THE NEURAL NETWORK APPROACH FOR REACTIVE CONTROL OF A MOBILE ROBOT

Vladimir Golovko*, Pedro Albertos, Klaus Schilling***,**
Valentin Dimakov*, Oleg Ignatiuk*

**Department of Computers and Mechanics,*
Brest polytechnic institute, Moscowskaja 267, 224017 Brest, Republic of Belarus,
Phone(+375162)421081, Fax:(+375162)422127, E-mail:cm@brpi.belpak.brest.by

***Dep. Ing. de Sistemas y Automatica,*
Universidad Politecnica de Valencia, P.O.Box: 22012. E-46071. Spain,
Tel.: 34-96-3879570 (9570),Fax.: 34-96-3879579,E-mail: pedro@aii.upv.es

****Steinbeis Transferzentrum ARS,*
FH Ravensburg-Weingarten,Postfach 1261, 88241 Weingarten, Germany
Fax: +49-751- 48523, E-mail: schi@ars-sun1.ars.fh-weingarten.de

Abstract: This paper describes the neural system for reactive control of a mobile robot. The neural system consists of different types of neural networks combined in the intelligent system. The efficient techniques for the training of neural networks are considered. The general 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.

Keywords: mobile robots, autonomous control, robot navigation, real-time systems, neural networks.

1. INTRODUCTION

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

This paper describes the intelligent neural system for reactive control of the mobile robot. The coordinates of final goal position and sensor data define input information for the system. The ultrasonic transducers and infrared scanner are 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 processes such map and generates motion direction and velocity. The neural system solves the following tasks:

- Sensor data fusion
- Building of the local environment map
- Obstacle detection and definition of a free interval of motion
- Definition of optimal direction in the chosen interval of motion

The general feature of this system is *interval-based model* (Golovko, *et al.*, 1997; Golovko, and Dimakov, 1998) of workspace round the robot instead of traditional *occupancy grid*. The idea here is no sense to pay attention to obstacles and their shape if we try to achieve a target. We pay attention only to intervals allowing going between obstacles, so called *free-intervals of motion*. This approach gives more compact representation of workspace and more nature behavior algorithm of the system.

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

2. ARCHITECTURE OF THE NEURAL SYSTEM

The general architecture of the neural system is shown in fig. 1. The system consists of various types of neural networks. Sensors location is shown in fig. 2.

The block of data fusion is intended to integrate different sensors and to build local environment map (LEM). It is formed in the view radius of 2.4 meter and angular range of 180 degrees:

$$LEM = \{S(i), -90° \le i \le 90°\}$$

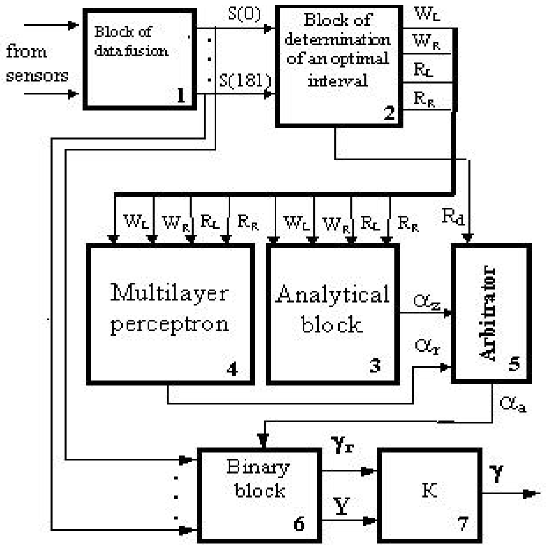

Fig. 1. Neural system for autonomous control of the mobile robot

where *S(i)* is a distance up to an obstacle if the angle between the current direction of the robot and the obstacle is equal to *i* degrees. LEM is considered as input information for the block of the 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 selected interval of motion (fig. 3).

In case if no free interval of motion is chosen, the turn of the robot on 90° takes place, if it is possible and the system searches new free interval.

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

The analytical block defines an optimal direction a_z in the selected interval of motion.

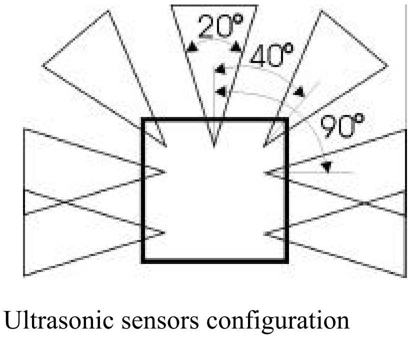

Fig. 2. Ultrasonic sensors configuration

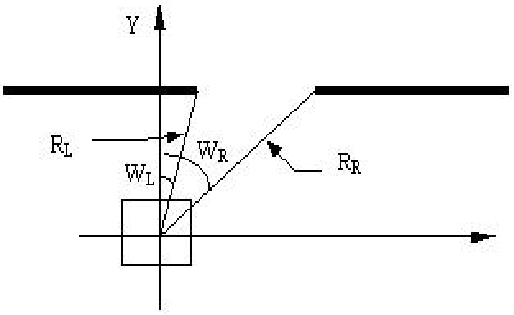

Fig. 3. The linear and angular characteristics of the interval

The optimal direction is characterized by the motion direction ensuring minimal angular distance up to the target. The analytical block controls the robot motion at wide motion intervals, if $R_d>2d$, where R_d is a width of the chosen interval and *d* is a robot width. The architecture of this block consists of different neural layers and processor elements performing different functions. The input information of the analytical block is an 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 were considered in (Golovko, e*t al.*, 1997; Golovko, and Dimakov, 1997, 1998a, b, c; Golovko, e*t al.*, 1998).

The multilayer perceptron (MLP) is intended to orient the robot on narrow intervals of motion ($R_d<2d$). It forms the robust direction of motion a_r. at inexact environment map. In this case to provide robust control, MLP is trained with inexact input information and desired output data. Here the input information is the linear (R_L,R_R) and angular (W_L,W_R) distances up to obstacles.

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

$$a_a = \begin{cases} a_z, if\ Rd > 2d \\ a\ \ ,otherwise \end{cases} \qquad (1)$$

The binary block is intended to ensure collision-free motion at sharp turns in case, if the robot has exceeded safe distance up to obstacle (*D£Dm*). This block transforms the input information to the binary array. The direction formed by the binary block is not large 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 the robot motion:

$$g = \begin{cases} a_a, if\ Y = 0 \\ g_r, otherwise \end{cases} \qquad (2)$$

where $Y=1$, if $D£Dm$.

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

- Analytical block
- MLP
- Binary block together with the analytical block
- Binary block together with the multiplayer perceptron

Such approach provides stable motion of the robot in various situations. The neural system uses the system of close and long view. The velocity and the step of the robot motion are normalized depending on a distance up to an obstacle.

3. MULTILAYER PERCEPTRON

The architecture of the given block consists of two multilayer networks MLP1 and MLP2 (Fig. 4)

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

$$a_r = \begin{cases} a_1, & if \;\; (R_L \vee R_R) \leq g_1 R_t \\ a_2, & if \;\; (R_L \wedge R_R) > g_1 R_t, \end{cases} \tag{3}$$

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

According to the equation (3) $a_r = a_1$ if the robot moves in the space between the obstacles and $a_r = a_2$ if the robot moves in the tunnel.

Block MLP1 forms the arc of the circumference as the trajectory for collision-free motion between obstacles, e.g. doorway passing. Block MLP2 uses the straight line as the trajectory providing the stable motion in tunnels.

Let's examine the structure of these blocks. MLP1 is the 3 layer neural network (fig. 5)

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

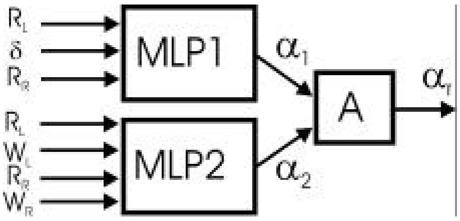

Fig. 4. The architecture of MLP

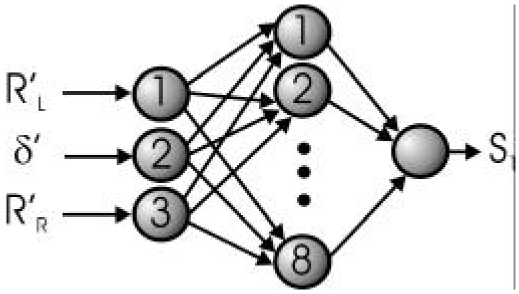

Fig. 5. The structure of the block MLP1.

Before entering the input information is scaled to the interval [0,1] according to the following rules:

$$R_L^{'} = {}^{R_L}\!\!\diagup_{600} \tag{4}$$

$$R_R^{'} = {}^{R_R}\!\!\diagup_{600} \tag{5}$$

$$d^{'} = \left({}^{d}\!\diagup_{100} + 1\right)\Big/2 \tag{6}$$

The output value of the neural network characterizes the robot direction a_1. As a result the output value S_1 is scaled in a range [-100°, 100°]:

$$a_1 = \left[(2 * S_1 - 1) * 100\right] \tag{7}$$

As the function of activation the sigmoid function is used.

MLP2 is 3-layer neural network as well (Fig.6). It consists of 4 input, 6 hidden and 1 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 information is scaled as follows:

$$N_1 = {}^{R_L}\!\!\diagup_{600} \tag{8}$$

$$N_2 = {}^{R_R}\!\!\diagup_{600} \tag{9}$$

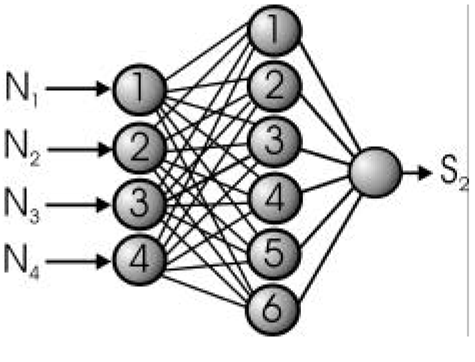

Fig. 6. Structure of the block MLP2

$$N_3 = \left(W_L/_{100} + 1 \right)/2 \qquad (10)$$

$$N_4 = \left(W_R/_{100} + 1 \right)/2 \qquad (11)$$

The output information of MLP2 characterizes the motion direction a_2 defined as follows:

$$a_2 = [(2 * S_2 - 1) * 100] \qquad (12)$$

where S_2 - the output value of MLP2.

4. THE TRAINING SET GENERATION

To generate training set for MLP1 and MLP2 each learning sample is presented in numeric form and consist of several input and one output value. MLPs are used for the robot motion in narrow intervals with width being less than two meters. The radius of the robot view is 2.4 meters. Therefore the training set is defined by the following area V:

$$V \in \begin{cases} 1 < R_d \leq 2m \\ R_L \leq 2.4m \\ R_R \leq 2.4m \end{cases} \qquad (13)$$

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 motion interval (X_A, Y_A, X_B, Y_B) are known it is possible to define the motion trajectory and the direction 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.

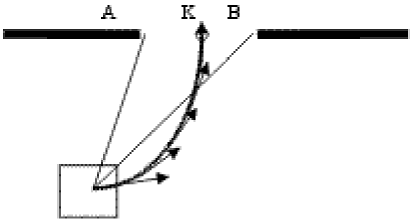

Fig. 7. The trajectory of motion

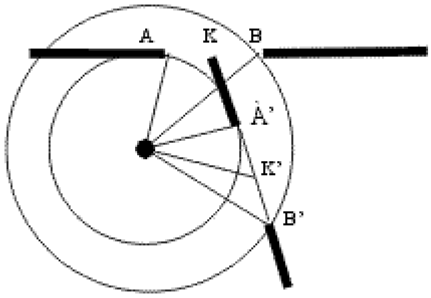

Fig. 8. The rotation of the selected interval of motion [A,B]

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

The formation of the training set for the block MLP2 is performed 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 motion (Fig. 9). If the robot position 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 position in the workspace seen by the robot. Such situation is shown in fig. 10, where the solid lines represent the position of the obstacle seen by the robot and dashed - real obstacle position.

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 fee robot in case of the inexact environment map.

The training concept of MLP generally consists of the following steps:

- Operator controls the robot by simulation of various motion intervals passing.
- For each interval of motion the point K describing the real robot position in this interval and the characteristics of the interval (W_L, W_R, R_L, R_R) are determined.

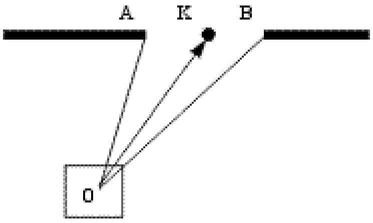

Fig. 9. The direction of motion $\overline{OK}$

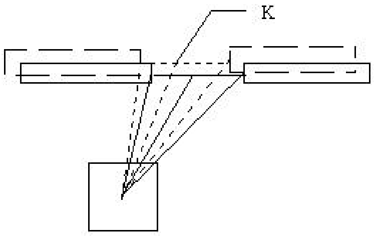

Fig. 10. Possible distortion of a visible motion interval.

- By means of rotation of input and output data in a range of 180° the 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 the point *K* and the characteristics of the motion interval. The computer simulation of MLP was performed. The size of training set for the MLP consisted of 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 performance of the maneuvers by the robot the collision with obstacles can take place.

To ensure collision-free motion it is necessary, that side distance up to the obstacle is large than radius of the circle round the robot:

$$S > \frac{d}{2}\sqrt{2} \tag{14}$$

where *d* - the width of the robot.

If the condition (14) 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 constant and is equal to one degree.

The binary block is shown in fig. 11.

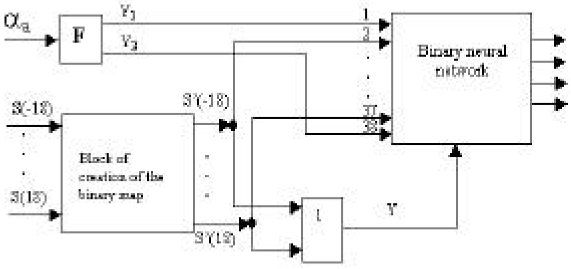

Fig. 11. The binary block

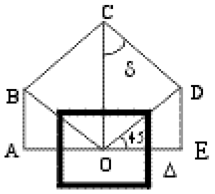

Fig. 12. Configuration of the environment map

The block *F* is intended for conversion of the angular direction of motion a_a, in binary array. The block *F* performs the following functions:

$$Y_1 = \begin{cases} 1, & \text{if } a_a > 0 \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

$$Y_2 = \begin{cases} 1, & \text{if } a_a < 0 \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

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 activation *Y* of the binary neural network.

Such map is intended for the robot control in situations, when the obstacle is too close to the robot side (fig. 12).

The triangular form has been chosen to provide smooth maneuvers with the presence of obstacles in front of the robot. As the input information the block of creation of the binary environment map uses the compressed environment map consisting of 36 units. The technology of conversion is, that if the obstacle is located in zone *ABCDE*, the values of appropriate units *S'(p)* are set in one, otherwise in zero (fig. 13)

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.

The neurons perform the following functions:

$$S'(p) = \begin{cases} 1, & \text{if } S(p) \le T(p) \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

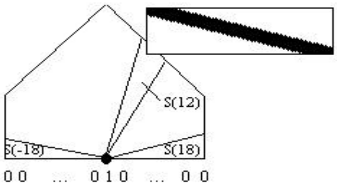

Fig. 13. Example of the binary environment map creation.

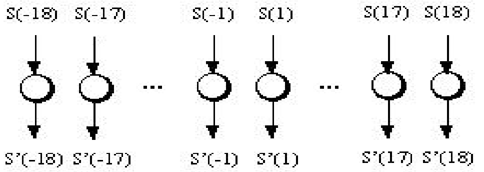

Fig. 14. Creation of the binary environment map

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.

To avoid collision-free sharp turn the binary neural network is used. It forms resulting solution for the binary block (fig. 11).

6. EXPERIMENTS

This ANN controller was developed for robot “WALTER” (Germany). The navigation system has been also adapted to robot “Pioneer 1” equipped only ultrasonic transducers. The testing has been shown the problems with perception ability of ultrasonic transducers, especially with detection of flat walls at the certain angle of beam reflection. They have been overcome by means of memorizing sensor data during motion with further filtering of random values. In some cases the obstacles were detected enough late to perform corresponding maneuver. It was connected with the feature of sensor transducer arrangement of the robot. In this case the robot came back on safe distance from an obstacle and taking into account new information it continued its motion. Fig. 15 shows example of door passing maneuver, where dots determine collected sensor information. The tests have shown a good conformity to the theoretical results.

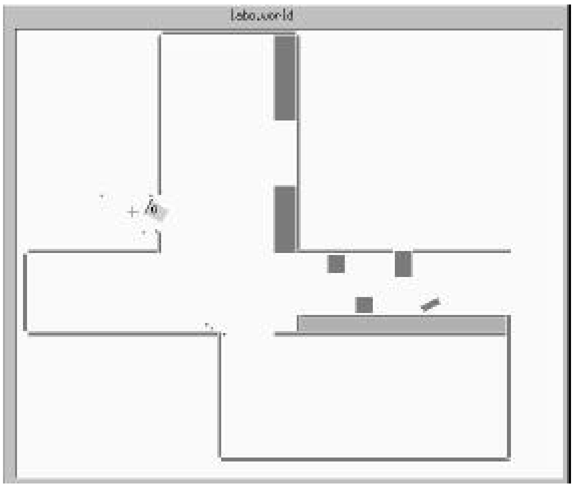

Fig. 15. Example of the robot “Pioneer 1” motion with inexact information of sensor devices.

7. CONCLUSION

The ANN controller for mobile robot reactive control has been presented. The controller consists of different types of neural networks 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 supervised way with use of the backpropagation algorithm. A set of training patterns has been obtained, placing the robot heuristically in different situations. The possibility of the ANN controller use has been shown on different types of robot.

ACKNOWLEDGEMENT

This work is performed within the project INTAS 97-2028 "Intelligent neural system for autonomous control of a mobile robot" and the project DLR BLR99/003. The authors are thankful to the European Community and the Deutches Zentrum fur Luft und Raumfahrt (DLR) for the financial support of the projects.

REFERENCES

Golovko V., V. Dimakov and K. Schiling (1997). Intellectual system for control of mobile robot. In: Proc. of Int. Conf. *New trends in Artificial Intelligence and Neural Networks.* EMO Scientific Books, Ankara, Turkey. 5p.

Golovko V., and V. Dimakov (1997). Neural Control System For Mobile Robot. In: *Preprints of Intern. Workshop on Intelligent Control INCON'97.* Technical University, Sofia, Bulgaria. 5p.

Golovko V., and V. Dimakov (1998a). Intellectual Simulation of Mobile Robot Control System. In: *Poceedings of the High Performance Computing Symposium, Boston, USA.* The Society for Computer Simulation International, San Diego. 440-445.

Golovko V., and V. Dimakov (1998b). Intelligent Neural System for Vehicle Control. In: *Poceedings of the High Performance Computing Symposium, Boston, USA.* The Society for Computer Simulation International, San Diego. 110 -115.

Golovko V., and V. Dimakov (1998c). Architecture of Neural System for Control of Autonomous Vehicles In: *Preprints of the 3rd IFAC Symposium of Intelligent Autonomous Vehicles, Madrid, Spain*. Elsevier Science Ltd, Oxford UK. v.1.

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