An Approach to Mobile Robot Self-training

Vladimir Golovko, Oleg Ignatiuk, Vladimir Sauta

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

Abstract

The unsupervised learning of the autonomous mobile robot is one of the actual research topics. It permits the artificial system to interact successfully with their environment and to avoid obstacles. This paper presents an intelligent control architecture which integrates self-training methods and available to operate in complex, unknown environment in order to achieve the target. Our approach is based on the reactive obstacle avoidance. The intelligent model integrates different neural networks and permits to perform on-line learning. The results of experiments are discussed.

Keywords Neural networks, self-organizing, mobile robot.

1 Introduction

Intelligent robot systems need the ability to train in order to adapt to changes in the environment. Such an interaction between the robot and environment is of great importance for artificial autonomous system. Life is full of situations, which are impossible to predict. In these cases the ability of a robot to self-training during the interaction with dynamically changing environment permits it to progress without a person (self-progressing).

There are basically two principal types of learning methods: supervised learning methods [1], in which the information about the desired outputs is contained, and reinforcement learning methods [1,2,3], in which reinforcement signal is presented. However in many control tasks it is difficult to obtain training data set. Therefore supervised learning is rarely used in comparison with the reinforcement learning.

In this paper we propose the approach, which permits the robot itself to collect the training samples through real-world experimentation. Such a learning is performed through trial-and-error. The backpropagation networks are used basically as neural networks, which are combined in intelligent system. However there is in general no supervisor. The robot starts with little a priori knowledge. In process of interaction with the unknown environment the robot has to find the optimal behavior and to accumulate the knowledge. As a result the robot can be self-trained while exploring its environment. Such an approach permits to operate in the absence of the model of the dynamics of the system. The proposed method was examined through simulation experiments.

2. General principles of obstacle avoidance.

Let's examine the main principles of building a reactive system for control of mobile robot. In this case only the current and target robot locations are known. The robot has to act in a priori unknown environment and to find corresponding action as the function of the current situation. As it follows from this, that the knowledge of the reactive system is represented as the situation-action. Such a system is the modular network that receives the situation and computes the actions.

The action corresponds to the current direction of the motion. We will examine the collision avoidance problem as the task of the robot motion in an indoor environment without colliding with any obstacles in the absence of any prior knowledge about obstacle positions. The robot has to find the shortest path between the current and the goal position.

2.1. Input information

The input information of the reactive system is the data from different sensors. These data are processed by means of data fusion. As a result the local environment map is generated. This map is formed in the certain view radius and the 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 heading direction of the robot and the obstacle is equal to *i* grades.

Such an approach we used for the mobile robot "Walter" [4]. This robot has 7 sonar sensors and the infrared scanner. These sensors provide distances to the obstacles in the view radius of 2,4 meter and the angular range of 180° grades. The local environment map is built as a result of the simple processing of data from the sensors.

2.2. Determination of the suitable passage of motion

The first task of the reactive system is the computing of the suitable passage of motion. Such a passage is considered to be the nearest to the target. It is obtained as a result of analysis of the local environment map. Such a passage is characterized by the linear (R_L, R_R) and angular distances to the obstacle (Fig. 1).

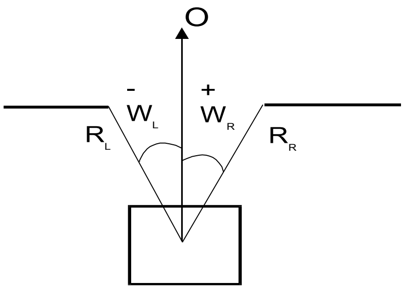


Figure 1: The linear and angular characteristics of the passage: O is the current heading direction

We use the dynamical network with fixed weights and the analytical approach for selecting the appropriate passage of motion.

In case when the free passage of the motion has not been chosen there is a turning of the robot on 90° if it is possible and there is the search of the suitable interval of motion.

The structure and the algorithm of functioning of such a module was considered in [5].

2.3. Definition of the optimal direction of motion

The second task of the reactive system is the definition of the suitable direction in the selected passage of motion. The optimal direction is such a direction of motion, which ensures minimal angular distance to the target in the chosen interval of movement. The definition of such a direction is performed by way of analysis of the selected passage. For this we use the analytical and neural network approaches.

Analytical approach

The analytical approach is used for control of the robot on large passage of motion, if $R_d>2d$, where R_d is the width of the chosen interval and d is the width of the robot. This approach is based on the analysis of the following data: angle α between the current and the target direction; angular (W_L, W_R) and linear (R_L, R_R) characteristics of the selected interval of motion. As a result we obtain the optimal direction of motion, which corresponds to the shortest path to the target. The analytical module has two operation modes. The first one corresponds to the robot motion if there is a free space or a tunnel without any obstacles. In this case the current direction of motion is a straight line. The angular and linear distances characterize the position of the door posts relative to the robot. The second mode corresponds to the robot motion in the space with obstacles. For instance, it characterizes the movement through door frames. In this case the robot has to cross the passage between two door posts perpendicularly. Then, the trajectory of motion must be the arc of the circumference. It passes through the certain point K in the selected interval of motion (Fig. 2). The position of the point K is defined by such a way to provide the shortest path to the target. The structure and the algorithm of functioning of such a module was considered in [5].

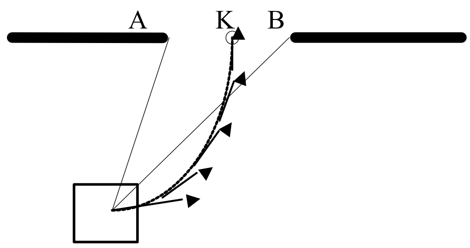


Figure 2: The trajectory of motion

Neural Network approach

Whenever the robot is moving obstacles have to be avoided. It is the complex problem, if the passage like door has a certain minimum size. In this case the inexact environment map can lead to the contact with an obstacle. The neural network approach is used for the robust control of the robot on the narrow passages of motion, if $R_d<2d$. If one trains a neural network to target output data in case of inexact input information it will provide the robust control of the robot. For these purposes we apply the backpropagation neural networks. The structure of the neural network module is shown in Fig. 3.

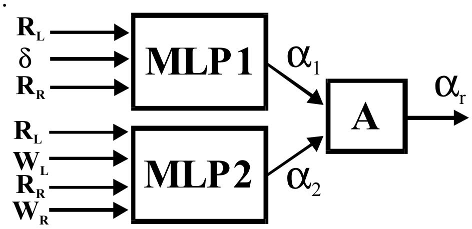


Figure 3: The architecture of the neural networks module

The architecture of the given module consists of two multilayer networks MLP_1 and MLP_2.

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

$$a_r = \begin{cases} a_1, \ \text{if} \ (R_L \vee R_R) \leq g_1 R_t \\ a_2, \ \text{if} \ (R_L \wedge R_R) > g_1 R \ , \end{cases} \quad (1)$$

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

According to the expression (1) $a_r = a_1$ if the robot moves in the space between various obstacles(through door posts, etc.), and $a_r = a_2$ if the robot moves in the tunnel. 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. It consist 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 $d=W_L+W_R$. The sigmoid function is used as the function of activation. Block MLP2 is also the 3-layer neural network. 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. We used the backpropagation algorithm with adaptive step for neural networks training [6].

3. The generation of the training set

It is necessary to generate training set for training of neural networks MLP1 and MLP2. Each learning sample is presented in numeric form and it consists 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 2d. Let R_t be the radius of the robot view. Therefore it is necessary to generate training set in the following area V:

$$V \in \begin{cases} R_d \le 2d \\ R_L \le R_t \\ R_R \le R_t. \end{cases} \qquad (2)$$

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. 2). 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. 2). 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 *[ÀÂ]* and the rotation of the point *Ê* around the center of the robot it is possible to get different learning samples. 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 the 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. 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* the training set is formed.

In case of the inexact information the real position of the obstacle can differ from the environment map that the robot sees.

For the support of robust control of the robot in case of the inexact environment map it is necessary in appropriate way to select the position of point K in the selected interval of driving. 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 given approach is characterized by the minimum set of the experimental data. It is enough to define only the position of point K and the characteristics of the interval of driving. The computer simulation of the neural networks module was carried out. The size of learning set for this module is 120 patterns. After training the robot successfully passed from various positions through narrow intervals.

4. Module of precision control

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

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

$$S > \frac{d}{2}\sqrt{2}, \qquad (3)$$

where d is the width of the robot.

If the condition (3) is not fulfilled, the control of the robot performs the module of precision control (fig. 4). In this case the angle of the turn of the robot in any direction is constant and is equal to one grade.

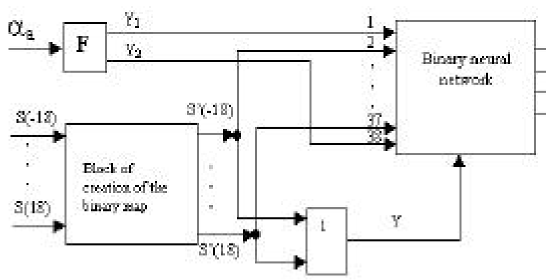


Fig. 4. Module of precision control.

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

$$Y_1 = \begin{cases} 1, & if\ a_a > 0 \\ 0, & otherwise \end{cases} ;\ Y_2 = \begin{cases} 1, & if\ a_a < 0 \\ 0, & otherwise \end{cases} \quad (4)$$

The block of creation of the binary environment map is intended for generating the environment map of the given configuration (fig. 5) and for the formation of the signal *Y* of the 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 (at the distance less than *D*) to the side of the robot (fig. 5).

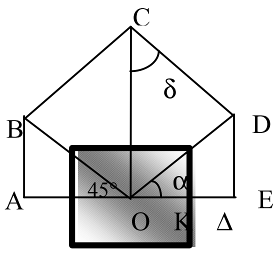


Figure 5: Configuration of the environment map

The triangular form is selected on the basis of providing smooth maneuvers with the presence of obstacles in front of the robot. The block of creation of a binary environment map uses the compressed environment map consisting of 36 units as the input information. 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. 6).

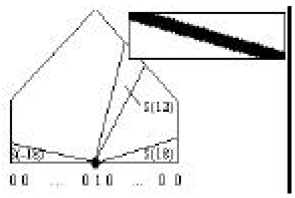


Figure 6: Example of the binary environment map creation

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. 7), each of which corresponds to the defined sector of the environment map.

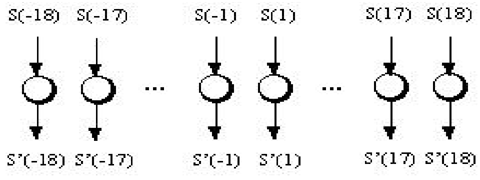


Fig. 7. Creation of the binary environment map

The neurons perform the following functions:

$$S'(p) = \begin{cases} 1, & if\ S(p) \le T(p) \\ 0, & otherwise \end{cases} \quad (5)$$

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

Nine neurons (*S'(-18),S'(-17)...S'(-10)*) correspond to the area *OAB* and nine neurons (*S'(10),S(11)...S'(18)*) correspond to the area *ODE*. These neurons take part in the formation of the signal of excitement *Y* of the binary neural networks (fig. 4).

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. 4). Such a network represents the three-layer feed forward neural network.

The sigmoid function is used as the function of activation of units. The commands of the robot are formed by such a neural network (Fig. 8).

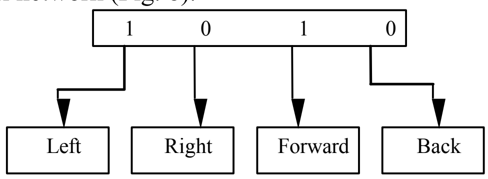


Fig. 8. Control commands of the robot

Thus the turns are performed on 1°, that eliminates collision of the robot with the sides obstacles. Also the signals Y_1 and Y_2 are used for the control of the binary network. So, if $Y_1=1$, that corresponds to $a_{\grave{a}} > 0$, the binary network will form the command of turn to the right on the value of 1°. Here $a_{\grave{a}}$ is the current direction of the robot, which is defined by analytical or neural network modules. Such an interaction between these modules provides the driving of the robot in the correct direction to the target. This is especially actual at the existence of the alternate paths of driving in narrow intervals.

The binary network operates on the principle of overcoming the obstacle.

For training 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.

For training the binary network the back propagation algorithm was used. The size of training set is 40 patterns. The regime of modeling can be used together with the logical way for creation of the training set. It provides the creation of the correct direction of driving in case of inexact environment map.

5. Self-training

As pointed out before the robot has to learn in the absence of a priori knowledge, which is determined by logical way, as it was shown in the previous sections. The initial knowledge of the robot can be filled up and corrected through real-world experimentation. Then the task is to train neural networks (section 2.3.2) for providing the robust control on the narrow gap of motion in the process of robot functioning. For this the robot must collect itself the training data set. Such a process of self-organizing take place through trial and error on the narrow passages of motion. The robot simply tries to find different actions for every situation and to collect the training samples. We assume, that the control of the robot in regime of self-training is performed by means analytical module and module of precision control.

The usage of the module of precision control in the regime of self-training gives an opportunity to decrease the number of mistakes while performing the maneuvers and consequently to accelerate the process of self-training. By this self-training can take place both for obtaining new knowledge and for correction of the old knowledge. As a result the adaptation of the robot to the environment is provided.

The process of self-training takes place by means of tries and mistakes on the narrow intervals of motion. If the maneuver is carried out successfully the training data for the learning of the multilayer perceptron are formed. If the try is not a success there is a return of the robot to the initial point for several steps back and the repetition of the maneuver (Fig. 9).

It is performed by means of the reconstruction of the situation on the previous step of the robot (t-1) and the formation of the correcting direction of the movement *?(k₁)*:

$$?(k_1)=?(k)\pm d, \qquad (6)$$

where *?(k)* is the direction of the motion, formed by the analytical block in the given point during the previous try of the maneuver; δ is the angel of the correction of the motion direction.

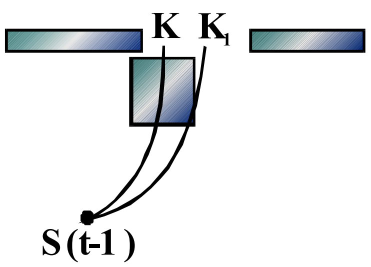
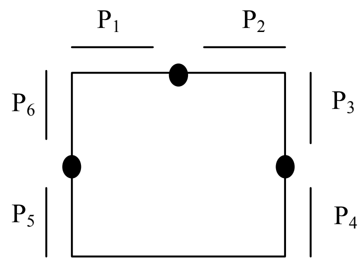


Figure 9: An example of an incorrect trajectory of the robot (point K): K_1 – corrected direction of motion

The robot has 6 tactile sensors evenly placed around its perimeter (Fig. 10). Tactile sensors are built into the bumpers and are binary; that is, they turn on when the robot has contact with an obstacle.

Figure 10: Tactile sensors disposition



By this the input signal of the sensor i equals one *($P_i=1$)*, if there has been the contact of the corresponding sensor with the obstacle.

In the opposite case $P_i=0$. The correction of the direction of the robot motion is performed by means of logical analysis of the information of the tactile sensors and the previous direction of the motion:

$$(P_1=1\vee P_6=1)\rightarrow d\ ;$$

$$(P_2=1\vee P_3=1)\rightarrow -d\ ;$$

$$((P_4=1)\vee(P_5=1))\wedge(y=1)\rightarrow -d\ ;$$

$$((P_4=1)\vee(P_5=1))\wedge(y=0)\rightarrow d \qquad (7)$$

In the expressions given above the signal y is formed as follows:

$$y=\begin{cases}1, & if\ \ g(k)>0\\ 0, & otherwise\end{cases} \qquad (8)$$

Thus the positive or negative meaning of the angle of the correction of the motion direction is formed if there has been a collision with the obstacle. It means that the point K in the selected interval of movement has not been correctly chosen and it is necessary to define the coordinates of the point K_1 according to the new direction *?(k₁)*.

As the module of precision control operates under the control of the analytical module as a result there is also the correction of output data of the binary neural network. In some situations it is necessary to correct the output meanings of the binary neural network by means of changing of the variant of its functioning. It is carried out by means of the

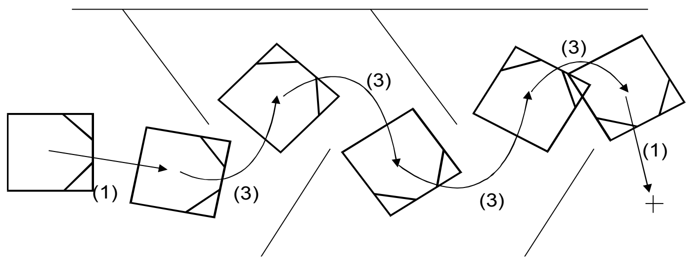


Figure 11.

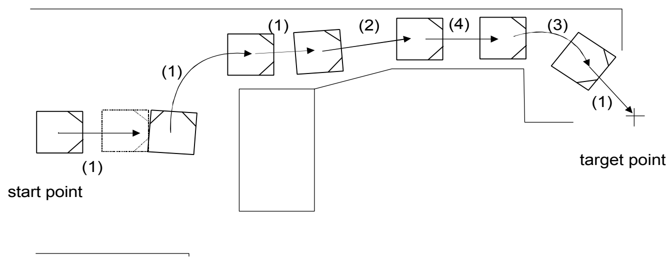


Figure 12.

(1)-Analytical module (2)-MLP_1 (3)-MPL_2 (4)- Precision control module analysis of the information of the tactile sensors and previous output meanings of the module of precision control. For example, if

$$(P_1 = 1) \wedge K(t-1) = 0110 \to K(t) = 0100\,;$$

$$(P_2 = 1) \wedge K(t-1) = 1010 \to K(t) = 1000\,,$$

where *K(t-1)* and *K(t)* – are accordingly the output meanings of the binary block of the previous and the present stage of functioning.

The maneuver is considered to be a success, if the robot reaches the point K in the selected interval of movement without collision with obstacles. By means of rotation are defined for blocks MLP_1 and MLP_2 the training patterns, on the basis on coordinates of point K and the characteristics of the interval of movement. As a result of the modeling of different situations the training set is formed.

6. Simulation results

The online learning capabilities of our approach were investigated in a typical real-world environment by simulation. In these experiments the weights of the neural networks (section 2.3.2) were initialized randomly and in process of interaction with the unknown environment the training of these networks was performed. For such a training the backpropagation algorithm with adaptive rate was used [6].

First of all the agent has to collect the training set by means trial and error and to employ the backpropagation for training. The inexact data from sensors was used for simulation. For instance the linear and angular distances to the obstacles was differed from real values. After self-training, the agent was tested in various situations. The slalom task is shown in Fig. 11. One can see, that the robot control is performed both by analytical module and MLP_2 module. Fig. 12 finally illustrates the behavior of the robot in environment with narrow passages. As can be seen the agent uses the different modules for control.

7. Conclusion

This paper presents our approaches to autonomous mobile robot navigation. By this during robot motion self-training and self-organization is performed, using the neural networks. The neural networks have been trained in an unsupervised way. During the interaction of the robot with the environment the training data are collected, which are used for training. Such an approach permits to adapt the robot to different situations. By this the described self-training system works in real time.

Acknowledgements

The given work is carried out within the framework of the project INTAS-BELARUS-97-2098 “Intelligent neural system for autonomous control of a mobile robot” and project BLR99/003 (Deutsches Zentrum fuer Luft und Raumfahrt). The authors thank European Union for the informational and financial support.

References

[1] S.Thrun. *An approach to learning mobile robot navigation.* // Robotics and Autonomous Systems. 1995, V.15, N.4, pp.301-319.

[2] Ron Sun. *Autonomous learning of sequential tasks: Experiments and analyzes.* // IEEE transactions on Neural networks. 1998, V.9, N.6, pp.1217-1233.

[3] Jose del R.Millan. *Reinforcement learning of goal-directed obstacle avoiding reaction strategies in an autonomous mobile robot.* // Robotics and Autonomous Systems. 1995, V.15, N.4, pp. 275-299.

[4] V.Golovko, K.Schilling, H.Roth, R.Sadykhov, P.Albertos, V.Dimakov. *The architecture of the neural system for control of a mobile robot.* // Proceedings of the ICNNAI’99, Brest, Belarus, 1999, pp. 57-61.

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

[6] V.Golovko, Y.Savitsky. *New approach of the recurrent neural network training.* // Proceedings of the ICNNAI’99, Brest, Belarus, 1999, pp.32-35.