of sensorics and mechatronics, as well as to study the regularities of modern methods of control; acquire skills in designing and finding applications, building models that arise in engineering, and perform computations with these models. The main goal of the course is to conduct a practical study of the foundations of information-sensor technologies for solving creative and typical problems in future scientific or industrial fields.

The main didactic units (chapters) based on our own works and experiences are listed below.

Contact sensors (force-torque and tactile sensing; internal data sensors in robotics): feedback concepts, specific features of remote and contact sensors; sensor types: tensometry, principle of force-torque measurement; multicomponent sensors, tactile matrix and its implementation into real-time computer networks; pulse and code sensors for position and velocity measurements, Hall sensors, current and voltage measurements in control systems (CS), environment monitoring (pressure, humidity, temperature, etc.).

Remote sensors (sensor–locators and computer vision systems): acoustic location systems (models and methods of data processing; Fresnel–Kirchhoff models and their investigation; embedding filtered data flows from sensors into tracking/servo systems, into PID-regulators; triangulation and navigation methods); optical and laser range sensors (processing of discrete-range measurements conducted from mobile platforms for approximation of 3D objects); radar, electromagnetic, pneumatic, and infrared location sensors, pyrometers, side-view locators; conventional and non-conventional computer vision systems (classification of sources of matrix images – CCD, IR, light-sensitive memory, gamma locators; properties and synthesis of optical elements; conventional image processing algorithms – Huckel-like operators, chain code for contour description; stereo vision, photogrammetry); models of human vision and bionic approaches to designing image processing algorithms (human and insect – constancy, overexposure, gray field; inspection vision; segmentation and histogram analysis of color images); Technology of intellectualizing the sensor-control systems (knowledge representation, automatic inference in predicate calculus, heuristic methods): methods of knowledge representation (graphs, Petri nets, flows in networks; production rules, frames, reduction to disjunctive normal form), new and classical expert techniques for SLAM tasks and group control; Prolog-like descriptions, construction of inference mechanism and Herbrand's theorem; Herbrand-based and heuristic algorithms, A* optimal search in the space of states; new types of logic and deductive procedures.

Design of program-controlled autonomous systems (by examples of mobile robots and manipulators), intellectualization of sensor-control systems: control systems of mobile robots with AI elements (classification of ground and underwater robots; dimension of the operating environment and mathematical simulation, design of simulators; expert schemes of coordination of information-motion activity); intellectualization of supervised sensor-control systems based on wireless internet-technologies, control methods with unpredictable delays; manipulator control (automatic assembly in engineering industry: avoiding 10–100x difference in accuracy, specification/construction of operation and testing graphs; direct and inverse kinematic problems in CS for welding, laser cutting, and painting; machine processing of parts by force manipulators).

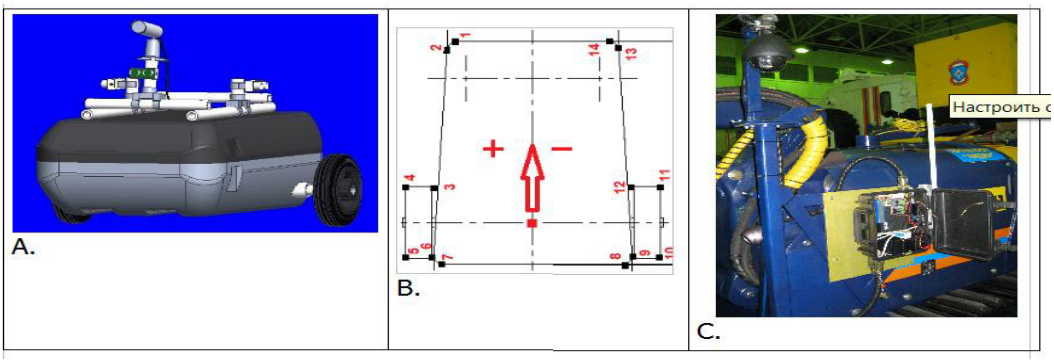

Fig. 4. "Knowledge engineers" (students) find the key points for three types of robots

Some results/examples of the student laboratory practice (as part of the work in the "Intelligent robotronics" course) are shown in Figs. 4–6. Figure 4 demonstrates principally different dynamical models of AMUR-105: (A) 3D-model and (B) plane model. The arrow (on B) indicates the relative position of the center of mass and center of rotation, while the control signals have the same value but the opposite direction of rotation. For Robotino, these centers merge in a single point. For caterpillar-type robots of Brokk-110D (our CCVS is shown on C) and AMUR-107, the center of rotation is unstable and can move along the arrow unpredictably.

Points 1–14 determine the geometry of robots (Fig.4) and can be minimized depending on the robot type and control algorithms used in the simulation model. For example, the parking conditions are absolutely different; for caterpillar robots, points 3, 4 and 11, 12 have to be moved forward to the level of line 1–14.

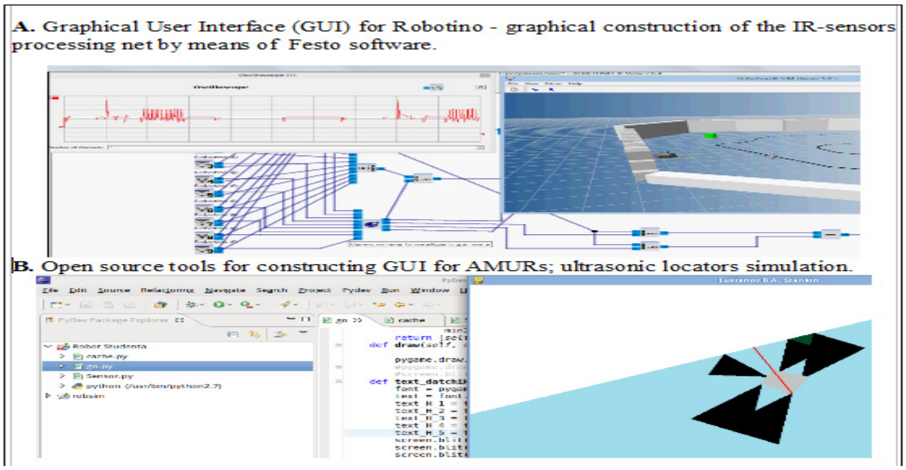

Fig. 5. Determination of key parameters and sub-optimal positions of 3–5 ultrasonic (B) and 9 IR locators (A) mounted on the robot.

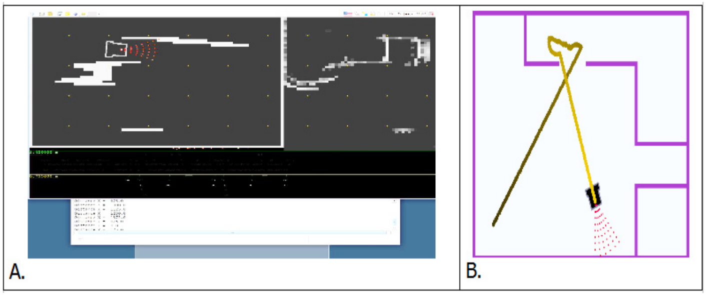

Fig. 6.Examples of GUI: two path finding algorithms for robots with ultrasonic locators (simulation).

Another investigation is shown in Fig. 5. The problem is to determine the parameters and positions of 3–5 ultrasonic (B) and 9 IR locators (A) mounted on robots. The simulation and modeling indicates equal sub-optimal results. The international laboratory "Sensorika" standardized the software development kit (SDK) for AMUR and other robots (both for simulation and modeling) in the form of a virtual box, that can be set over all main operating systems and also integrates other software tools; for example, the Robotino software (A). The final example of simulation refers to models of ultrasonic locators. The results are shown in Fig. 6. These experiments make it easy to understand the difference of the environment model and its reflection into the robot "brain", depending on the sensor model. (A) The left panel shows the environment matrix model of obstacles available for robot control system only through a sensor probability model (right panel), (B) the same algorithm with the vector environment model and with sensor processing, based on expert rules for doors finding control algorithms.

Conclusion

We have developed efficient technological hardware and software, that can be used for solving a wide range of SLAM-type problems, including the determination of mobile robot trajectory coordinates, the approximation and interpretation of indications of ultrasonic, infrared and other sensors, the identification of the characteristics of different sensors and construction of their mathematical models, and the construction of algorithms for sensory data processing and adequate terrain maps. This basic soft is not critical for experiments with only a single robot but is practically obligatory for distributed mobile robotic systems and groups of MTRs. The construction and implementation/modernization of these algorithms is an efficient tool for the training of students in computer sciences, mechatronics, and sensorics.

Acknowledgements

This study was supported by the University of Zadar, the Russian Academy of Engineering, project "Intelligent Robotronics", partially by the Russian Foundation for Basic Research, projects 13-07-01032 and 13-07-00988, and by the Russian Government, project no. 14.604.21.0054, 30.06.2014.

References

[1] V.P. Andreev, V.E. Pryanichnikov, and E.A. Prysev, 'Multi-access control of distributed mobile robotic systems based on networking technologies,' in Annals of DAAAM for 2010 & Proceedings of the 21st International DAAAM Symposium, B. Katalinic, Ed. Vienna, Austria: DAAAM International, 2010, pp. 15–16.

[2] V.E. Pryanichnikov, V.P. Andreev, 'Intellectulazation of special mobile robots, including return algorithm to a zone of stable RC' in Proceedings of the XXI Int. Conference on Extreme Robotics, St.Peterburg: Poliectnika-service, 2010, pp. 46–49.

[3] V. Pryanichnikov, V. Andreev, K. Kirsanov, et al., 'Technologies of modernization and sensing of special mobile robots', in Annals of Southern Federal University (Techn.sc.): Taganrog, 2011, no. 3, pp. 166–171.

[4] V.E. Pryanichnikov, V.P. Andreev, E.A. Prysev, 'Group control of mobile robots, based on the net-technologies', in Proceedings of the Int. conference on Robotics for Security & Space (Planet & Earth rovers), St.Peterburg: Poliectnika service, 2010, pp. 279–283.

[5] B.Katalinic, V.E. Pryanichnikov, K. Ueda, I. Kukushkin, P. Cesarec, R. Kettler, 'Bionic Assembly System: hybrid control structure, working scenario and scheduling', in Proceedings of 9th National Congress on Theoretical and Applied Mechanics, Brussels: 2012, pp. 111–118.

[6] V.E. Pyanichnikov, 'Algorithmic base for remote sensors of mobile robots', in Mechatronics, Automation, Control, 2008, no. 10 (91), pp.10–21.

[7] K. Kirsanov, B.Levinsky, V. Pryanichnikov, 'Integrating software for intelligent robots', in Informational-Measuring and Control Systems, Radiotechnika, 2009, Vol. 7, no. 6, pp.35–43.

[8] A.A. Frolov, E.V. Biryukova, P.D. Bobrov, O.A. Mokienko, A.K. Platonov, V.E. Pryanichnikov, 'Principles of Neurorehabilitation Based on Brain-Computer Interface and Biologically Plausible Control of the Exoskeleton' in Human Physiology, 2013, Vol. 39, No. 2, pp. 196–208. Original Russian text published in Fiziologiya Cheloveka, 2013, Vol. 39, No. 2, pp. 99–113.

[9] M.K. Holden, T.A. Dyar, L. Schwamm, E. Bizzi, 'Virtual environment based telerehabilitation in patients with stroke', in Presence: Teleoperations and Virtual Environments, 2005, Vol. 14, p. 214.

[10] V.V. Klyuev, Yu.A. Kondrat'ev, D.E. Okhotsimsky, E.P. Popov, V.E. Pryanichnikov, et al.,'Sensor Systems and Adaptive Industrial Robots', in monograph: Mashinostroenie, 1985, Chapters 3–5, pp. 56–172.

[11] D.E. Okhotsimsky, V.A. Veselov, A.K. Platonov V.E. Pryanichnikov, 'Educational Modular transport robot with PC control', in Proceedings of the IV Int. conf. on Flexible production systems, Moscow: Int. R&D Institute on the Control Problems, 1987, pp. 62–64.

[12] ROS: http://www.ros.org. Accessed: 22.09.2014.

[13] Microsoft Robotics Developer Studio. – URL: http://www.microsoft.com/en-us/download/details.aspx?id=29081. Accessed: 22.09.2014.

[14] W. Emmerich, M. Aoyama, J. Sventek, The impact of research on the development of middleware technology, ACM Transactions on Software Engineering and Methodology. New York: ACM, 2008, Vol. 17, no. 4. pp. 19–48.

[15] V. Andreev, S. Kuvshinov, V. Pryanichnikov, Yu. Poduraev, Education on the basis of virtual learning robotics laboratory and group-controlled robots. 24th DAAAM International Symposiumon Intelligent Manufacturing and Automation, 2013, Procedia Engineering, 2014, Vol. 69, pp. 35–40.

[16] Scientific-educational distributed laboratory - software and hardware means // V. Pryanichnikov, V. Andreev, P. Bobrov, E. Biryukova, A. Frolov, K. Kharin, K. Kirsanov, A. Kostin, S. Kuvshinov, Y. Marzanov, and E. Prysev, Annals of DAAAM for 2012 & Proceedings of the 23rd International DAAAM Symposium, Vol. 23, no. 1, ISSN 2304-1382, ISBN978-3-901509-91-9, DAAAM International, Vienna, Austria, EU, 2012.