

## Implementation of Kohonen Network in Behavioral Control of the Amigobot Wheeled Mobile Robot

Andrzej BURGHARDT  
*Rzeszów University of Technology,  
Department of Applied Mechanics and Robotics*

Received (25 November 2008)

Revised (30 November 2008)

Accepted (15 December 2008)

This paper presents a new view on the coordination of elementary behavior of the wheeled mobile robot. The task was solved with the use of a Kohonen self-organizing neural network. This paper includes the results of behavioral control algorithm tests using the AmigoBot mobile robot.

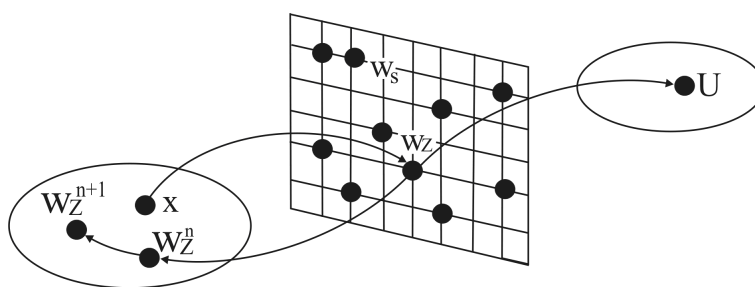
*Keywords:* Autonomous mobile robots, robot control, wheeled mobile robot, path planning, Kohonen self-organising map, behaviour control

### 1. Introduction

The evolution of autonomous mechanical systems is one of the main, fastest developing trends in the field of robotics. In spite of considerable progress in the area, there are still many matters that need to be addressed. Most of the difficulties arise while describing the natural working environment of an autonomous robot. Generally, the knowledge of the environment is incomplete, approximate, and not fixed. Thus, the main challenge of autonomous robotics amounts to the synthesis of a movement control algorithm for autonomous robots and hence fulfill complex tasks in imprecisely described working spaces. It is due to this problem complexity that there are as yet no universal methods for the planning and realization of autonomous wheeled mobile robot movement. Movement planning and its realization, one of the most significant notions for robotics, has been analyzed by many authors and their work published in many papers and dissertations [1, 2, 9], although much of this is theoretical and rarely has a practical application.

The main motive for engaging in planning of mobile robot non-collision movement is the need for methods with a practical application. This paper describes the results of investigations based on earlier work by a number of authors [3, 6, 7, 8].

A SOM (*Self Organizing Maps*) neural network [10], also known as a Kohonen feature map, is a self-learning network where topological mapping of input data occurs. In practice, WTM (*Winner-Takes-Most*) learning algorithms are used where, apart from the learning neuron itself, the neighboring neurons also influence the individual weighting.

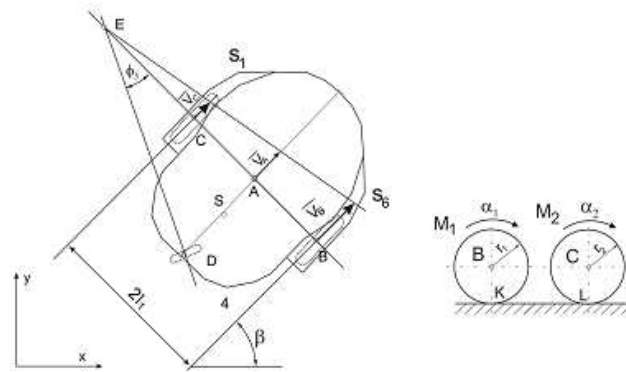


**Figure 1** Kohonen neural network

$$d(x, w_s) = \|x - w_s\|^2. \quad (1)$$
$$w_Z = \arg \min_s d(x, w_s). \quad (2)$$
$$w_Z^{n+1} = w_Z^n + c(n)h(w_Z, w_s)(x - w_Z^n), \quad (3)$$
$$h(w_Z, w_s) = \exp(-\frac{\|w_Z - w_s\|^2}{2\lambda^2}), \quad (4)$$
$$u_B = Dw_Z, \quad (5)$$

where the weight vector  $w_Z$  is dependent on the signal from the robot's sensors, and where  $D$  is a control identity matrix.

The mobile robot in question is a non-holonomic, two-degrees-of-freedom system. The drive wheel rotation angles  $\alpha_1$  and  $\alpha_2$  were treated as independent variables describing the movement of the mobile robot.



The velocity vector distribution of the characteristic points A, B, C resulted in the following:

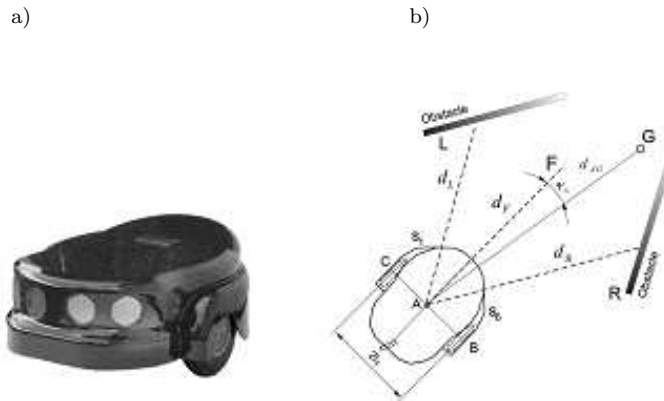
where:  $r_1 = r_2 = r$  – radii of the wheels,  $u_V$  – velocity of point A, and  $u_\beta$  is the angular velocity of the frame.

In the immobile system, the position of the mobile robot is described by coordinates defined as  $[x_A, y_A, \beta]^T$ . The kinematics of the wheeled mobile robot has been defined as:

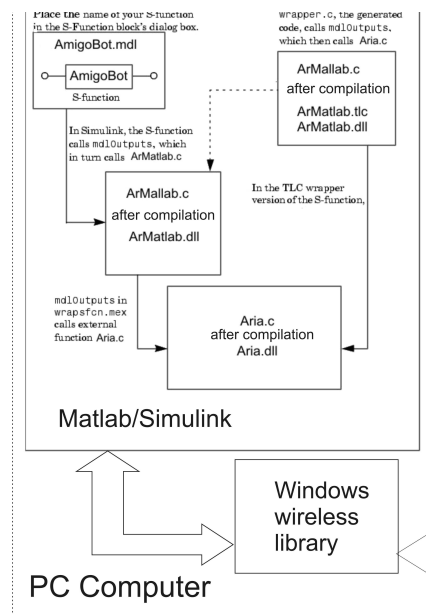
The wheeled mobile robot kinematics as described by relation (7) uses the quantities  $v_{Am}$  and  $\omega_m$  as the maximum velocities of point A and the frame of the mobile robot, respectively.

For the purpose of verification of the suggested behavioral control solutions, a rapid prototyping environment for the control algorithms was created with an environment based on Matlab/Simulink software and the AmigoBot mobile robot (Fig. 3a). The software selected was developed in-house at the Department of Applied Mechanics and Robotics, and was used during the verification of the algorithms presented here. It can be used to control one or several AmigoBot mobile robots,

facilitating the monitoring of a group of mobile robots, recording movement and distance sensor indications as well as operating the drive motors. Moreover, this flexible solution can be applied simultaneously to a number of robots (defined by the user) and meets the requirements of a real-time system.



**Figure 3** Mobile laboratory robot: a – robot overview, b – operation in an unknown environment



**Figure 4** Rapid prototyping environment overview

See Fig. 4 for an overview of the measurement and feedback environment. A Wi-Fi wireless network communication system was used to manage information between the robots and the control computer. The rapid prototyping environment controlling the operation of the group of mobile robots connected selected fragments of ARIA software using a Simulink S-function. In Simulink a library was created that included the "Robot" block in order to connect the control algorithms with the real object[4].

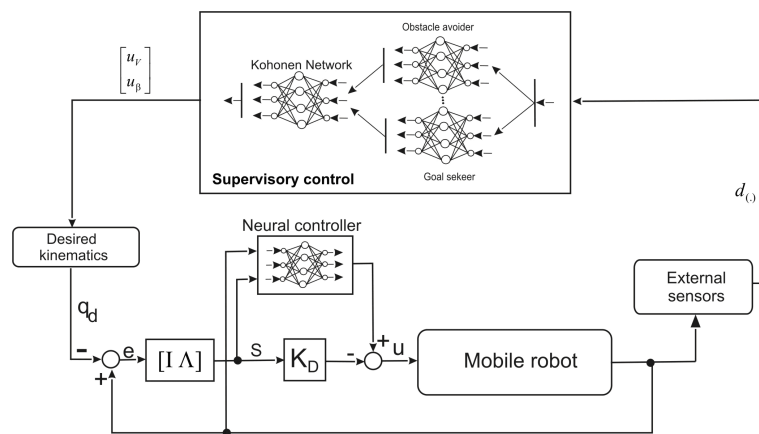
The robot type in question was equipped with 8 distance sensors. The distance sensors were located on the circumference of the robot (Fig. 4) and divided into groups:  $d_R$ ,  $d_L$ ,  $d_F$ , where:  $d_L = \min(s_1, s_2, s_3)$ ,  $d_F = \min(s_3, s_4)$ ,  $d_R = \min(s_4, s_5, s_6)$ . The value of any sensor indication was restricted by its operating scope and was within the range  $d_{\min} \leq d_{(\cdot)} \leq d_{\max}$ .

In order for the solution to be universal, i.e. for the control algorithm to become insensitive to the type of the sensors used, normalization of the distance indication was introduced.

The signals from the sensors were normalized as follows: right sensor distance measurement:  $d_R^N = d_R(d_R + d_L)^{-1}$ , left sensor distance measurement:  $d_L^N = d_L(d_R + d_L)^{-1}$ , and middle sensor distance measurement:  $d_F^N = d_F \rho^{-1}$ , where  $\rho$  is the sensor's maximum measurement range.

## 5. Supervisory control

The decision-making process concerning obstacle-avoidance navigation was intended to ensure achieving predefined intermediate and final states of the mobile robot. This area is in the domain of artificial intelligence and is practically achievable through a supervisory control system. The tasks for the mobile robots described here involved the following: "move through the middle of the free space", "move to the target", and coordination in view of behavioral control. See Fig. 5 for an overview of the control system.



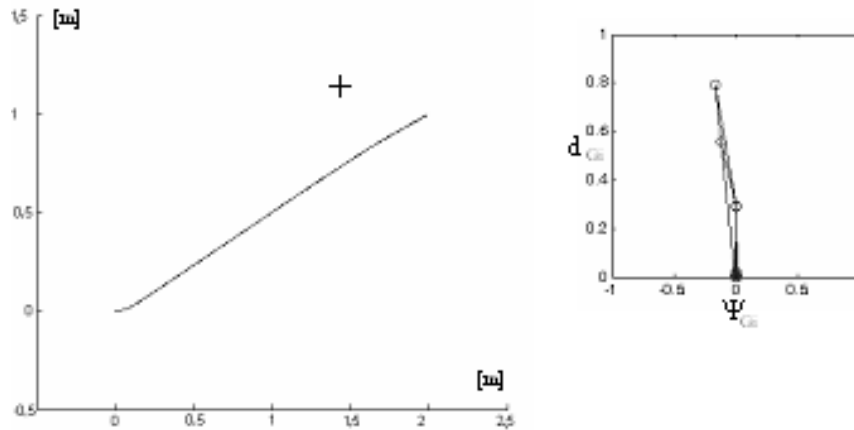
**Figure 5** Behavioral control system general overview

The article describes a supervisory control system which generates a predefined path according to the accepted behavioral task. However, the low level follow-up control system was not described and was realized through the AmigoSH software distributed by the manufacturer of the AmigoBot robots with the task of the correct realization of the generated path [4, 6].

### 5.1. "Move to the target" task

The "move to the target" task involved leading the mobile robot to a defined point within its working space. The task was fulfilled such that  $d(\dots)$ , the distance from the target  $d_{AG} = \|A, G\|$  and the angle of the deviation from the target  $\psi_G$  were minimized.

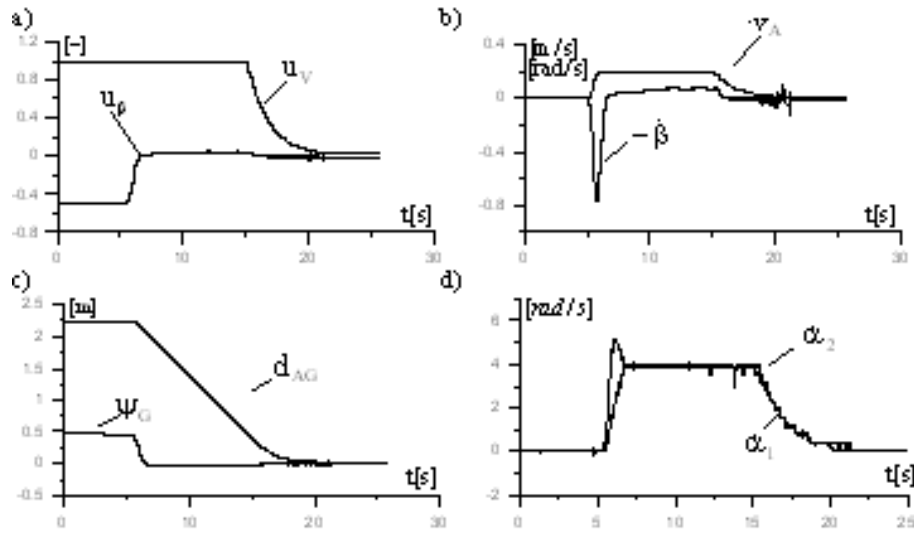
This task was achieved with the use of a Kohonen neural network, whose input signal was defined as the vector  $u_G = [d_{AG}, \psi_G]^T$ . Analysis of Fig. 4 shows that the value of  $d_{AG}$  can vary from 0 to  $d_{G \max}$ , whereas the angle  $\psi_G$  is included in the range  $\langle -\pi, \pi \rangle$ . The input signals  $d_{AG}$  and  $\psi_G$  were normalized with respect to the range  $\langle 0, 1 \rangle$ ,  $\langle -1, 1 \rangle$ . The winning neuron adapted its weighting towards the input vector according to the learning rule. The new coordinates of the winning neuron's location in the network were defined as  $w_Z = [d_{GZ}, \psi_{GZ}]^T$  and generated a control vector, defined by (5). Then a non-collision path, which was realized through a lower level of hierarchical control system, was generated on the basis of relation (7). Fig. 6 shows the path for point A of a robot obtained during a "move to the target" type task involving seven different targets. Fig. 7 shows selected signals obtained in the process of verification.



**Figure 6** Simulation of the "move to the target" task for different target coordinates: a – visualization of task fulfillment, b – neuron distribution in the Kohonen network after reaching target no. 3

### 5.2. Obstacle avoidance task

The "move through the middle of the free space" task involved generating a path to direct the mobile robot between the obstacles. The task was fulfilled with the use



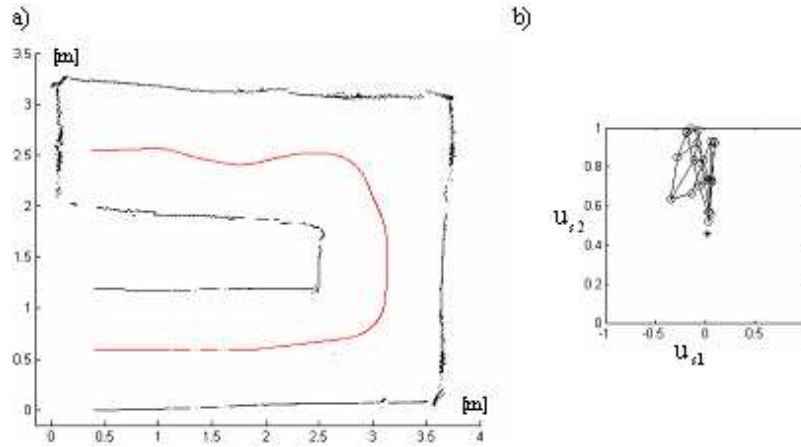
**Figure 7** Mobile robot's movement parameters during "move to the target" task fulfillment for target coordinates of  $x_G = 0.7[m]$ ,  $y_G = 0.7[m]$ : a – control signals  $u_\gamma, u_\beta$ , b – velocity of point A and angular velocity of the frame, c – distance between the robot and the target and the angle of the deviation from the target, d drive wheels angular velocities  $\dot{\alpha}_1, \dot{\alpha}_2$

of a Kohonen neural network. The input vector, subject to mapping by the neural network, was defined as  $u_S = [d_F^N, (d_L^N - d_R^N)]^T$ . Every neuron in the network had a corresponding vector of location, in this case defined for a neuron of weight  $w_i$  as  $[u_{s1i}, u_{s2i}]$ . The winning neuron adapted its weighting towards the input vector according to the learning rule. The new "winner" neuron location coordinates in the network, which through the substitution to the relation (5) became the output signal of the robot's neural navigation system, constituted an output from the neural work. Fig. 8 shows the generated path of point A for a mobile robot and the neural network weighting distribution during the final stage of the movement. Fig. 9 shows selected runs involving the fulfillment of the verified task

### 5.3. Behavior coordination

The central problems in trying to achieve satisfactory fulfillment of a complex task in behavioral control are in specifying the relationship between choices and coordinating the available elementary behaviors. In general, there are two groups of behavior coordinating method: competitive and cooperative. Here, the cooperative coordination of the elementary behavior was used.

Behavior control task verification was applied to the "move to the target and avoid the obstacles" task. A concave obstacle was assumed, for which the elementary tasks generate a stable equilibrium state resulting from the assumed obstacle symmetry (Fig. 10). The application of a Kohonen network to the fulfillment of elementary behavior made it possible to solve the task in this environment.



**Figure 8** Simulation of the "move through the middle of the free space" task a – visualization of task fulfillment, b – Kohonen network neuron distribution

For this task, the control signal  $u_B$  was defined as:

$$u_{B1} = [u_{vOA}, u_{\beta OA}]^T \quad (8)$$

whereas the control signal for the "move to the target" task was defined as:

$$u_{B2} = [u_{vGS}, u_{\beta GS}]^T \quad (9)$$

The following were adopted as elements of behavioral control vector  $u_B = [u_v, u_\beta]^T$ :

$$u_v = \min(u_{vGS}, u_{vOA})^T \quad (10)$$

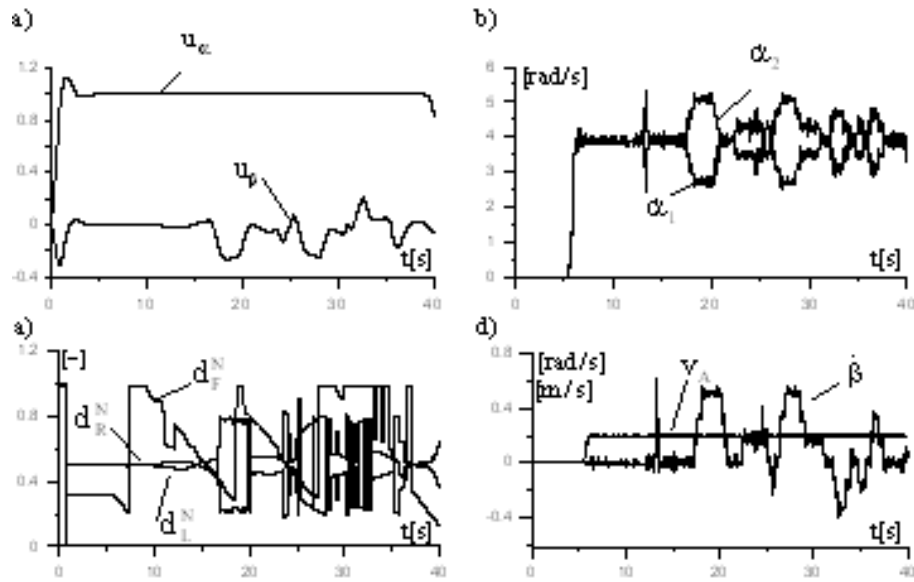
$$u_\beta = b_1 u_{\beta GS} + b_2 u_{\beta OA} \quad (11)$$

where:  $b_i$  was chosen experimentally, so that the assumed task was achieved correctly.

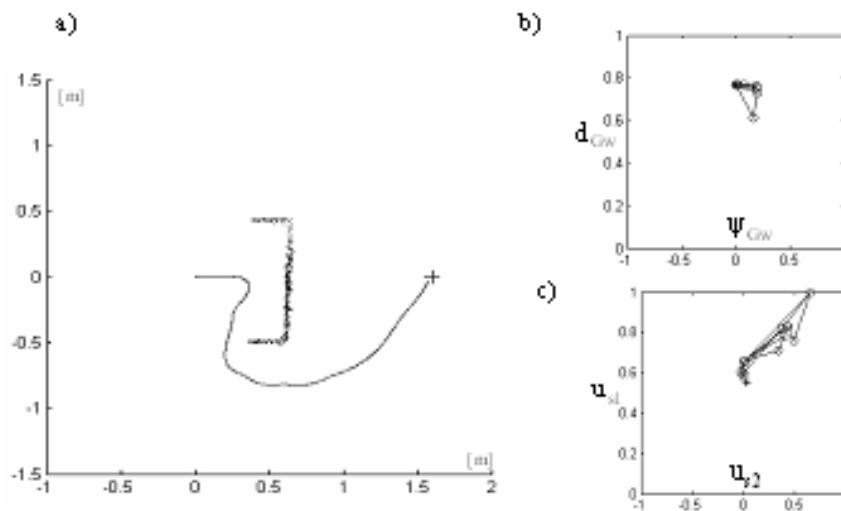
Figure 10 gives the generated path of mobile robot movement and the neural network weighting distribution at the final stage of the movement during the "move to the target and avoid the obstacles" task. Figure 11 gives selected runs involving the fulfillment of the verified task.

Fig. 10 Simulation of the "move to the target and avoid the obstacles" task: a – visualization of task fulfillment, neuron distribution in the Kohonen network after task realization b – "move to the target", c – "move through the middle of the free space"

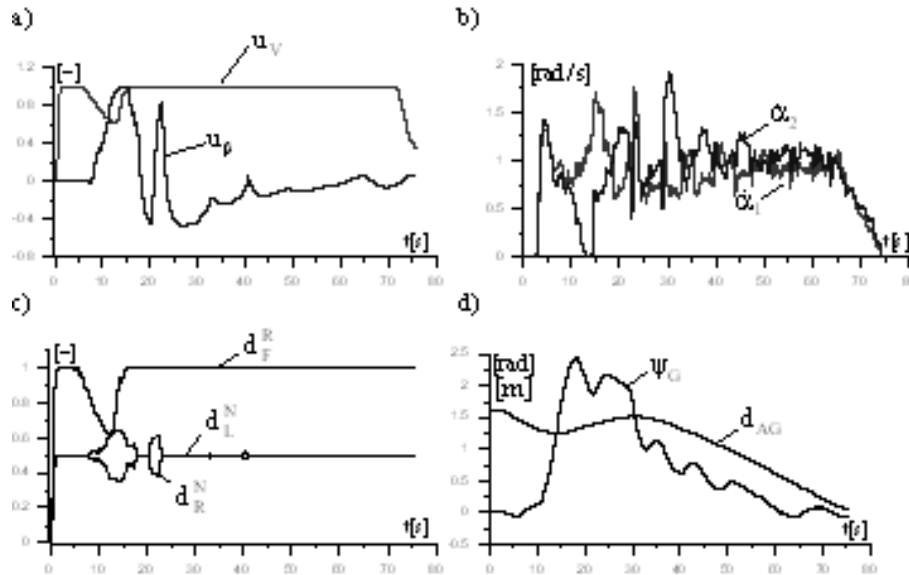




**Figure 9** The robot movement parameters: a – control signals  $u_V, u_\beta$ , b – drive wheel angular velocities  $\dot{\alpha}_1, \dot{\alpha}_2$ , c – normalized distances from the obstacles, d – velocity of point A and angular velocity of the frame



**Figure 10** Simulation of the "move to the target and avoid the obstacles" task: a - visualization of task fulfillment, neuron distribution in the Kohonen network after task realization b - "move to the target"  $\alpha_1, \alpha_2$ , c - "move through the middle of the free space"



**Figure 11** Mobile robot movement parameters during the "move to the target and avoid the obstacles" task fulfillment: a - control signals  $u_V, u_\beta$ , b - drive wheel angular velocities  $\dot{\alpha}_1, \dot{\alpha}_2$ , c - normalized distances from the obstacles, d - distance between the robot, the target and the angle of the deviation from the target

Fig. 11. Mobile robot movement parameters during the "move to the target and avoid the obstacles" task fulfillment: a – control signals  $u_V, u_\beta$ , b – drive wheel angular velocities  $\dot{\alpha}_1, \dot{\alpha}_2$ , c – normalized distances from the obstacles, d – distance between the robot, the target and the angle of the deviation from the target.

## 6. Conclusions

The use of the generalizing properties of Kohonen neural networks allowed the implementation of comprehensive behavioral control with two types of elementary behavior only, constituting an extension of the existing methods of solving stable and unstable equilibrium states generated by concave obstacles. The suggested rapid prototyping environment, based on the AmigoBot mobile robots, enabled the chosen solution to be verified and hence confirming the results obtained in the numerical simulations.

## References

- [1] **Arkin, R.C.:** Behaviour-Based Robotics, *The MIT Press*, **1998**.
- [2] **Berenstain, J. and Koren, J.:** Real time obstacle avoidance for fast mobile robots, *IEEE Transaction on Systems, Man and Cybernetics*, Vol.19, No.5, s.1179-1186, **1989**.
- [3] **Giergiel, J., Hendzel, Z. and Burghardt, A.:** Nawigacja zachowaniami mobilnego minirobota, *Przegld Mechaniczny*, zeszyt 7-8, s. 47-50, **2004**.

- [4] **Burghardt, A.:** Proposal for a rapid prototyping environment for algorithms intended for autonomous mobile robot control, Edit. T. Kapitaniak, *Mechanics and Mechanical Engineering*, WPL, 1, **2008**.
- [5] **Giergiel, M., Hendzel, Z. and W. Żylski:** Modelowanie i sterowanie mobilnych robotów koowych, *PWN*, Warszawa, **2002**.
- [6] **Hendzel, Z.:** Collision free path planning and control of wheeled mobile robot using Kohonen self-organising map, , *Bulletin of The Polish Academy of Sciences Technical Sciences*, Vol. 53, No. 1, **2005**.
- [7] **Hendzel, Z. and Burghardt, A.:** Sterowanie behawioralne minirobotu kołowego, *Pomiary Automatyka Kontrola* 11, s. 26-29, **2004**.
- [8] **Hendzel, Z. and Burghardt, A.:** Sterowanie behawioralne minirobotu kołowego, Postępy Robotyki Sterowanie robotów z percepcją otoczenia, Red. K. Tchonia, *Wydawnictwa Komunikacji i cznoci*, Warszawa, s.169-176, **2005**.
- [9] **J.C. Latombe.** *Robot Motion Planning*, Kluwer Academic Publishers, **1991**.
- [10] **T. Kohonen:** Self-Organising Maps, Springer Series in Information Sciences, *Springer*, Berlin, Heidelberg, vol. 30, **1995**.

