THE 2-WHEELED MOBILE ROBOTS CAPABLE OF COOPERATING IN GROUP OF ROBOTS

Received 26th February 2007; accepted 20th March 2007.

Tadeusz Uhl, Tomasz Buratowski

Abstract:

One of the basic problems connected with wheeled mobile robots is the choice of the appropriate navigational method. This refers to individual robots and also, in particular, the group of mobile robots. "Navigation" can be defined as an issue connected with systems control, defining their position and orientation as well as choosing appropriate trajectory. The consisting elements of mobile robots navigation are: self-localization, path planning and creating a map of the explored area. Each of the above enumerated elements is independent; however, the realization of the target navigation task of the mobile robot or the group of mobile robots depends on the correct construction of particular elements.

Keywords: *mobile robot, wireless, bluetooth, navigation*

1. Introduction

Now the main problem in robotics is to control more than one robot at the some time. We can distinguish a swarm of robots and a group of robots. If we talk about a swarm it means that we have more then 100 robots to control. The group of robots consists of maximum 100 robots. In this paper we are trying to present wheeled mobile robot project capable of cooperating in a group. We decided to build 3 robots capable of communicating and sending data in the process of navigation. These robots can be used in indoor localization.

In these mechatronic systems we have several problems to solve. The first problem is connected with communication between robots, also to find the robots' position by themselves is an essential problem but it is not easy to solve.

2. 2-Wheeled mobile robot description

A. Basic assumptions connected with kinematics model

If we look into the reference connected with 2-wheeled mobile robots, we can say that for indoor solutions this type of mechatronic systems is treated as nonholonomic [1,2,3]. This nonholonomic condition has been presented below and is connected with point A in Fig. 1:

$$\dot{y}_A = \dot{x}_A \tan(\Theta) \tag{1}$$

Also we are assuming that there is no skid between the wheel and the ground [2]. The basic elements of the model are: wheel unit drive of wheels 1 and 2, self-adjusting supporting wheel 3 and the frame of unit 4 with all electronics. When describing motion phenomena for a complex system such as mobile 2-wheeled robot, it is beneficial to attach the coordinates system to the particular robot's elements. This mechatronic system has 2DOF (degrees of freedom) it means that its position should be presented by coordinates x, y and the orientation is represented by angle . In Fig. 1 the mode of calculation focused on the system of kinematics description has been presented [1,2,3].

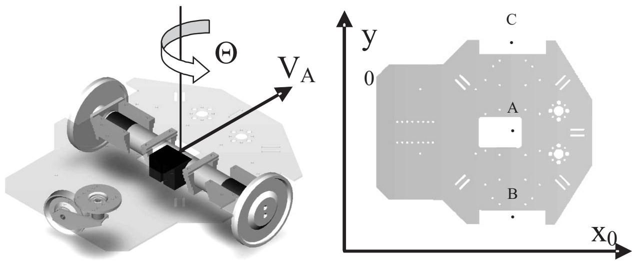

Fig.1. The mode of calculation connected with system kinematics description.

System coordinates have been attached to the robot basic element. The system coordinates $x_0\ y_0\ z_0$ is motionless and based. If our kinematics model is defined in this way it is possible to describe the position and the orientation of any point on our robot. As an example characteristic points B and C connected with wheels can be described with the use of Denavit-Hartenberg notation as follow [2,3]:

$$T_{B,A} = \begin{bmatrix} \cos(\Theta) & -\sin(\Theta) & 0 & x_A + l_1 \sin(\Theta) \\ \sin(\Theta) & \cos(\Theta) & 0 & y_A - l_1 \cos(\Theta) \\ 0 & 0 & 1 & r_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$T_{C,A} = \begin{bmatrix} \cos(\Theta) & -\sin(\Theta) & 0 & x_A - l_1 \sin(\Theta) \\ \sin(\Theta) & \cos(\Theta) & 0 & y_A + l_1 \cos(\Theta) \\ 0 & 0 & 1 & r_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

Where l_1 stands for a distance between points A,B and A,C. Also parameters r_1,r_2 this are particular wheel radiuses.

B. Basic assumptions connected with path planning

The basic matter in the issue of the wheeled mobile robots (and not only) trajectory planning is describing basic definitions connected with an object moving in space. The basic law connected with trajectory planning may be defined as follows [4]:

$$FS = W - (\bigcup C_i) \tag{4}$$

and trajectory $c \in C^0$ is defined as: $c\!: [0,1] \rightarrow FS$
where $c \rightarrow p_{start}$ and $c(1) \rightarrow p_{target}$

If we take into consideration kinematics constraints for our particular mobile robot presented above there are 2 possible schemes of motion on our defined surface [2,4]. The first possible motion scheme is connected with metric L1 which is described as follows:

$$(x, y): |x| + |y| = const \tag{5}$$

The second possible scheme of motion is described by metric L2 defined as:

$$(x, y): x^2 + y^2 = const \tag{6}$$

If we present those metrics in the graphical way we receive presentation described in Fig. 2:

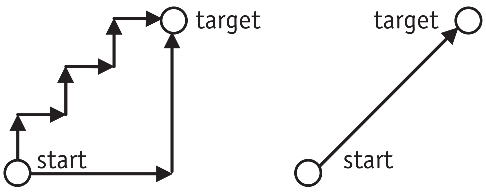

Fig.2. The graphical presentation of metrics connected with robot motion scheme.

In this project L2 metric has been assumed. This mode of motion is the most efficient and will be used in the process of path planning. According to the scheme of motion assumption in Fig. 3 robot movement has been presented.

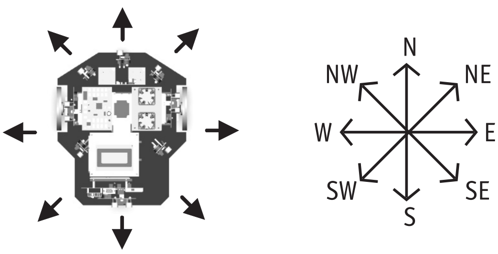

Fig.3. The possible movement for particular robot.

According to a basic definition connected with robots, basic definition of those mechatronic systems describes them as autonomous devices owning 3 units presented in Fig. 4. The first unit is connected with power supply. In this unit we have accumulator, converters and charger. The Robot's controller still controls voltage level and sends the robot to the docking station. The *MOTION UNIT* is connected with mechanical construction and it is mobile platform equipped with wheels, two driven and one self-adjusted, and drives MAXON A-max 32 with planetary gear GP 32 A and optic encoders HEDS5540.

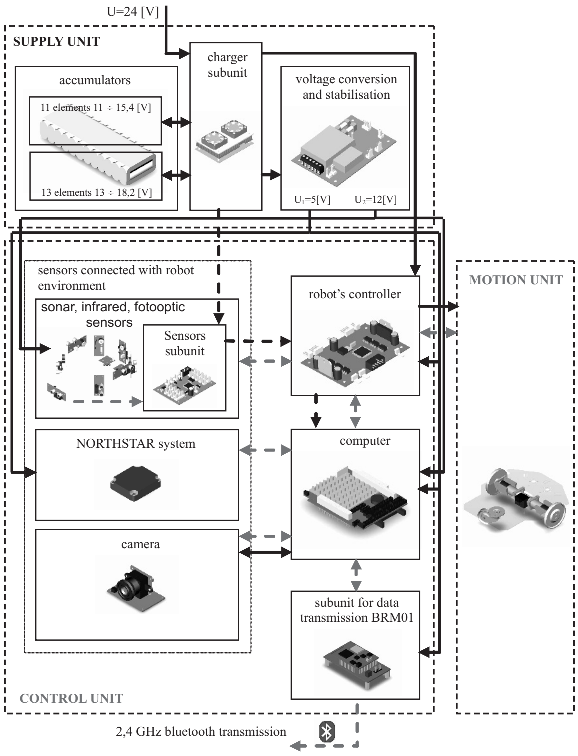

Fig.4 The Block diagram of 2-wheeled mobile robot

The CONTROL UNIT is the most complex. In this unit there are infrared, sonar and optic sensors controlled by sensors subunit All those sensors are connected with robot environment. For sensors work the robot controller is based on ATMEGA128. Additionally the robot is equipped which embedded computer PCM3370 which is responsible for data acquisition and managing control, because this mechatronic device should cooperate in group each robot wireless communication device based of bluetooth technology. In this project hierarchical control system for particular robot has been assumed. The basic description connected with this kind of control has been presented in Fig. 5.

The managing control is the highest control level. It includes special algorithms and artificial intelligence elements based on swarm behavior. Those algorithms allow computing sensor information and recognizing situation in robot environment. It means that the received information is interpreted and appropriate action is taken.

Another control level is called strategic and is responsible for dividing global tasks on the series of basic tasks. At this level path planning is calculated in Cartesian coordinates. In this project we decided to create simple robot's language based on basic assumptions connected with path planning. For example if our robot should go straight 1 meter, the command connected with robot motion is !NN100/r. On tactical level of hierarchical control system robot position is mapping from Cartesian on joint coordinates. At this level the control system is using sensor information (encoders) to estimate appropriate position. The main task of the control system on executive control is to drive control in order to execute tasks planned on strategic and tactical levels. In this project for managing, strategic control and robot's language embedded computer is responsible. Especially designed robot's controller is responsible for tactical and executive control. In our project each robot from the group of three is an agent. An agent [5,6,7] is an animate entity that is capable of doing something on purpose.

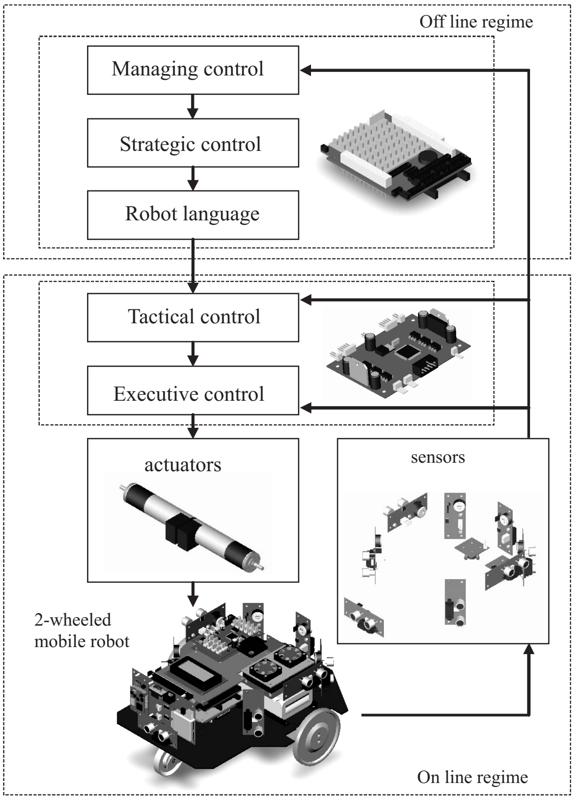

Fig.5 The hierarchical control system for mobile robot

That definition is broad enough to include humans and other animals, the subjects of verbs that express actions, and the computerized robots.

In the presented group of robots each agent is equipped with ORACLE XE database installed in embedded computer. The database is a very important source of agent environment information. Each robot receives and sends information to another with the use of the scheme presented in Fig. 6. The communication master agent initiates communication and sends information from slave 2 to agent slave 1. Slave agent 1 sends information about itself to the master agent. Master agent sends information from slave 1 to the agent slave 2 and then slave 2 sends data about itself to the master. This is one cycle. After the assumed cycle quantity master sends information about agent's positions and obstacles to the base station.

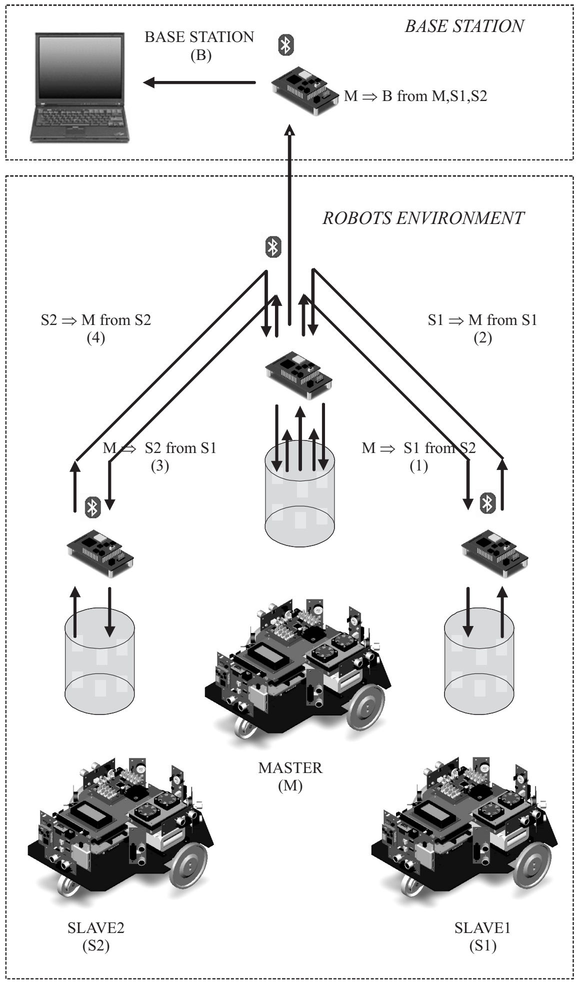

Fig.6 The scheme of wireless communication between robots

3. Navigation

Very often the boundary between machine and environment is difficult to describe, but generally there is assumption that the machine interacts with the environment through sensing and actuation [1]. It is possible to present it as a simple relation (Fig. 7).

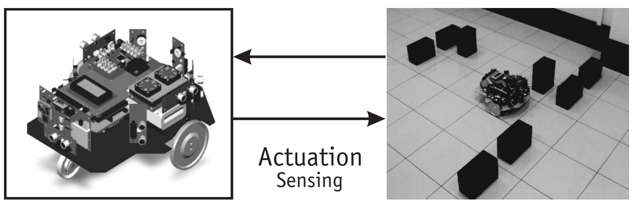

Fig.7 The boundary between machine and environment.

In the process of navigation there are two main problems to solve self-localization and path planning.

A. Self-localization

Self-localization of a robot in space consists in defining its position and orientation in reference to the base coordinates. “Base coordinates” should be understood as constant base coordinates. Defining position and orientation takes place on the basis of data connected with a robot's state parameters and the robot's state surroundings parameters as well as a proper interpretation of the data stored in the robot's memory. Generally we can say that there are several ways of a method of self-localization classification. Examples of such methods are local and global methods [9]. Local methods consist in calculating the position and the orientation of a robot in reference to its prior localization, while global methods are based upon defining position and orientation without knowing the prior system localization. Another classification of self-localization methods is based upon the environment description. We may differentiate static environment, in which only the robot is moving and dynamic environment, in which other systems are also moving and the robot is able to detect this motion. As far as self-localization is concerned, a very important role is played by the methods of measuring the robot's position and one may differentiate relative methods (the measurements are done by means of sensors placed on the robot) to which we may include odometric methods consisting in calculating relative vehicle movement on the basis of the measurement of angle rotations of driven wheels, as well as inertial methods consisting in the usage of accelerometers, gyroscopes for speed and acceleration measurement. There also is a possibility of using absolute methods of measuring position and orientation of a robot in space, to which we may include recognizing artificial or natural markers, i.e. defining position and orientation on the basis of markers whose position is known by the robot e.g. NORTHSTAR system.

B. Path planning

One of the basic needs in robotics is to have algorithms that can bee able to convert high level specifications of tasks from humans into low-level descriptions of how to move. The term path planning is often used for these kinds of problems.

In project connected with group of robots each robot have embedded computed equipped with ORACLE XE database required services and planner. The functional scheme of path planning for one robot has been presented in Fig.8.

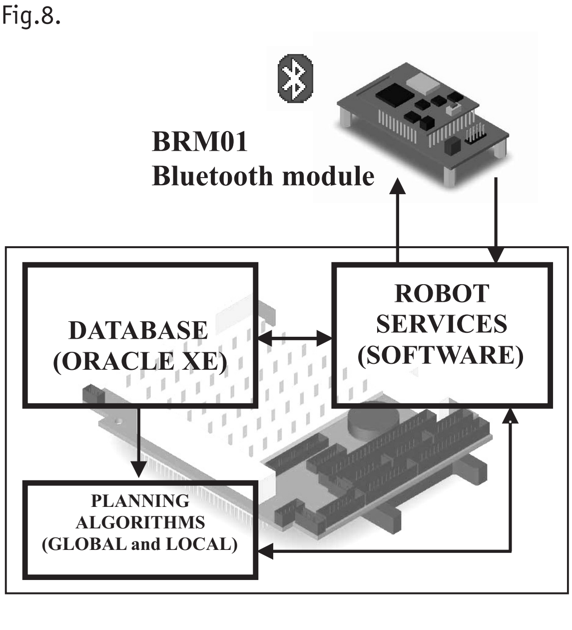

Fig.8. The functional scheme of path planning for robot

Each embedded computer PC104 (equivalent to PC Celeron 650 MHz) which is responsible for robot control on managing control level have the some parameters.

B.1 Planners

A planner is responsible for plan generation and its application. There are two possibilities, planner may be a machine or a human [9]. If the planner is the machine than we have planning algorithm. Sometimes, because of some reasons humans become planners and taking control of the system. In our project related with group of robots booth planner's possibilities have been applied. Human can send information about desired trajectory (each step should be defined) with the use of simple GUI or operator give information about start point and target point. In this solution robot autonomously should find shortest path with respect obstacle avoidance.

B.2 Plans

A planner generates a plan of motion for particular robot or for all robots from the group. In the presented project each robot is responsible for path planning but exchange information about obstacles and other robots position with the use of Bluetooth technology. According to the basic assumptions connected with kinematics model and scenario described above two types of plans have been taken in to consideration:

- Simple path algorithm

- Wavefront algorithm

Let we take into consideration exemplary scenario presented in Fig. 9.

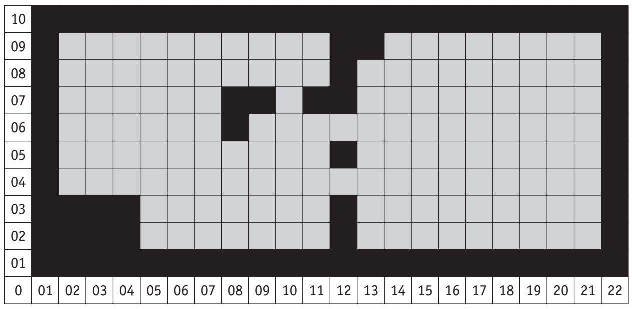

Fig.9. The exemplary scenario for robot.

The Simple path algorithm was designed to provide simple way of building user trajectories.

After user path assumptions it is required to check each defined step to find out that this trajectory is clear-cut and is appropriate with metric L2 .

For instance, exemplary user trajectory has been assumed and presented in Fig.10. Each cluster is defined by coordinate x and y. According to our laboratory dimensions every cluster is a square (300mm x 300mm).

Let us consider the possibility of next steps for those parts of robot trajectory. When we look at example in Fig. 10 where robot trajectory is straight, we have three possible moves for next robot step to expand trajectory and 2 which are corrected according to metric L2 but not expanding trajectory (Fig. 11).

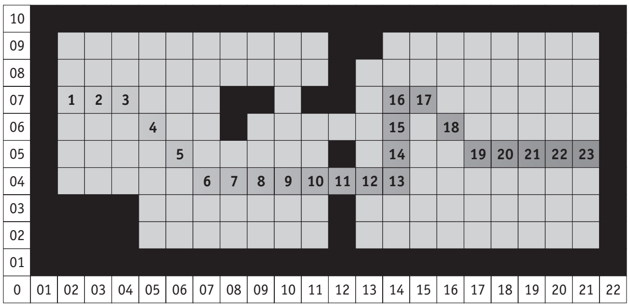

Fig.10. The exemplary user trajectory for robot

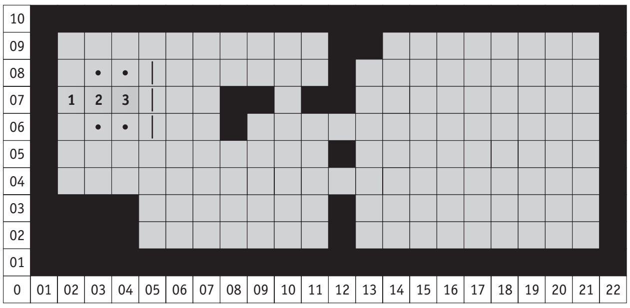

Fig.11. The first part of the exemplary user trajectory for robot

Next example shows a situation when robot is turning ninety degrees right and is going to cluster (7,4). For this move we can find also five steps, which are possible in the next move. Step marked by white dot is not considered due to reasons like above (Fig. 12).

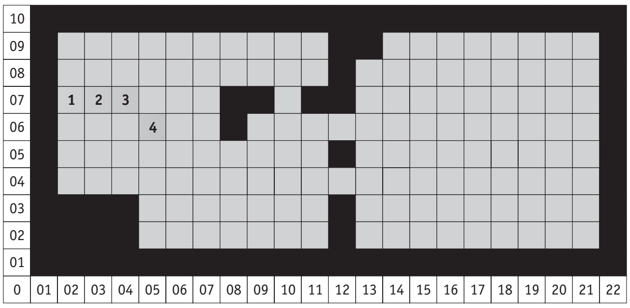

Fig.12. The second part of the exemplary user trajectory for robot

The third example shows the part of robot trajectory where robot is moving straight. Here we have five steps which are possible to reach in next robot step (Fig. 13).

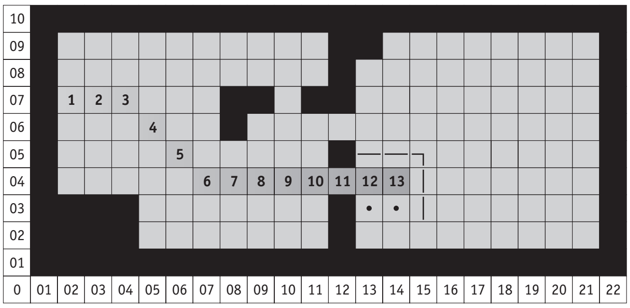

Fig.13. The third part of the exemplary user trajectory for robot

To continue assumed trajectory, robot should turn left 90 degree (!WW command). After this operation the robot should be in cluster (14,7) presented in Fig. 14.

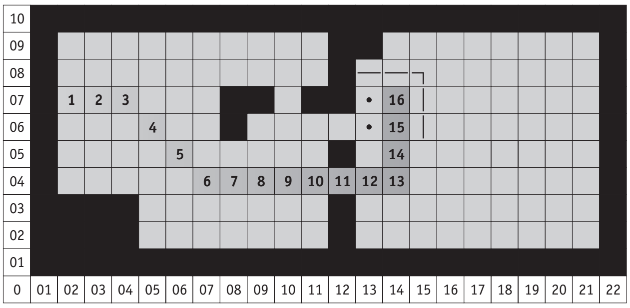

Fig.14. The fourth part of the exemplary user trajectory for robot

Now the robot should turn right 90 degree by the use of command !EE (Fig. 15)

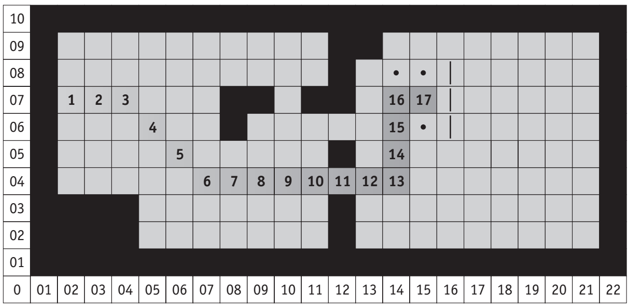

Fig.15. The fifth part of the exemplary user trajectory for robot

In this part of the robot trajectory there is situation similar to first part it means robot have three possible steps presented in Fig. 15 and should continue movement to cluster (17,5) presented in Fig. 16.

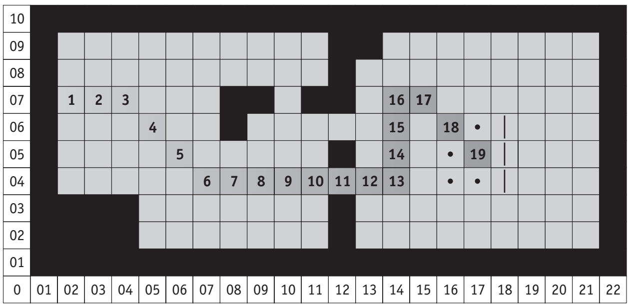

Fig.16. The sixth part of the exemplary user trajectory for robot

The seventh part of trajectory is connected with the mask presented in Fig. 16 after this check robot is going to cluster (21,5) presented in Fig. 10. To sum up, if our robot should reach assumed trajectory we need to check trajectory with the use of set of especially designed masks (Fig. 17) (there are sixteen masks which we have to consider to make our trajectory clear-cut) and than we need to make optimization connected with the cluster collection.

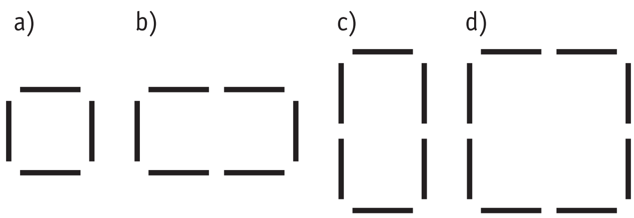

Fig.17. The set of especially designed masks

After optimization the set of commands required to move robot from start point to target point can bee performed in the form presented in table 1.

Table 1. The set of commands for assumed trajectory

	command	Θ [°]	x [cm]	y [cm]	x cluster	y cluster
1	!NN60	0	60	0	4	7
2	!NE127	-45	150	-90	7	4
3	!NW210	0	360	-90	14	4
4	!WW90	90	360	0	14	7
5	!EE30	0	390	0	15	7
6	!NE85	-45	450	-60	17	5
7	!NW120	0	570	-60	21	5

The second algorithm is called wavefront [9] and is very efficient in robot's path planning. This part is going to present automatic planner. There are 2 types of this mode of path planning, first is connected with 2 dimensional area and second operating in 3 dimensional space. For mobile robots solution first mode is used.

The wavefront algorithm involves a breadth first search of the graph beginning at the destination point until it reaches the start point. In Fig. 18 exemplary robot or group of robots environment has been presented and will be used in laboratory test rig.

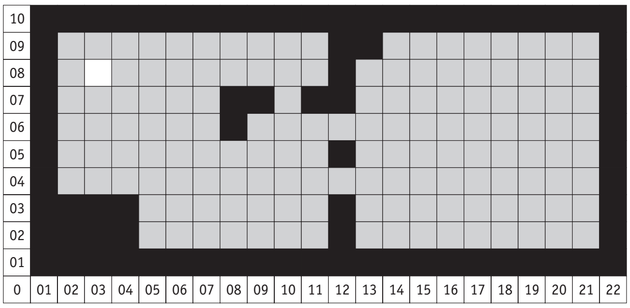

Fig.18. The exemplary scenario for robot.

At the beginning, obstacles are marked with a 1 and goal point is marked with a 2 (Fig. 19).

10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
09	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1
08	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
07	1	0	0	0	0	0	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	1
06	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
05	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	2	0	0	0	0	1
04	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
03	1	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
02	1	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
01	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22

Fig.19. The binary representation of the robot scenario.

You can optionally surround the entire world with 1s as well as tell your robot to avoid those squares, and/or "expand" the size of the obstacle to avoid collisions due to dead-reckoning errors. After those operations our computational environment will look like this presented in Fig. 19. To create the "wave" of values, begin at the destination, and assign a distance of 3 to every square adjacent to the goal. Then assign a distance of 4 to every square adjacent to the squares of distance 3 (Fig. 20).

10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
09	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1
08	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
07	1	0	0	0	0	0	0	1	1	0	1	1	0	0	4	4	4	4	4	0	0	1
06	1	0	0	0	0	0	0	1	0	0	0	0	0	0	4	3	3	3	4	0	0	1
05	1	0	0	0	0	0	0	0	0	0	0	1	0	0	4	3	2	3	4	0	0	1
04	1	0	0	0	0	0	0	0	0	0	0	0	0	0	4	3	3	3	4	0	0	1
03	1	1	1	1	0	0	0	0	0	0	0	1	0	0	4	4	4	4	4	0	0	1
02	1	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
01	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22

Fig.20. The binary representation of the robot scenario after 2 iterations.

Continue to do this until you reach the start point. Once this is complete, you can simply follow the numbers in reverse order until you reach the goal (Fig. 21), avoiding any square with the value 1. You are free to chose if you would like your robot to be able to move diagonally or just in 4 directions, but obviously motion scheme in 8 directions is faster and more efficient.

10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
09	1	17	16	15	14	13	12	11	11	11	11	1	1	6	6	6	6	6	6	6	6	1
08	1	17	16	15	14	13	12	11	10	10	10	1	6	5	5	5	5	5	5	5	6	1
07	1	17	16	15	14	13	12	1	1	9	1	1	6	5	4	4	4	4	4	5	6	1

Fig.21. The binary representation of the robot scenario after 15 iterations.

A possible path has been marked on red and transformed from binary representation to real scenario (Fig. 22).

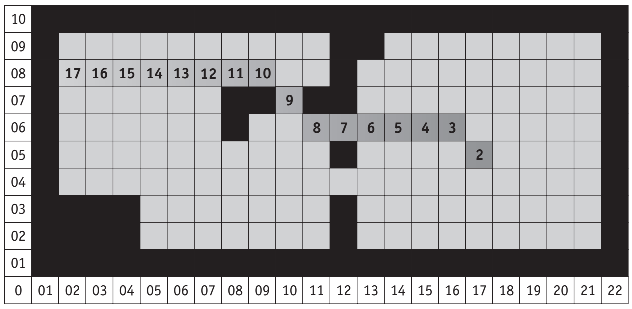

Fig.22. The robot trajectory in laboratory environment.

This is just one of the many paths that can be used to reach the goal, but is shortest, and any path that follows the descending numbers correctly will be acceptable. It means planner should check all solutions and find minimum time consuming trajectory. This trajectory should be performed in form presented in table 2.

Table 2 The set of commands for assumed trajectory generated with the use of wavefront algorithm.

	command	Θ [°]	x [cm]	y [cm]	x cluster	y cluster
1	!NN180	180	0	0	9	8
2	!NE127	-45	240	-60	11	6
3	!NW210	0	390	-60	16	6
4	!WW90	-45	420	-90	17	5

In our developing software we applied this algorithm as an autonomous planner for each robot from the group. An exemplary path calculated on the basis of this algorithm for laboratory scenario (Fig. 18) has been presented in Figs. 23 and 24. In Fig. 23 configuration software has been presented. This part of software is required to set all information from serial communication port plugged to robot.

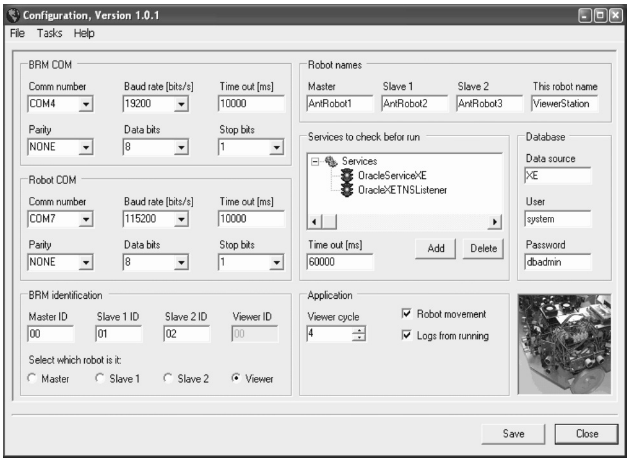

Fig.23. The configuration software.

The second part of software directly connected with autonomous planner based of wavefront algorithm give possible to plan trajectory according to the information from the database about obstacles (Fig. 23).

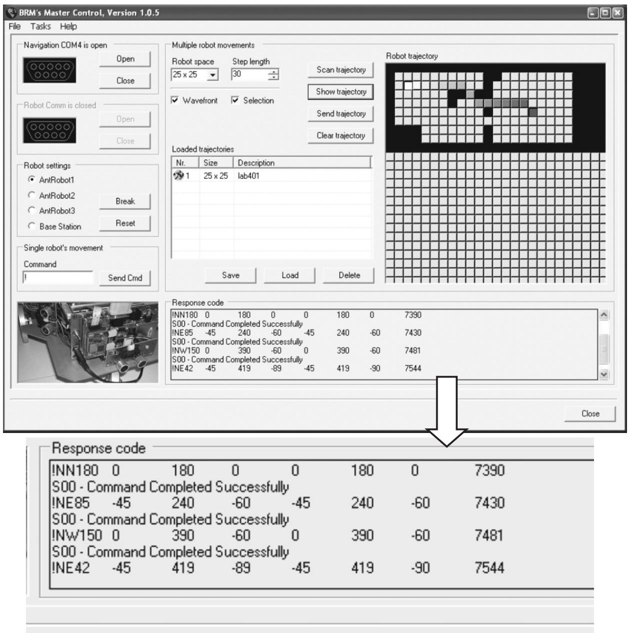

Fig.24. The software connected with autonomous planner.

Additionally this software enables robots to communicate by Bluetooth devices and to send data between robots (Fig. 6.)

After test rig made in especially prepared laboratory we see that the result is acceptable. It means that the robot moved to the target point and the error was not greater than 1 cm. The exemplary, significant trajectory points have been presented in Fig. 25.

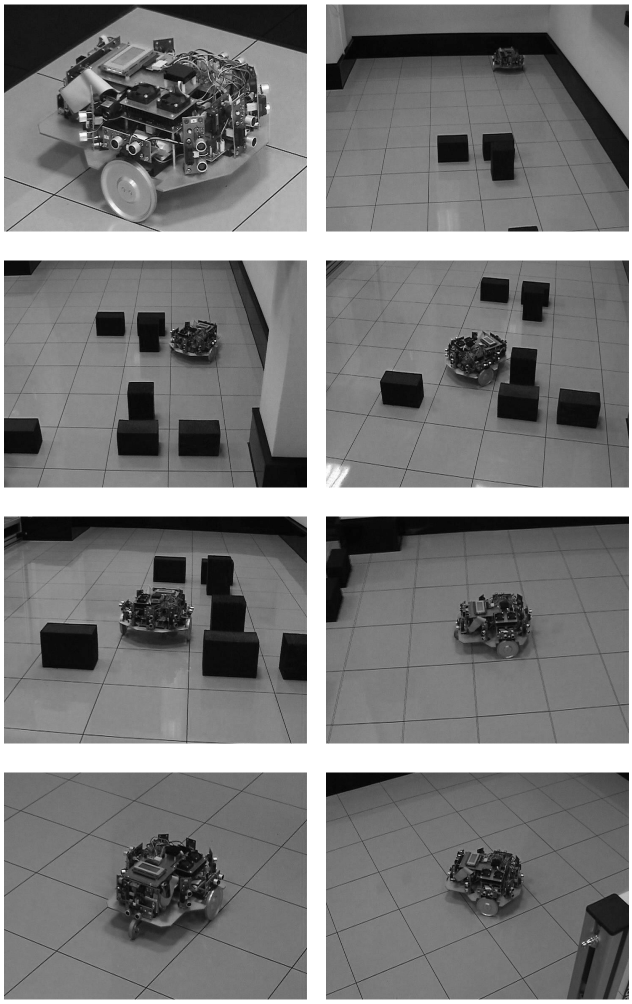

Fig.25. Test rig in laboratory environment.

4. Conclusion

Since, robots which are developed in our Department communicates by Bluetooth devices, we have to configure this devices. To configure Bluetooth devices, robot ports - where commands are sent and other required settings the special software have been developed.

The discussed settings are sufficient when we are talking about robot trajectories. When we turn on robots there are a couple of things we have to check to make them ready to run. For instance we ought to wait for start services which are required to correct robots runs. Our software utilizes Oracle database to store all information about obstacles, clusters and also running history, so services for which we have to wait are Oracle services.

The project is still developing. To sum up, current the robots gather information by means of sensors and pass them on the database. In the database this information is classified by means of SQL procedures. Than the data is passed on the particular robots. So the collaboration rule consist in the collective view of environment by which the robots are surrounded. Currently work are being carried out to find an algorithm which will divide the area of search to search for cluster of a different color. Now these points are indicated at random excluding the possibility of the robots trajectories crossing.

In Fig. 26 group of robots designed and manufactured in our Departments of Robotics and Mechatronics University of Science and Technology AGH (except embedded computer and bluetooth modules) has been presented.

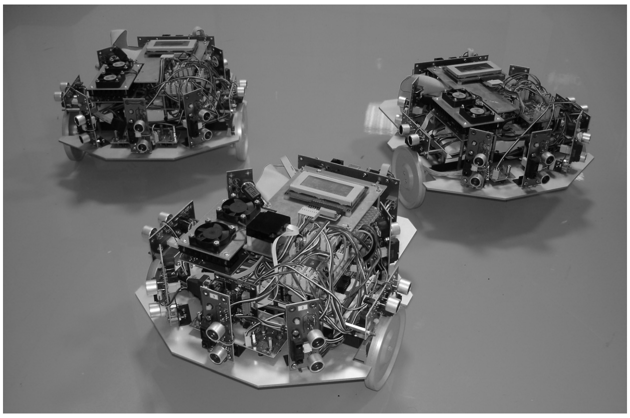

Fig. 26. The group of robots.

The result of interpreting data connected with robots environment is a geometric map. Equipping the robot with the possibility of generating a map lets for effective trajectory planning and self-localization in space.

This work is supported by Polish State Committee for Scientific Research under contract 4 T12C 046 29.

AUTHORS
Tadeusz Uhl - Department of Robotics and Machine Dynamics, Faculty of Mechanical Engineering and Robotics, AGH University of Science and Technology, Kraków, Poland, e-mail: tuhl@rob.wibro.agh.edu.pl.
Tadeusz Buratowski* - Department of Robotics and Machine Dynamics, Faculty of Mechanical Engineering and Robotics, AGH University of Science and Technology, Kraków, Poland, e-mail: tburatow@uci.agh.edu.pl.
* corresponding author

References

[1] T. Buratowski, T. Uhl, "The Project and construction of group of wheeled mobile robots", *Pomiary Automatyka Kontrola*, Warszawa, no. 5, 2006.
[2] T. Buratowski, T. Uhl, "The fuzzy logic application in rapid prototyping of mechatronic systems" in *Proceedings of the Fourth International Workshop on Robot Motion and Control ROMOCO'04*, Poznań, 2004.
[3] T. Buratowski, T. Uhl, W. Żylski, "The comparison of parallel and serial-parallel structures of mobile robot Pioneer 2DX state emulator" in *Proceedings of the Seventh Symposium on Robot Control SYROCO'03*, Wroclaw, 2003.
[4] J.J. Leonard, H. F Durrant-Whyte, *Directed sonar sensing for mobile robot navigation*, Kluwer Academic Publishers, Norwell MA, 1992.
[5] Konishi M., Nishi T., Nakano K., and Sotobayashi K., "Evolutionary Routing Method for Multi Mobile Robots in Transportation" in *Proceedings of the IEEE International Symposium on Intelligent Control*, October 2002.
[6] Abbass H.A., Xuan H., McKay R.I., "A New Method to Compose Computer Programs using Colonies of Ants" in *Proceedings of the IEEE Congress on Evolutionary Computation*, 2002.
[7] N. Baskar, R. Saravanan, P. Asokan, and G. Prabhaharan, "Ants Colony Algorithm Approach for Multi-Objective Optimisation of Surface Grinding Operations" in *Advanced Manufacturing Technology*, 2004.
[8] Shapiro S.C., *Encyclopedia of Artificial Intelligence*, NY: John Wiley & Sons, 1992.
[9] LaValle S. M., *Planning algorithms*, Cambridge University Press, 2006.