

**CONTROLLING A SMALL MOBILE 3-PI ROBOT MOVEMENT  
IN A MAZE VIA THE NEURAL NETWORK USING  
BACK-PROPAGATION LEARNING METHOD**

Dušan HORVÁTH<sup>1</sup>, Zuzana ČERVENANSKÁ<sup>1</sup>

<sup>1</sup>SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA  
FACULTY OF MATERIALS SCIENCE AND TECHNOLOGY IN TRNAVA  
INSTITUTE OF APPLIED INFORMATICS, AUTOMATION AND MECHATRONICS  
ULICA JÁNA BOTTU Č. 2781/25, 917 24 TRNAVA  
e-mail: dusan.horvath@stuba.sk, zuzana.cervenanska@stuba.sk  
*Received 26 August 2021, Accepted 22 October 2021, Published 24 November 2021*

**Abstract**

*The contribution is focused on technical implementation of controlling a small mobile 3Pi robot in a maze along a predefined guide line where the control of the acquired direction of the robot's movement was provided by a neural network. The weights (memory) of the neuron were calculated using a feedforward neural network learning via the Back-propagation method. This article fastens on the paper by the title "Movement control of a small mobile 3-pi robot in a maze using artificial neural network", where Hebbian learning was used for a single-layer neural network. The reflectance infra-red sensors performed as input sensors. The result of this research is the evaluation based on the experiments that served to compare different training sets with the learning methods when moving a mobile robot in a maze.*

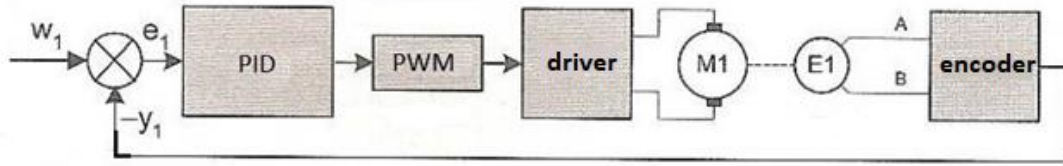
**Keywords**

*Mobile robot's control subsystem, navigation, neural network, maze solving robotic vehicle, back-propagation*

**INTRODUCTION**

Autonomous vehicles and mobile robot movement control keep attracting the attention for decades, namely owing to their key role in meeting navigation requirements in the conditions unsuitable or dangerous for humans. The robot control subsystem can be implemented by the software which evaluates the data from the input sensors based on various software conditions (if, switch), where the input data (input vector) from the robot sensors are assigned the required outputs (e.g. motors control). Also, the control can be ensured by applying the PID controllers, where the input data is compared to the anticipated value; the difference of those values

determines the size of the feedback control e.g. motors [1–3]. An example of applying the PID control can be seen in Figure 1.



**Fig. 1** A scheme of electric motor control using a PID controller

Fundamental navigation problems such as recognition or adaptation are analogous to cognitive tasks of human brain. An artificial neural network (NN) which represents a system of parallel distributed processing elements (neurons) connected in a graph topology can represent a way to deal with them [1,4]. After learning, the neural network can express the knowledge implicitly in the weights associated to the inputs of neuron [5]. Neuron memory in the terms of calculated weights depends on the data from the input sensors (input vector) and the anticipated outputs in the process of neurons learning. During robot's operation, the control subsystem calculates the robot's response to the input vectors, based on the weights and input vector.

Monitoring the robot position requires application of a suitable sensing method. A wide-ranging overview of sensors classification, their characteristics, and utilization of laser, sonar, and infrared sensors for detecting position was published e.g. in [2]. Reflectance infra-red sensors or a single-line video camera are used to detect the guide line. The method of creating a map of the environment is also used for the fastest possible movement along the guide line [6]. Motion subsystem sensors, i.e. encoders, are used to calculate the travelled distance. The sensors belonging to the robot motion subsystem are used to help navigate the robot. The most widespread method of navigation is odometry [1, 3]. Applying that method, the robot updates its position and records it on a map of the environment. With the environment map, the robotic car can go very quickly along the guide line. A brief description of some effective maze exploration algorithms is presented in [7]. The authors also provided details of specification for a maze solving robotic vehicle equipped with Arduino Uno platform. It possesses three ultrasonic sensors, two electro-mechanical encoders, and a motion tracking device with 3-axis gyroscope and 3-axis accelerometer for detection of position in the maze.

## MATERIALS AND METHODOLOGY OF EXPERIMENT

### Learning NN with Back-Propagation method

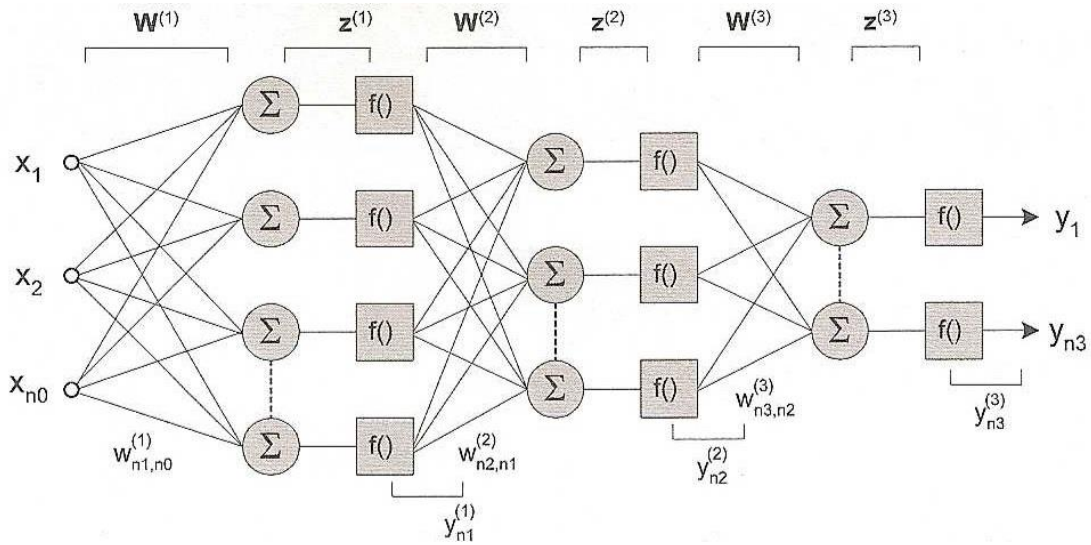
Generally, feedforward artificial neural networks (Multilayer perceptrons - MLPs) are used to solve linearly non-separable tasks [8]. The MLP comprises an input layer, hidden layers, and an output layer. The topology is depicted in Figure 2. The sigmoid-shaped function is most often used as the activation function in the form of (1)

$$y=1/(1+e^{-\lambda z}). \quad (1)$$

In feedforward neural networks, the response of the output layer of neurons to the input vector is determined by the forward propagation of the signal in a few steps.

First, the neurons of the input layer are excited. Then, these excitations are brought by means of bonds to the next layer and adjusted via synaptic weights. Finally, each neuron belonging to the higher layer performs the sum of signals from the lower layer neurons, and it is excited to the level given by its activation function.

The process takes place throughout all layers up to the output layer where excited states of all neurons of the output layer are finally obtained. Information flows through the network via fully connected layers.



**Fig. 2** Scheme of multilayer perceptron. Adapted from [1]

Training the applied NN is based on the neuron weights adaptation. We used the most well-known and effective algorithm for learning the MLP, that is the Back-propagation (BPG) method [8]. Back-propagation refers to the method for computing the loss function gradient. It utilizes supervised learning for updating weights to minimize loss function. For BPG, the loss function calculates the difference between the network output and its expected output, after a training example has propagated through the network. The actual calculation (adaptation) of weights and biases takes place until the learning error (difference between the expected and calculated value) is minimal for all input vectors  $X$ .

### **The implementation of a neural network for the movement of a 3-Pi robotic car control with the BPG learning and bipolar sigmoid activation function**

For the implementation of NN to control the movement of mobile robot, we used a 3-Pi two-wheeled robotic car produced by Pololu [9] (see Figures 3 and 4 for the top and bottom views) to test functionality of the designed neural network. Five reflectance infra-red sensors were used to detect the black guide line. Since the reflection of infrared light from a black surface is different from that of a white one, a different voltage occurs at the infrared sensors output. The voltages from the sensors were digitized, and they represented the input vectors in the table of the expected movement of the mobile robot. An 8-bit microcontroller ATMEGA328P by ATMEL [9] controlled the movement of the 3-Pi robotic car. All main components are shown from the bottom view in Figure 4.

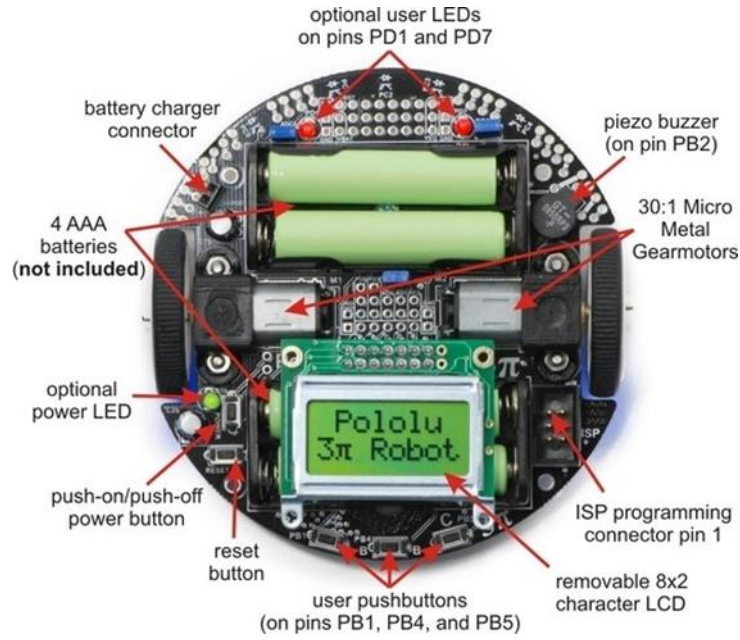
A simple neural network consisting of two input and two output neurons was used to control the 3-Pi robot in the maze, one neuron in the output layer for each motor. As can be seen in Figure 5, the neural network has five inputs. The inputs are generated by infra-red sensors for sensing the black guide line.

The calculated values of weights and biases were entered into the control subsystem of the robot. The values  $w_{1,1}$  to  $w_{1,5}$  (Figure 5) are the weights for individual inputs of the neuron controlling the left motor (LM). Similarly,  $w_{2,1}$  to  $w_{2,5}$  are the weights for the individual inputs of the right motor (RM) neuron. To ensure the desired robotic car movement in sense of motor

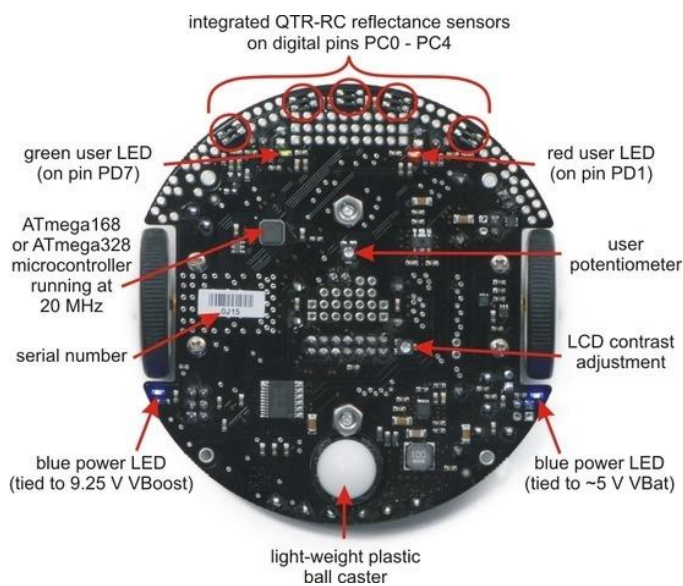
turning into both directions – forward and back, the applied NN was designed using the bipolar sigmoid (hyperbolic tangential [2]) activation function (2) with the values in the range (-1,1).

$$y = \frac{1 - \exp(-z)}{1 + \exp(-z)} . \quad (2)$$

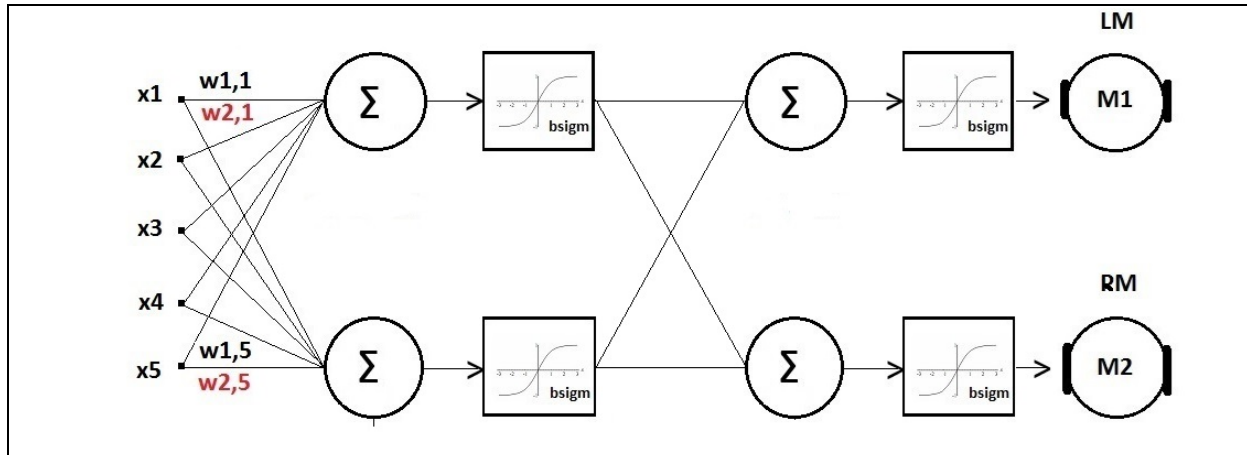
The value from the range (-1, 0) ensures the back direction of an engine rotation, and the value from the range (0,1) the forward direction. The maximum number of iterations in the learning process was limited to 1000000, and learning error  $\Delta = 0.001$ .



**Fig. 3** Top view of a 3-Pi robotic car. Adapted from [9]



**Fig. 4** Bottom view of a 3-Pi robotic car. Adapted from [9]

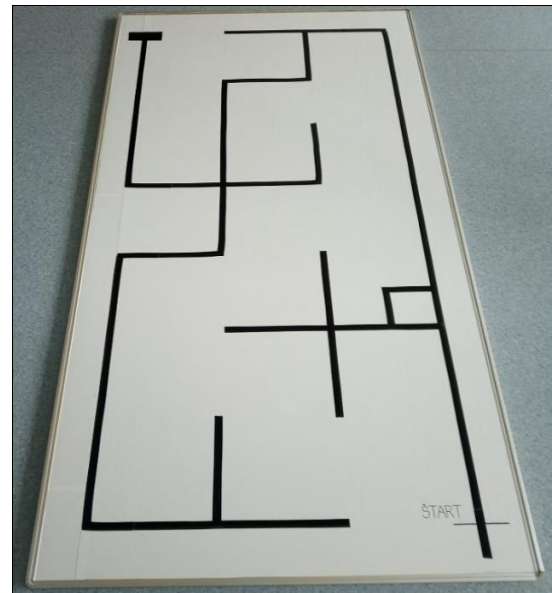


**Fig. 5** The scheme of the topology of the applied neural network

For NN, we used a training set identical to the training set described in article [10]. The training set constructed for a well-defined robotic car movement in the maze is shown in Table 1 and Figure 7. We can see the values of ten input vectors transmitted by five infrared sensors. The white area was evaluated as logical 0 by infrared sensors, the black area as logical 1, respectively, so all vectors had a binary representation.

The shape of the experimental maze is displayed in Figure 6. The beginning of the lane is marked with the inscription "START". The end of the maze is marked with a wide transverse black strip. The robot went through the whole maze until it found the end (exit) of the maze. The robot stopped on a wide transverse black tape.

<b>Table 1</b> The training set for 3-Pi robotic car motion in the maze			
<b>Order</b>	<b>Training set</b>	<b>Expected value for LM</b>	<b>Expected value for RM</b>
<b>1</b>	0 0 0 0 0	-1	1
<b>2</b>	0 0 1 0 0	1	1
<b>3</b>	0 1 0 0 0	0.1	1
<b>4</b>	0 0 0 1 0	1	0.1
<b>5</b>	1 1 1 0 0	-1	1
<b>6</b>	1 0 0 0 0	-1	1
<b>7</b>	1 1 1 1 1	-1	1
<b>8</b>	1 0 0 0 1	-1	1
<b>9</b>	0 0 1 1 1	1	-1
<b>10</b>	0 0 0 0 1	1	-1



**Fig. 6** The experimental maze. Adapted from [10]

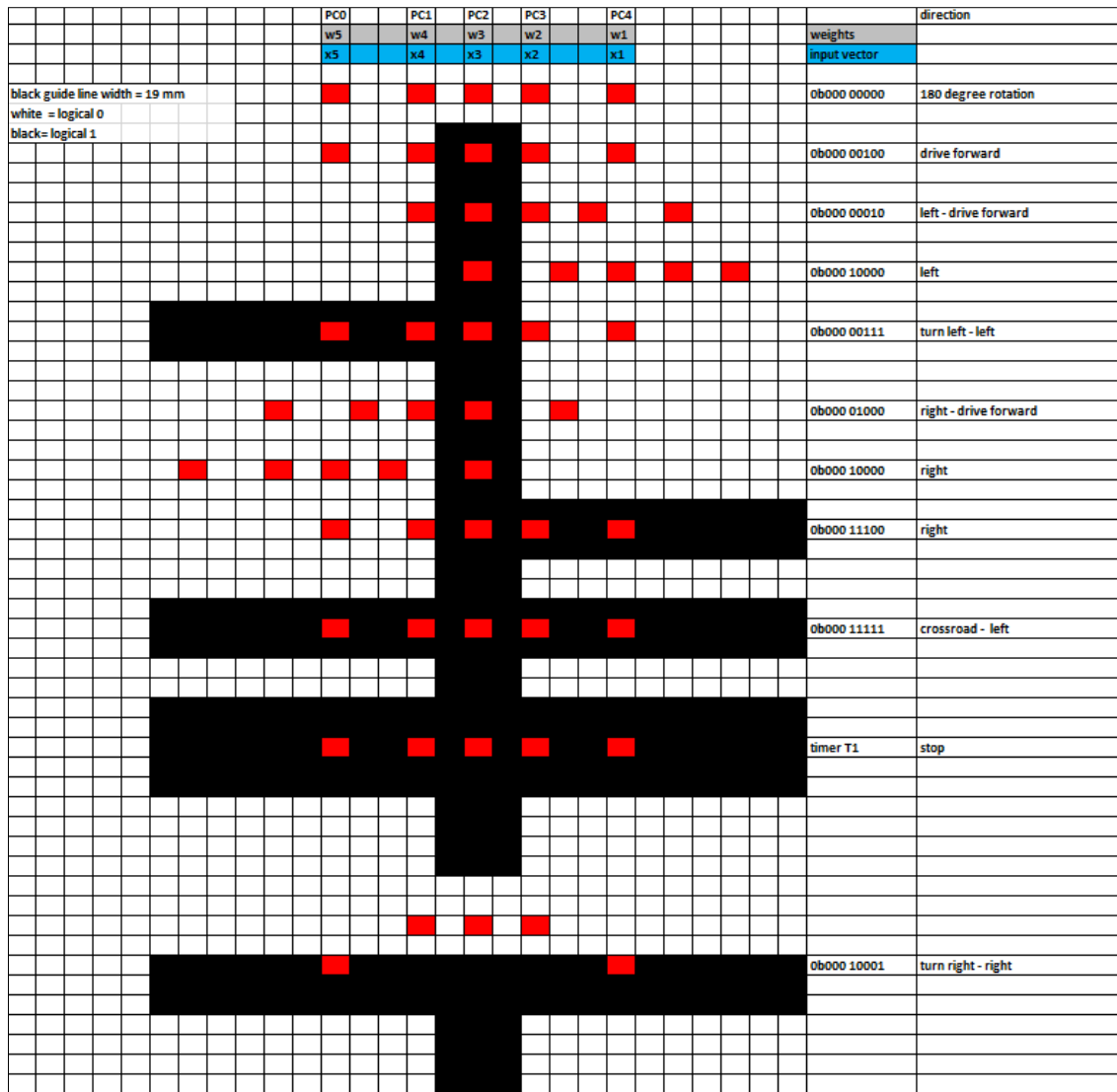


Fig. 7 Applied input vectors which determine the movement along the line. Adapted from [10]

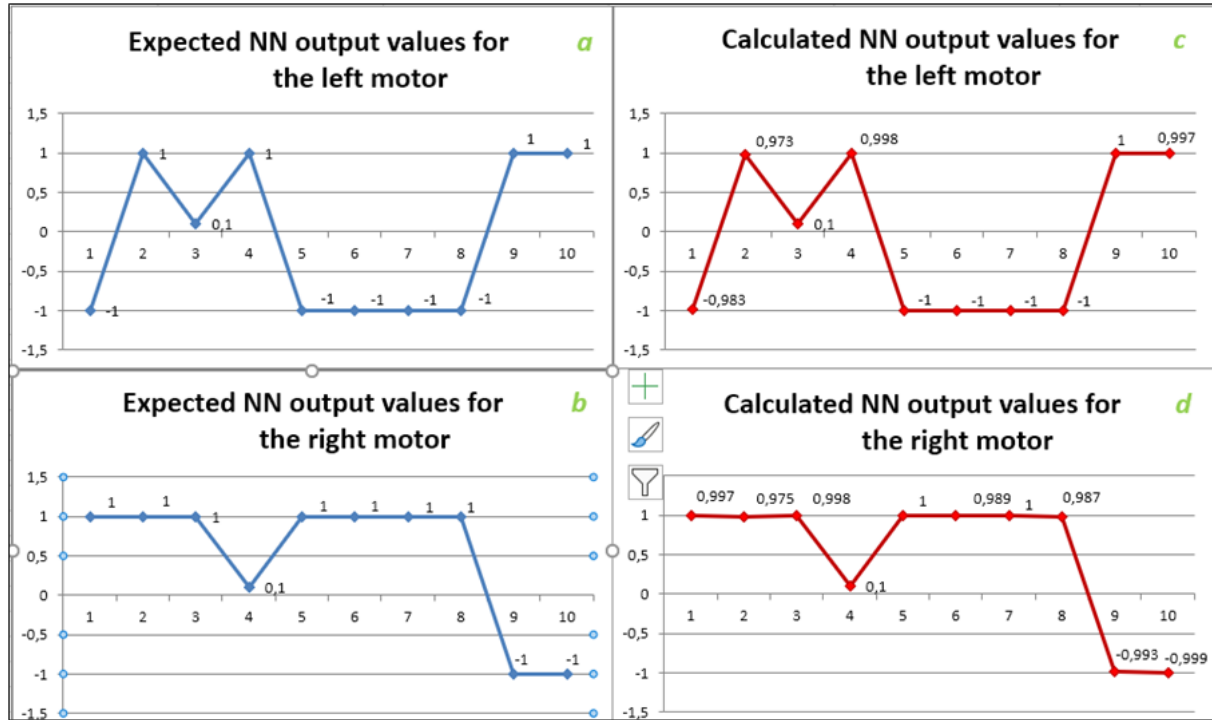
## RESULTS OF EXPERIMENTS

The results of the weights calculation (neurons memory) for individual neural network with bipolar sigmoid activation function are presented in Table 2.

Table 2 Weight coefficients of NN for 3-Pi robotic car control in the maze		
Neuron	Bias	Weights
1st neuron in input layer	0.81	$w_{1,1}=6.277$ ; $w_{1,2}=-0.480$ ; $w_{1,3}=-1.008$ $w_{1,4}=-1.537$ ; $w_{1,5}=-2.024$
2nd neuron in input layer	-0.89	$w_{2,1}=5.813$ ; $w_{2,2}=-2.034$ ; $w_{2,3}=-4.794$ $w_{2,4}=-1.247$ ; $w_{2,5}=1.083$ ;
1st neuron in output layer	-0.83	$wo_{1,1}=-14.298$ $wo_{1,2}=-3.759$
2nd neuron in output layer	-0.52	$wo_{2,1}=11.715$ $wo_{2,2}=-6.072$



Figure 8 displays the comparison of expected and predicted values acquired via applied NN according to the order of elements of training set (presented in Table 2). The blue line in Figure 8 shows the anticipated values, and the red line shows the calculated values for the individual motors calculated via the Back-propagation method with a learning error  $\Delta = 0.001$ . We can see that, owing to the small learning error, the calculated weight coefficients provide practically the same values of the individual neurons at the output as the anticipated ones.



**Fig. 8** Comparison of expected (a, b) and calculated (c, d) values for the left and right motor of the robotic car for each of ten elements of the used training set

## DISCUSSION

During the movement along the black line, the robotic car sensed the reflection of infrared light from the space below with its infra-red sensors. The response of neural network to input vector are the values which are directly going to the motion subsystem of the robotic car. The result of this action is the movement of the mobile robot along the path.

According to the network design process, this experiment fastens on the research described in article [10], yet it differs from the former one in application of a different type of the neural network learning (Back-propagation method) and the network topology. The designed NN contains no hidden layer, which is due to the limited computing power of the P328 microprocessor and the use of bipolar sigmoid as a computing power-intensive activation function. Practical experiments showed that, when using Hebb's learning, a training set can have eleven input vectors at maximum, otherwise the training set is "linearly non-separable", and thus the neural network cannot be trained.

Based on the performed experiments with the NN movement control of the smooth and precise movement of a 3-Pi robotic car, it is necessary to process the data from the input sensors quickly and as accurately as possible, with the smallest possible deviation of the calculated data from the expected data. The aim of the network design process was to achieve the smallest possible learning error when neurons were learning, so that the robot can follow the guide line as accurately as possible. Compared to [10], the result of the robot motion in the maze was the

same as that in the case when the control was based on the Hebb learning using the same training set.

## CONCLUSION

The presented neural network is suitable for the movement control of a 3-Pi robot in a maze or along a circular or oval paths. Compared to results of the experiments performed on a single-layer with Hebbian learning [10], the 3-Pi robotic car reliably follows the path regardless of using either Hebb's learning or the Back-propagation method with the same training set. The advantage of the back-propagation method is the possibility of using more input vectors than in the Hebbian case.

## References

- [1] NOVÁK, P. 2005. *Mobilní roboty (1 díl). Mobile robots (1st part)*. Ben- technical literature, Praha, 2005. 247 p. ISBN 80-7300-141-1.
- [2] TZAFESTAS, S. 2014. *Introduction to Mobile Robot Control (1st edition)*. Elsevier, 2014. 691 p. ISBN 978-0-12-417049-0.
- [3] SIEGWART R., NOURBAKHSH, I. R., SCARAMUZZA, D. 2011. *Introduction to Autonomous Mobile Robots* (2 nd ed.). Massachusetts Institute of Technology, 2011. 472 p. ISBN 978-0-262-01535-6.
- [4] GUL, F., RAHIMAN, W., ALHADY, S.S. I. N. 2019. A comprehensive study for robot navigation techniques. 2019, *Cogent Engineering*, **6**(1), 1-25. ISSN 2331-1916.
- [5] JANGLOVÁ, D. 2004. Neural Networks in Mobile Robot Motion. *International Journal of Advanced Robotic Systems*, 2004, **1**(1), pp. 15-22, ISSN 1729-8806.
- [6] Modelovanie a riadenie mobilného robota. (Mobile robot modeling and control) [Online]. [Accessed: 08-2021] Available at [http:// www.atpjournals.sk/buxus/docs/casopisy/atp\\_plus/plus\\_2006\\_2/plus63\\_67.pdf](http://www.atpjournals.sk/buxus/docs/casopisy/atp_plus/plus_2006_2/plus63_67.pdf).
- [7] BIENIAS, Ł., SZCZEPANSKI, K., DUCH, P. 2016. Maze exploration algorithm for small mobile platforms. *Image Processing & Communications*, **21**(3), 15-26, ISSN 1425-140X.
- [8] GOODFELLOW, I., BENGIO, Y., COURVILLE, A. 2016. *Deep Learning*. MIT Press, 2016. 800 p. ISBN 978-0262035613.
- [9] [www.pololu.com/docs/0J26/all](http://www.pololu.com/docs/0J26/all). Pololu 3pi Robot User's Guide. [Online]. [Accessed: 08-2021] Available at <https://www.pololu.com/docs/0J26/all>.
- [10] HORVÁTH, D., ČERVENÁNSKÁ, Z., ĎURIŠ, R. 2020. A movement control of a small mobile 3-Pi robot in a maze using artificial neural network. *International Journal of Mechatronics and Applied Mechanics*, 2020, **8**(1), 81-88. ISSN 2559-6497.

## ORCID

Dušan Horvách	0000-0002-9647-4627
Zuzana Červeňanská	0000-0002-6380-4634