Reactive Control of a Mobile Robot Based on Neural Networks

Vladimir Golovko
Department of Computers and Mechanics,
Brest polytechnic institute, Moscowskaja 267,
224017 Brest, Republic of Belarus
cm@brpi.belpak.brest.by

Abstract

This paper describes the neural system for reactive control of the mobile robot. The neural system consists of different types of neural networks, which are combined in the intelligent system. The efficient techniques for the training of neural networks are considered. The main problem of the neural network is the robust control of the robot in case of inexact information from sensors. Experimental results are given in the paper.

1. Introduction

One of the most important problems in the design and development of intelligent mobile robots is the robust control, which is the ability of a vehicle to execute collision-free motions in case of inexact information from sensors. Such an approach permits to use inexpensive sensors and to adapt to different environments.

This paper describes the intellectual neural system for reactive control of the mobile robot. The inputs of such a system are the final goal position and the sensor system data. The ultrasonics and infrared scanner is used as sensors. The information from various sensors is combined by data fusion. As a result the local environment map is obtained. The intelligent neural system will process such a map and generate the direction and the velocity of motion. The neural system solves the following tasks:

Sensor data fusion

Building of the local environment map

Obstacle detection and definition of the free interval of motion

Definition of the optimal direction in the chosen interval of motion

The approaches described in this paper can be used for various mobile robots.

2. Architecture of a neural system

The common architecture of a neural system for autonomous control of the robot is shown in fig.1. The system consists of various types of neural networks. In the figure only the main links and blocks of the system are shown. Sensors location is shown in fig.2.

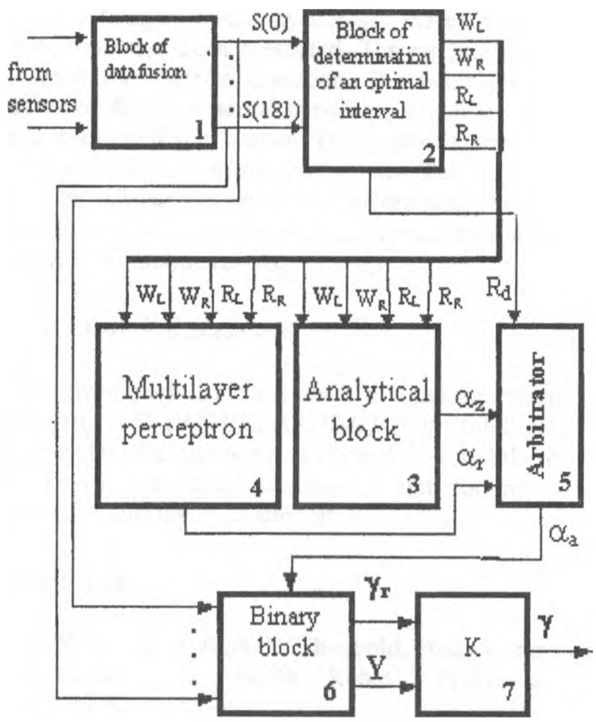

Fig. 1. Neural system for autonomous control of the mobile robot

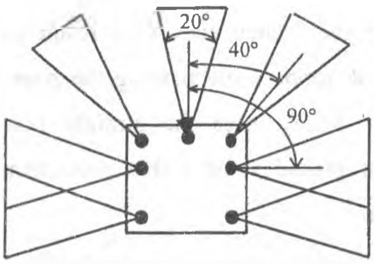

Figure 2. Ultrasonic sensors configuration

The block of data fusion is intended for integrating of different sensors and for the creation of the local environment map. This map is formed in the view radius of 2.4 meter and angular range of 180 grades:

OG = {S(i), -90<=i<=90},

where S(i) is the distance up to the obstacle if the angle between the current direction of the robot and the obstacle is equal i grades. The local environment map is considered to be the input information for the block of the

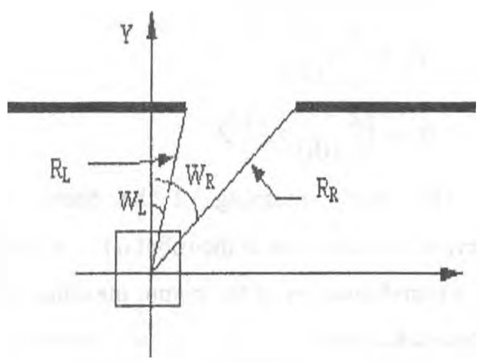

Figure 3.The linear and angular characteristics of the interval

determination of the optimal interval of motion.

Besides this block generates the compressed environment map S(p), $p = \overline{1,36}$, which is used for the control of the binary block.

The block of determination of the interval of motion is intended for the selection of the optimal interval of motion in the environment with obstacles. This interval is considered to be the nearest to the target. The output information of this block corresponds to linear (R_L, R_R) and angular (W_L, W_R) distances of the selected interval of driving (fig. 3).

In case when the free interval of driving is not chosen there is a turning of the robot on 90°, if it is possible and there is the search of the free interval.

Structurally the block 2 consists of 3 layers of neural elements which carry out different functions. This neural network is the dynamic neural network with fixed weights.

The analytical block is meant for the definition of the optimal direction α_z in the selected interval of driving. The optimal direction is characterized by such a direction of driving, which ensures minimal angular distance up to the target. The analytical block controls the motion of the robot on large intervals of driving, if R_d>2d, where R_d is the width of the chosen interval and d is the width of the robot. The architecture of this block consists of different neural layers and processor elements which carry out different functions. The input information of the analytical block is the angle between the current direction and the target, and also angular (W_L,W_R) and linear characteristics (R_L,R_R) of the chosen interval of motion.

The structure and the algorithm of functioning of these blocks was considered in [1-6].

The multilayer perceptron is intended for the orientation of the robot on narrow intervals of driving, where R_d < 2d. It forms the robust direction of motion α_r. The inexact environment map is of great importance for the orientation of the robot on the narrow intervals of motion. If one trains a multilayer perceptron to target output data in case of inexact input information it will provide the robust control of the robot. The input data of this block is the linear (RL,RR) and angular (WL,WR) distances to the obstacles.

The arbiter depending on the situation forms the current direction of the robot:

$$\alpha_a = \begin{cases} \alpha_z, if\ Rd > 2d \\ \alpha_r, otherwise \end{cases} \tag{1}$$

The binary block is intended for the control in the situation, when side distance up to the obstacle $\Delta \leq \Delta m$ is too small for the realization of sharp turns. This block transforms the input information to the binary array. The direction, which is formed by the binary block is not higher than 1°. It ensures the avoidance of the contact of the robot with the side obstacles. The commutator depending on the situation forms the final direction of driving of the robot:

$$\gamma = \begin{cases} \alpha_a, if\ Y = 0 \\ \gamma_r, otherwise \end{cases} \tag{2}$$

where Y=1, if $\Delta \leq \Delta m$.

Thus depending on the situation the robot can be controlled by the following units:

- Analytical block
- Multilayer perceptron
- Binary block together with the analytical block
- Binary block together with the multilayer perceptron

Such an approach provides stable driving of the robot in various situations. The neural system uses the system of close and long view. The velocity and the step of driving of the robot are normalized depending on the distance up to the obstacle. The break of the robot is performed, if distance up to the target is less than the defined value ε.

3. Multilayer perceptron

The multilayer perceptron is intended for the orientation of the robot on narrow intervals of driving. It provides stable control of the robot in case of inexact information from the environment map. This block forms the robust direction of driving α_r. The architecture of the given block consists of two multilayer networks MLP1 and MLP2 (Fig.4)

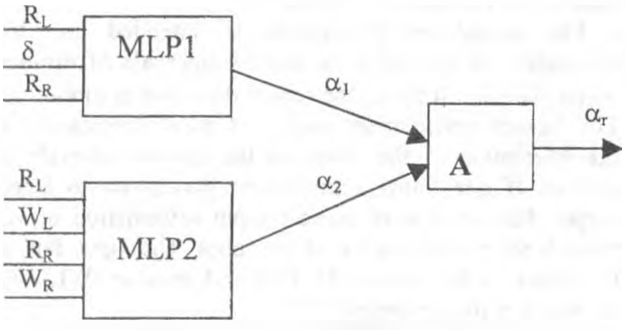

Figure 4. The architecture of the multilayer perceptron

The robust direction of the robot movement is formed by the arbiter:

$$\alpha_r = \begin{cases} \alpha_1, & if \ (R_L \vee R_R) \le g_1 R_t \\ \alpha_2, & if \ (R_L \wedge R_R) > g_1 R_t, \end{cases} \tag{3}$$

where $0 < g_1 \le 1$ – is the constant coefficient; R_t – is the critical threshold of visibility of the accepted system of the view.

According to the expression (3) $\alpha_r = \alpha_1$ if the robot moves in the space between the obstacles and $\alpha_r = \alpha_2$ if the robot moves in the tunnel.

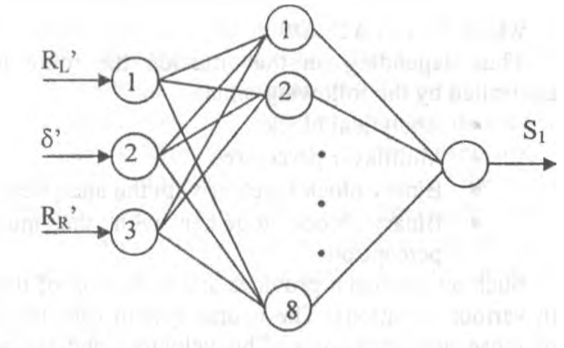

Fig.5 Structure of the block MLP1

Block MLP1 forms the arc of the circumference as the trajectory what secures the exclusiveness of the collision of the robot with the left or right side of the obstacle during the maneuvers. As a result the stable movement of the robot while passing the door ways is realized. Block MLP2 uses the straight line as the trajectory, what provides the stable movement of the robot in tunnels.

Let's examine the structure of these blocks..

Block MLP1 is the 3 layer neural network (fig.5)

It consist of 3 input, 8 hidden and 1 output units. Linear (R_L and R_R) and angular (δ) characteristics of the selected interval of motion are used as the input information. Here $\delta = W_L + W_R$.

Before entering the input of the neural network the data are scaled to the interval [0, 1] according to the following rules:

$$R_l' = R_L / 600, \tag{4}$$

$$R_R' = R_R / 600 \tag{5}$$

$$\delta' = \left(\delta / 100 + 1 \right) / 2 \tag{6}$$

The output meaning of the neural network characterizes the direction of the robot α1. It is formed by means of transformation of the output meaning S1 of the neural network:

$$\alpha_1 = \mathrm{int}(2S_1 - 1)100. \tag{7}$$

As a result the output meanings of the neural network may alter in the interval [- 100°, 100°]. As the function of activation the sigmoid function is used.

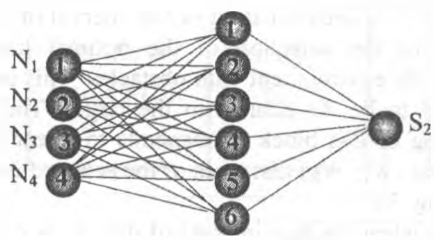

Fig. 6. Structure of the block MLP2

Block MLP2 is also 3 layer neural network (Fig.6). It consists of four input, six hidden and one output units.

Linear (R_L, R_R) and angular (W_L, W_R) distance of the chosen interval of motion are used as the input data. Before entering the input of the neural network the data are scaled as follows:

$$N_1 = R_L/600, \tag{8}$$

$N_2 = R_R/600$, (9)

$N_3 = (W_L/100+1)/2$, (10)

$N_4 = (W_R/100+1)/2$. (11)

The information on the output of the neural network characterizes a direction of driving of the robot α_2, which is defined as follows:

$$\alpha_2 = \mathrm{int}(2S_2-1)100, \quad (12)$$

where S_2 - the output value of the neural network.

4. The generation of the training set

It is necessary to generate training set for learning of neural networks MLP1 and MLP2. Each learning sample is presented in numeric form and consist of several input and one output meanings. Multilayer perceptron is used for the robot orientation on the narrow intervals of motion, the width of which is less than two meters. The radius of the robot view is 2.4 meters. Therefore it is necessary to generate training set in the following area V:

$$V \in \begin{cases} 1 < R_d \leq 2м \\ R_L \leq 2.4м \\ R_R \leq 2.4м. \end{cases}$$

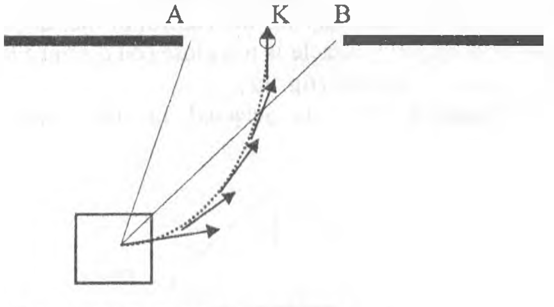

Figure 7 The trajectory of motion

Let's examine the formation of the training set for the block MLP1 .The block MLP1 forms the arc of the circumference as the trajectory of movement. It passes through the center of the robot and through the certain point K in the selected interval of motion (Fig.7) If the coordinates of the point K and the coordinates of the interval of the movement (X_A, Y_A, X_B, Y_B) are known it is possible to define the trajectory of the movement of the robot and the direction of the movement in each point(Fig.7).As a result a lot of learning samples for one position of the robot are formed.

If one performs the rotation of the selected interval of motion [A B] and the rotation of the point K around the center of the robot (Fig.8) it is possible to get different learning samples.

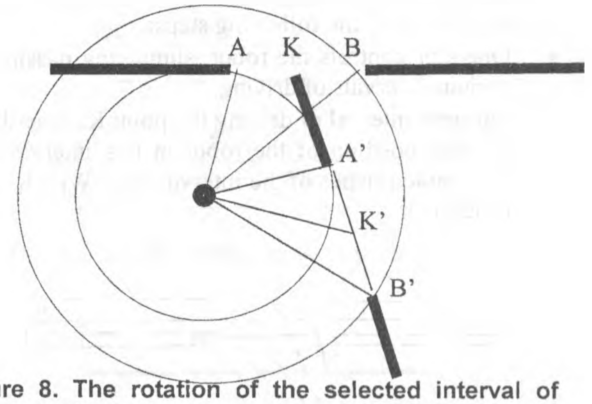

Figure 8. The rotation of the selected interval of motion [AB]

If the position of the robot relatively the interval of motion is changed in the area V and if the operations mentioned above are carried out it is possible to get training set, which consist of different patterns.

The formation of the training set for the block MLP2 is made in the same way. In this case it is necessary in a proper way to select the position of point K as a direction of the robot movement (Fig.9).

If the position of the robot is changed in the area V and if it is rotated with the selected step around the point O (Fig.8) a training set is formed.

In case of the inexact information the real position of the obstacle can differ from the environment map that sees the robot. Such situation is shown in fig 10, where the solid lines represent the position of the obstacle, which the robot sees and dashed - real position of an obstacle.

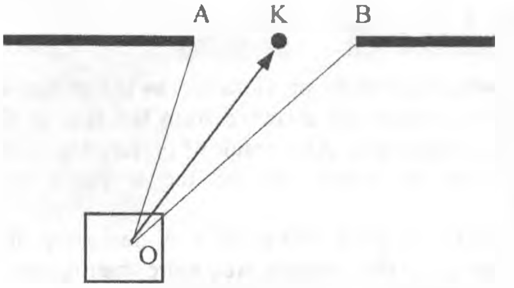

Figure 9. The direction of motion $\overline{OK}$

For the support of robust control of the robot in case of the inexact environment map it is necessary in appropriate way to select a position of point K in the selected interval of driving (fig. 10). If the neural network is trained to target output data, this can provide stable control of the robot in case of the inexact environment map.

The concept of training of the multilayer perceptron generally consists of the following steps:

- Operator controls the robot, simulating passing of various intervals of driving.
- For each interval of driving the point K, describing the real position of the robot in this interval and the characteristics of the interval (W_L, W_R, R_L, R_R) is defined.

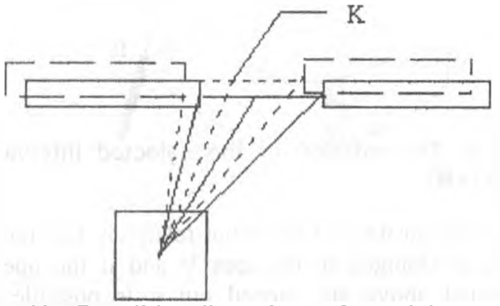

Fig. 10. Possible distortion of a visible interval of driving

- By means of rotation of input and output data in range of 180° learning patterns are formed.
- Training the neural networks MLP1 and MLP2 by means of back propagation algorithm is performed.

The given approach is characterized by the minimum set of the experimental data. It is enough to define only position of a point K and the characteristics of the interval of driving. The computer simulation of the multilayer perceptron was carried out. The size of learning set for the block MLP is 120 patterns. After training the robot successfully passed from various positions through narrow intervals.

5. Binary block

The disadvantage of the previous blocks is that they do not take into account the distance from the side of the robot up to the obstacle. As a result of performing of the maneuvers by the robot can be the collision with obstacles.

To maneuver without collisions it is necessary, that side distance up to the obstacle was more than radius of the circle, circumscribed around of the robot:

$$S > \frac{d}{2}\sqrt{2}, \qquad (13)$$

where d - the width of the robot.

If the condition (13) is not fulfilled, the control of the robot makes the binary block. In this case angle of turn of the robot in any direction is a constant and is equal to one grade.

The binary block is shown in fig. 11.

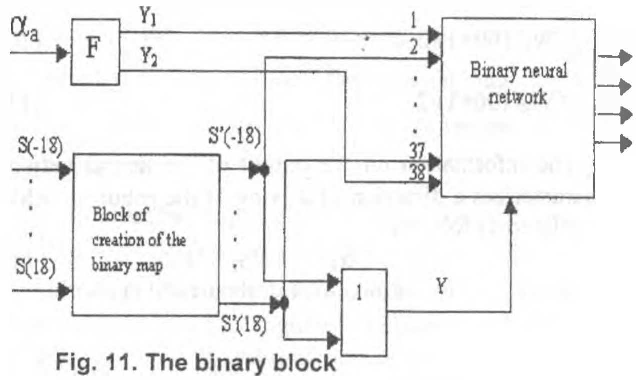

Fig. 11. The binary block

The block F is intended for conversion of the angular direction of driving α_a in binary array. It is necessary for the control of the binary neural network. The block F performs the following functions:

$$Y_1 = \begin{cases} 1, & if\ \alpha_a > 0 \\ 0, & otherwise \end{cases} \qquad (14)$$

$$Y_2 = \begin{cases} 1, & if\ \alpha_a < 0 \\ 0, & otherwise \end{cases} \qquad (15)$$

The block of creation of a binary environment map is intended for generating of the environment map of the given configuration (fig. 12) and formation of the signal Y of activation of the binary neural network.

Such a map is necessary for the control of the robot in situations, when the obstacle is too close (on distance less Δ)to the side of the robot (fig. 12).

The triangular form is selected on the basis of providing of smooth maneuvers with the presence of obstacles in front of the robot. As the input information the block of creation of a binary environment map uses the compressed environment map consisting of 36 units. The technology of conversion is, that if the obstacle is in zone ABCDE, the appropriate units S' (p) are installed in single values, otherwise in zero values (fig. 13)

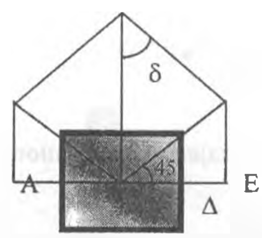

Fig. 12. Configuration of the environment map

As a result the binary array characterizes the presence of obstacles in the given area. This block consists of one layer of threshold neurons (fig. 14), each of which corresponds to the defined sector of the environment map.

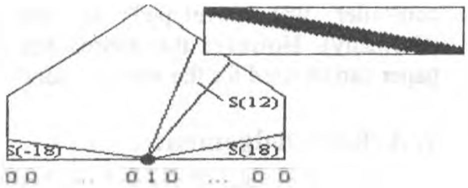

Fig. 13. Example of the binary environment map creation.

The neurons performs the following functions:

$$S'(p) = \begin{cases} 1, & if\ S(p) \le T(p) \\ 0, & otherwise \end{cases} \qquad (16)$$

Here T (p) - the threshold of the given neuron. For creation of the binary array in the given area it is necessary in appropriate way to form threshold values of neurons.

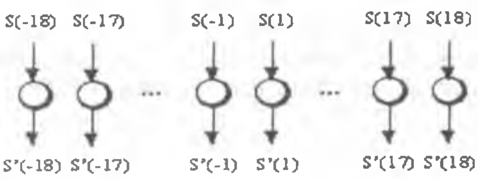

Fig.14. Creation of the binary environment map

The binary neural network is intended for the control of the robot, when the turns on the large values can evolve into collision with the obstacle. In this case Y=1 (fig. 11). Such a network represents the three-layer feed forward neural network (fig. 15).

The sigmoid function is used as the function of activation of units. The commands of the robot are formed by such a neural network(Fug.16).

Thus the turns are performed on 1°, that eliminates

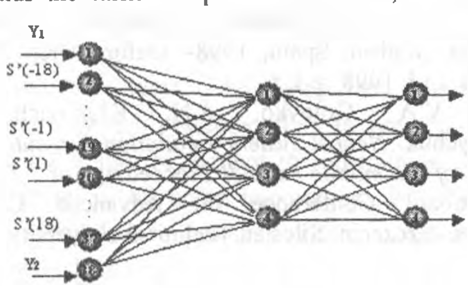

Fig. 15. Binary neural network

collision of the robot with the sides obstacles. For the control of the binary network also the signals Y_1 and Y_2 are used. So, if Y_1=1, that corresponds to α_a>0, the binary network will form the command of turn to the right on the value of 1°. Such an interaction of blocks 3,4 and 5 provides the driving of the robot in the nearest direction to the target. This is especially actual at the existence of the alternate paths of driving in narrow intervals (fig. 17).

The binary network works by the principle of overcoming of the obstacle. The possible variants of its operation are represented in fig. 18

In the second variant the turn takes place until the obstacle is not overcome.

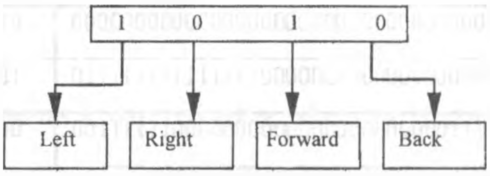

Fig. 16. Control commands of the robot

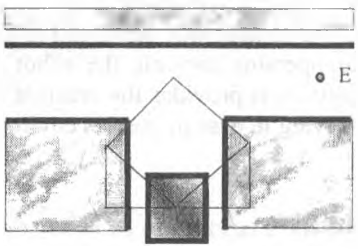

Fig. 17. Driving of the robot in case of the control from the binary block: E - target

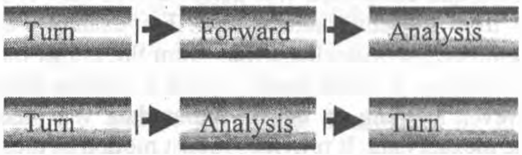

Fig. 18 Possible variants of the binary network functioning

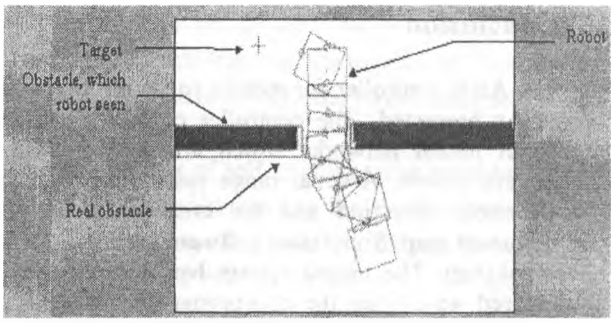

Fig. 19. Robot movement to the target with inexact information from sensor devices

For training of the binary network it is necessary to generate training sets. The generation of learning sampling is characterized by the simplicity and is performed by the logical way. The training sets for some situations are shown in table 1.

For training of the binary network the back propagation algorithm was used. The size of training set is 40 patterns. The regime of modeling can be used

Table 1

Input pattern	Output pattern
01100000000000000000000000000000000001	1010
10000000000000000000000000000000000000	0110
00000000000000000000000111111111111110	1000
00111111000000000000000000000011111100	0010
00000000000000000000000000000000000000	0010
01111111111111111111111111111111111110	0001

together with the logical way for creation of the training set when the operator controls the robot and simulate different situations. It provides the creation of the correct direction of driving in case of inexact environment map.

6. Experiments

For testing of the neural system the software has been developed which allows to simulate the robot motion The training and simulation was performed in case of inexact data from the environment map. For example, the real location of the obstacles differs from the things the robot sees. The tests were carried out for various situations. However learning to target output data will neutralize these inexact data. It provides robust motion of the robot.

Figure 19 shows the experiment where real obstacle differs from the things the robot sees. The tests have shown a good conformity to the theoretical results.

7. Conclusion

The ANN controller for mobile robot reactive control has been presented. The controller consists of different types of neural networks which are combined in the intelligent system. A sonar range system has been used for obstacle detection and for creation of the local environment map. Simulation software aspects have been accomplished. The neural system has been trained in a supervised way, using the backpropagation algorithm. A set of training patterns has been obtained, placing the robot heuristically in different situations. Such a controller was developed for the robot "Walter" (Germany). However the approaches described in this paper can be used for the various mobile robots.

7. Acknowledgement

This work is performed within the project INTAS 97-2028 " Intelligent neural system for autonomous control of a mobile robot". The authors are thankful to the European Community for the financial support of the project.

References

[1] V. Golovko, V. Dimakov and K. Schiling. *Intellectual system for control of mobile robot.* New trends in Artifical Intelegence and Neural Networks: Proc. Int. Conf. (Ankara, Turkey, may, 1997).- Ankara: EMO Scientific Books, 1997, 5p.

[2] V. Golovko and V. Dimakov. *Neural Control System For Mobile Robot.* / Preprints of Intern. Workshop on Intelegent Control INCON'97: (Sofia, Bulgaria, October 13-15, 1997). -Sofia: Technical University, 1997, 5p.

[3] V. Golovko and V. Dimakov. *Intellectual Simulation of Mobile Robot Control System //* Proceedings of the High Performance Computing Symposium, Boston, USA, 1998 – San Diego: The Society for Computer Simulation International, 1998, pp 440-445.

[4] V. Golovko and V. Dimakov. *Intelligent Neural System for Vehicle Control //* Proceedings of the High Performance Computing Symposium, Boston, USA, 1998 – San Diego: The Society for Computer Simulation International, 1998, pp 110 -115.

[5] V. Golovko and V. Dimakov. *Architecture of Neural System for Control of Autonomous Vehicles //* Preprints of the 3rd IFAC Symposium of Intelegent Autonomous Vehicles, Madrid, Spain, 1998– Oxford UK: Elsevier Science Ltd, 1998, v. l, p

[6] V.A. Golovko, A.N. Klimovich, D.Y. Nikolaychuk. *Neural system simulation for autonomous control of the mobile robot.//* Proceedings of The Fifth International Conference on Advanced Computer Systems,–Szczecin: Silesian Technical University.–1998.