BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES
Vol. 53, No. 1, 2005

Collision free path planning and control of wheeled mobile robot using Kohonen self-organising map

Z. HENDZEL*

Department of Applied Mechanics and Robotics, Rzeszów University of Technology
8 Powstańców Warszawy St., 35-959 Rzeszów, Poland

Abstract. In this paper we propose a sensor-based navigation method for navigation of wheeled mobile robot, based on the Kohonen self-organising map (SOM). We discuss a sensor-based approach to path design and control of wheeled mobile robot in an unknown 2-D environment with static obstacles. A strategy of reactive navigation is developed including two main behaviours: a reaching the middle of a collision-free space behaviour, and a goal-seeking behaviour. Each low-level behaviour has been designed at design stage and then fused to determine a proper actions acting on the environment at running stage. The combiner can fuse low-level behaviours so that the mobile robot can go for the goal position without colliding with obstacles one for the convex obstacles and one for the concave ones. The combiner is a softswitch, based on the idea of artificial potential fields, that chooses more then one action to be active with different degrees at each time step. The output of the navigation level is fed into a neural tracking controller that takes into account the dynamics of the mobile robot. The purpose of the neural controller is to generate the commands for the servo-systems of the robot so it may choose its way to its goal autonomously, while reacting in real-time to unexpected events. Computer simulation has been conducted to illustrate the performance of the proposed solution by a series of experiments on the emulator of wheeled mobile robot Pioneer-2DX.

Key words: wheeled mobile robot, path planning, Kohonen self-organising map, behaviour control, tracking control.

1. Introduction

Expansion of the range of robot tasks motivated by applications such as office cleaning, cargo delivery and assistance to disabled people, among others an increase in robot autonomy created a need to generate trajectories on-line. In order to achieve its task, the mobile robot must be able to realise a collision-free trajectory among the obstacles of the environment. Planning a path for a mobile robot means to find a continuous trajectory that leads the mobile robot from the initial position to the goal position. There are a lot of studies on trajectory generation for robots using various approaches e.g. [1–7]. The two main approaches for solving the path-planning problem are global and local methods. Model-based systems address the path finding problem in a global way, while sensor-based systems consider it in a local way. The artificial potential field method is a popular tool for on-line trajectory generation with inherent collision avoidance [1,3,5,6]. A comprehensive overview of the reactive navigation, robot behaviour and behaviour-based control field we can find in the book [1,4,6]. Several neural network models e.g. [2–7,10] were proposed to generate real-time trajectories. Several researchers have already argued the importance of looking at a mobile robot as a set of elementary behaviours [1,3–5]. Elementary behaviours are important components of reactive control in which mobile robot must continuously interact with their environment. Reactive control means that all decisions are based on the currently perceived sensory information [1,4,11]. Numerous behaviour co-ordination mechanisms have been proposed. For a detailed overview, discussion, and comparison of behaviour co-ordination mechanisms see [1,6,7]. Behaviour co-ordination mechanisms can be divided into two main classes: arbitration and command fusion [4]. Command fusion mechanisms provides for a co-ordination scheme that allows all behaviours to simultaneously contribute to the control of the system in a co-operative manner.

Although many solutions have already been reported in the literature, the continuing development of new proposals suggests that this field has not settled down yet.

In this paper we propose a sensor-based navigation method for navigation of wheeled mobile robot, based on the SOM.

We used a sensor-based system which does not need a model of the workspace and this is already an advantage in itself. Also, it is computationally inexpensive as it just reacts to sensor readings. But a sensor-based system generates sub-optimal paths, and it may get trapped into dead-ends. Moreover, it is difficult to program a sensor-based system, as we have to predict every possible situation the robot will encounter.

We discuss a sensor-based approach to path design and control of simple individual behaviours of wheeled mobile robot in an unknown 2-D environment with static obstacles. A strategy of reactive navigation is developed

*e-mail: zenhen@prz.rzeszow.pl

including two main behaviours: a reaching the middle of a collision-free space behaviour, and a goal-seeking behaviour. Each low-level behaviour has been designed at design stage and then fused by the combiner of behaviours to determine a proper action acting on the environment at running stage. The combiner can fuse low-level behaviours so that the mobile robot can go for the goal position without colliding with obstacles one for the convex obstacles and one for the concave ones. The combiner is a soft switch, based on the idea of artificial potential fields, that chooses more then one action to be active with different degrees at each time step. The output of the navigation level is fed into a neural tracking controller that takes into account the dynamics of the mobile robot. The purpose of the neural controller is to generate the commands for the servo-systems of the robot so it may choose its way to its goal autonomously, while reacting in real-time to unexpected events. This research continues prior researches of the author, concerning collision free path planning and control of mobile wheeled-robots e.g. [12–14]. The rest part of the paper is organised as follows. Dynamic equations of the mobile 2-wheeled-robot and control properties are included in Section 2. Section 3 displays a SOM neural network. Section 4 includes results of path finder tests, obtained after numerical simulation. Section 5 summarizes the results of the research.

2. Modelling and control properties

The mechanical structure of the mobile robot, like Pioneer-2DX, is shown in Fig. 1.

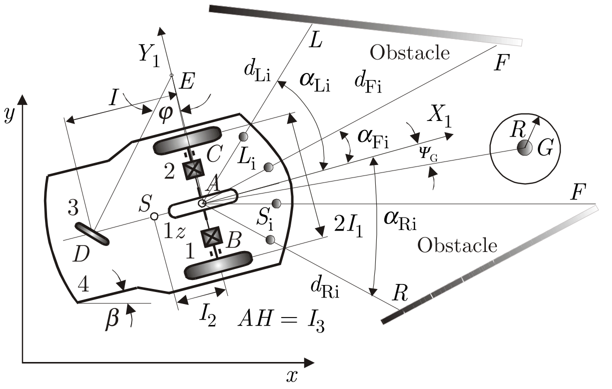


Fig. 1. The schematic diagram of mobile robot

Presented robot has two degrees of freedom. In the word co-ordinates a posture is defined as $[x_{\mathrm{A}}, y_{\mathrm{A}}, \beta]^{\mathrm{T}}$, where $(x_{\mathrm{A}}, y_{\mathrm{A}})$ is the position of the point A, and β is the heading angle of the robot with respect to absolute co-ordinates (x, y). The mobile robot's kinematics is defined by [15]

$$\begin{bmatrix} \dot{x}_{\mathrm{A}} \\ \dot{y}_{\mathrm{A}} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} V_{\mathrm{Am}} \cos(\beta) & 0 \\ V_{\mathrm{Am}} \sin(\beta) & 0 \\ 0 & \omega_{\mathrm{m}} \end{bmatrix} \begin{bmatrix} \mathrm{u_v} \\ \mathrm{u}_\beta \end{bmatrix} \tag{1}$$

with the maximum linear V_{Am} and angular ω_{m} speeds and $\mathrm{u_v}$ is the multiplying coefficient applied to the maximum linear velocity of point A of the robot and u_β is the multiplying coefficient applied to the maximum angular velocity of the frame. That coefficients form a vector $\mathrm{u_B} = [\mathrm{u_v}, \mathrm{u}_\beta]^{\mathrm{T}}$ generated by SOM. Tacking α_1, α_2 – angles of the self-turn of the propel wheels as independent co-ordinates, then from velocity vectors of points A, B, C and (1) we receive velocities expressed in mobile base co-ordinates

$$\dot{\alpha}_1 = \frac{V_{\mathrm{A}}}{r_1} + \dot{\beta}\frac{l_1}{r_1} \tag{2}$$

$$\dot{\alpha}_2 = \frac{V_{\mathrm{A}}}{r_2} - \dot{\beta}\frac{l_1}{r_2} \tag{3}$$

where $V_{\mathrm{A}} = V_{\mathrm{m}}\mathrm{u_v}$, $\dot{\beta} = \omega_{\mathrm{m}}\mathrm{u}_\beta$, $r_1 = r_2 = r$ is radii of the wheels and l_1, l_2 are adequate distances which result from the geometry of the system. Equations (2) and (3) define a prescribed motion trajectory $x_{\mathrm{d}} = [\alpha_1, \dot{\alpha}_1, \alpha_2, \dot{\alpha}_2]^{\mathrm{T}}$ which is determined in a higher-level path planner built on a SOM.

Using Maggi's formalism [15] the dynamics of wheeled mobile robot can be written as

$$\begin{bmatrix} a_1 + a_2 + a_3 & a_1 - a_2 \\ a_1 - a_2 & a_1 + a_2 + a_3 \end{bmatrix} \begin{bmatrix} \ddot{\alpha}_1 \\ \ddot{\alpha}_2 \end{bmatrix} + \begin{bmatrix} 0 & 2a_4(\dot{\alpha}_2 - \dot{\alpha}_1) \\ -2a_4(\dot{\alpha}_2 - \dot{\alpha}_1) & 0 \end{bmatrix} \begin{bmatrix} \dot{\alpha}_1 \\ \dot{\alpha}_2 \end{bmatrix} + \begin{bmatrix} a_5 \operatorname{sgn} \dot{\alpha}_1 \\ a_6 \operatorname{sgn} \dot{\alpha}_2 \end{bmatrix} = \begin{bmatrix} M_1 \\ M_2 \end{bmatrix} \tag{4}$$

where a is a vector of the mobile robot parameters, which results from the system geometry, weights distribution and motions resistance and is defined by the following equation:

$$a_1 = (2m_1 + m_4)\left(\frac{r}{2}\right)^2, \quad a_2 = (2m_1 l_1^2 + m_4 l_2^2 + I_S)\frac{r}{2l_1},$$
$$a_3 = I_{Z1} = I_{Z2}, \quad a_4 = m_4\left(\frac{r}{2}\right)^2 \cdot \left(\frac{rl_2}{l_1^2}\right),$$
$$a_5 = N_1 f_1, \quad a_6 = N_2 f_2 \tag{5}$$

and $m_1 = m_2$, m_4 – substitute mass of wheels 1 and 2 and the frame, $I_{Z1} = I_{Z2}$ – substitute massinertia moments of the adequate wheels relative to the axes of self-turn of the wheels. It was assumed that the axes of reference system connected with part "i" are the main central axes of inertia, however N_1, N_2 are pressure forces of the wheels 1 and 2, f_1, f_2 are turn friction coefficients of adequate wheels, $\mathrm{u} = [\mathrm{M}_1, \mathrm{M}_2]^{\mathrm{T}}$ is a vector of the moments propelling driving wheels, l, l_1, l_2, are adequate distances which result from the geometry of the system, $r_1 = r_2 = r$ are radiuses of adequate wheels.

The objective of mobile robot control is to select the control vector $\mathrm{u} = [\mathrm{M}_1, \mathrm{M}_2]^{\mathrm{T}}$ so that the mobile robot follows a prescribed motion trajectory $x_{\mathrm{d}}(t)$. The control objective can be achieved by defining a desired trajectory $x_{\mathrm{d}}(t)$ which is determined in a higher-level path planner built on a SOM as depicted in the Fig. 3. Using the Eq. (4) written in the form

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + F(\dot{q}) = u \tag{6}$$

and define the tracking error $e(t)$ and filtered tracking error $s(t)$ by [15–17]

$$e = q_\mathrm{d} - q, \quad s = \dot{e} + \Lambda e \tag{7}$$

the mobile robot dynamics are expressed in terms of the filtered error

$$\mathrm{M}\dot{s} = -\mathrm{C}s + f(x) - u \tag{8}$$

where unknown non-linear is defined as

$$f(x) = \mathrm{M}(q)(\ddot{q}_\mathrm{d} + \Lambda\dot{e}) + C(q,\dot{q})(\dot{q}_\mathrm{d} + \Lambda e) + F(\dot{q}). \tag{9}$$

A sort of approximation-based controller is derived by setting

$$\mathrm{u} = \hat{f} + Ks - v(t) \tag{10}$$

with $\hat{f}$ an estimate of $f(x)$, $Ks = Ke + K\Lambda e$ an outer PD loop and v(t) an auxiliary signal to provide robustness in the face of disturbances and modelling errors. Suppose that a neural network (NN) is used to approximate the non-linear function (9) according to [16]

$$f(x) = W^\mathrm{T} S(x) + \varepsilon \tag{11}$$

with W the ideal approximating weights and S(x) is activation functions. Then an estimate of f(x) is given by

$$\hat{f}(x) = \hat{W}^\mathrm{T} S(x) \tag{12}$$

but the control law (10) becomes

$$u = \hat{W} S(x) + Ks - v(t). \tag{13}$$

For a more complete overview of this NN approximation see [15]. Let the desired trajectory be bounded, reconstruction error ε is equal to zero and the control signal for (6) be given by (13) with $v(t) = 0$ and NN weight tuning provided by

$$\dot{\hat{W}} = \Gamma S(x) s^\mathrm{T} \tag{14}$$

with $\Gamma > \Gamma^\mathrm{T} > 0$ a constant design matrix. Then the tracking error $s(t)$ goes to zero with the weight estimated are bounded [13].

3. Kohonen map

The SOM is an unsupervised learning neural network method that produces a similarity graph of input data. The algorithm of this method operates recursively i.e. upon each presentation of input data vector, it performs a search for the neuron with minimum distance measure to input vector. This winning neuron is adapted by learning rules. In practice, Winner-Takes-Most (WTM) learning algorithms are applied in which not only the winning neuron but also neurons from its neighborhood update their weights.

A SOM is a two-layered network [8–11,18] consisting of an input layer of neurons directly and fully connected to an output layer. In this paper, the output layer is organized as a two-dimensional grid and w_s is the weight vector (reference vector) associated to the neuron placed at position s on grid.

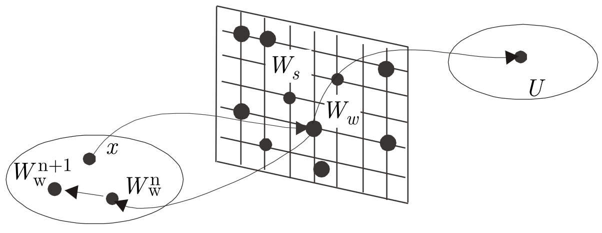


Fig. 2. The Kohonen map

The network is trained by unsupervised learning on a set of elements $\{x_1, x_2, \ldots, x_n\}$ and for each vector x presented to the input layer a competition between the neurons takes place. Each neuron calculates the distance

$$\mathrm{d}(x, w_\mathrm{s}) = \|x - w_\mathrm{s}\|^2. \tag{15}$$

The neuron w which weight vector is the closest to x is the winner of the competition

$$w = \arg\min_\mathrm{s} \mathrm{d}(x, w_\mathrm{s}) \tag{16}$$

and w is awarded the right to learn the input vector, i.e. to move closer to it in input space:

$$w_\mathrm{w}^{n+1} = w_\mathrm{w}^n + c(t)h(s,w)(x - w_\mathrm{w}^n) \tag{17}$$

Figure 2 illustrates the weight change process of neuron w in the original input space. In Eq. (17), $c(t)$ is the learning rate, a real parameter that decreases linearly with the learning process as

$$c(t) = c(0)(1 - tT^{-1}) \tag{18}$$

and $h(s,w)$ defines e.g. the Gaussian kernel weight of $\|w - s\|$.

The learning step is extended also to the neighbours of the winner neuron w. The neighbours of neuron w are those output elements whose distance to w, measured on the grid , is not greater than a neighbourhood parameter. At the beginning of the learning process, the neighbourhood parameter is large, as time progresses, fewer neurons are allowed to become closer to the presented input vector.

If a SOM is trained with perceptions as inputs, then its reference vectors will represent prototypical perceptions. The task of the SOM is to construct a set of prototypical perceptions out of a set of perceptions experienced by the robot during motion. The neuron which is the winner of the competition formulates the best prototypical perceptions. Thus the control vector $\mathrm{u_B}$ would be related to the sensory input vector by the equation

$$\mathrm{u_B} = \mathrm{D}w_\mathrm{w} \tag{19}$$

where D is a matrix of control parameters.

4. Path finder

Described mobile robot is equipped with eight an ultrasonic sensors (few of them are depicted in Fig. 1). The radius of the sensors $\mathrm{s_i}$, is $\mathrm{L_i}$ and sensors are divided into three group. A group is composed of three, two and three neighbouring sensors, gives a distance to the obstacle $d_{\mathrm{Li}}, d_{\mathrm{Fi}}, d_{\mathrm{Ri}}$, in its field of view $d_{\min} \leq d_{(\cdot)} \leq d_{\max}$, where and each sensor covers an angular view which are oriented by angles $\alpha_{\mathrm{Li}}, \alpha_{\mathrm{Fi}}, \alpha_{\mathrm{Ri}}$ respectively.

To solving the trajectory tracking problem for nonholonomic mobile robot with considering the vehicle dynamics [12–15], it is assumed that the current desired kinematics is generated at each time step, by SOM navigator which generates vector of multiplying coefficient $\mathrm{u_B} = [\mathrm{u_v}, \mathrm{u}_\beta]^{\mathrm{T}}$, based on the environment information, $d_{(\cdot)}$.

In this work, two navigation task is discussed: reaching the middle of a collision-free space behaviour and goal-seeking behaviour. Each behaviour has been fused in a cooperative manner based on the idea of artificial potential fields, to determine a proper actions is the environment at running stage. Diagram of the navigator and controller architecture is depicted in Fig. 3.

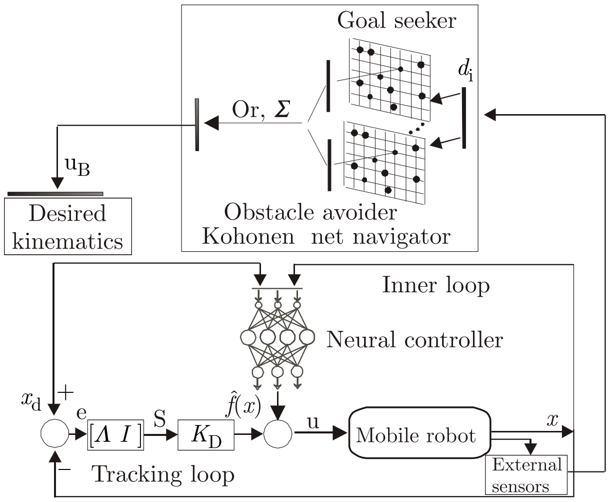


Fig. 3. Navigator and controller

4.1. Goal seeking behaviour. The task goal seeking relay on leading mobile robot to desired point G depicted in Fig. 1. It means minimizing of the distance $\mathrm{d_G} = [\mathrm{A}, \mathrm{G}]$ and the ψ_{G} angle which is the angular deviation needed to reach the goal. In this task the used navigator is built with a SOM network. This task adopts an egocentric representation of the sensory input vector $\mathrm{u_G} = [\psi_{\mathrm{G}}, \mathrm{d_G}]^{\mathrm{T}}$ where $\psi_{\mathrm{G}} \in (-\pi, \pi][\mathrm{rad}]$ and $\mathrm{d_G} \in [0, \mathrm{d_{G\,max}}][\mathrm{m}]$. The elements are normalized to the intervals $(-1, 1]$, $[0, 1]$ respectively. At the goal state at time T, $\mathrm{u_G} = [\psi_{\mathrm{G}}, 0]^{\mathrm{T}}$ for any ψ_{G}. Each neuron i in the SOM has a sensory weight vector $w_{\mathrm{i}} = [\psi_{\mathrm{Gi}}, \mathrm{d_{Gi}}]^{\mathrm{T}}$ that encodes a region in X centered at w_{i}. Based on each incoming sensory input $\mathrm{u_G}$, the SOM constructs a set of prototypical perceptions out of a set of perceptions experienced by the robot during motion. The winning neuron determines the vector of multiplying coefficient (19). Extensive simulation tests were done to validate and test the method exposed above. The workspace is a typical indoor square environment with dimensions $10 \times 10[\mathrm{m}^2]$. A SOM of 5×5 units arranged in a rectangular lattice was used. The maximum distance $\mathrm{d_{G\,max}}$ measured between the goal and the sensor was 6 [m]. The control vector $\mathrm{u_B} = \mathrm{u_G}$ was generated by the Eq. (19) with $\mathrm{D} = \mathrm{I}$ as unity matrix. All initial weights w_{S} were set randomly. The mobile robot starts with $(x_{\mathrm{A}}, x_{\mathrm{B}}, \beta) = (2, 2, 0)$ in all tests. The parameters set $\{V_{\mathrm{m}}, \omega_{\mathrm{m}}\}$ are equal for all tests $\{0.4[\mathrm{m/s}], 0.3[\mathrm{rad/s}]\}$. The parameters used in tracking control are presented in [6].

Figure 4a presents an numerical example of navigation. Figure 4b prints a set of prototypical perceptions out of a set of perceptions experienced by the robot during motion. The action $\mathrm{u}_\beta, \mathrm{u_v}$, generated by SOM navigator for point G3(9,9) are depicted in the Fig. 4c . Real linear velocity of point A and angular velocity of the frame, where $\dot{\beta} = \omega$ and angular velocities of the wheels, where $\dot{\alpha}_1 = \alpha_{1\mathrm{p}}$, $\dot{\alpha}_2 = \alpha_{2\mathrm{p}}$ are depicted in the Fig. 4d, 4e respectively. For this example driving torque M1 and M2 [Nm] generated by neural controller are presented in the picture 4f.

Test results show that the mobile robot was able to navigate to reach the goals and the trajectory of the mobile robot is very smooth. To be more specific the learning is done on-line. On the beginning of the learning each neuron to small random value was initialised. Based on each incoming sensory input $\mathrm{u_G}$, the weights are then adjusted in real time and mobile robot begins to move and so on. Based on results received from tests and repeated realizations of tasks it is concluded that learning process has no effect on robot motion. No significant deviation from the previous trajectories is observed.

4.2. Reaching the middle of a collision-free space behaviour. The objective of this behaviour is to keep the mobile robot at the middle of a collision-free space. In order to achieve this task, the mobile robot must be able to realise a collision-free trajectory among the obstacles of the environment. The obstacle avoidance task uses the SOM which is self-organised in the same way as the goal seeking behaviour.

Let the input variables of the SOM navigator are respectively normalised measured distances on the right $d_{\mathrm{R}}^n = d_{\mathrm{R}}(d_{\mathrm{R}} + d_{\mathrm{L}})^{-1}$, on the left $d_{\mathrm{L}}^n = d_{\mathrm{L}}(d_{\mathrm{R}} + d_{\mathrm{L}})^{-1}$ and in the front $d_{\mathrm{F}}^n = d_{\mathrm{F}}\eta^{-1}$, with $d_{\mathrm{L}} = \min(\mathrm{s_2}, \mathrm{s_3})$, $d_{\mathrm{F}} = \min(\mathrm{s_4}, \mathrm{s_5})$, $d_{\mathrm{R}} = \min(\mathrm{s_6}, s_7)$ as depicted in the Fig. 1 and η is a distance beyond which the obstacle is not taken into account. This task adopts the sensory input vector $\mathrm{u_S} = [d_{\mathrm{F}}^n, (d_{\mathrm{L}}^n - d_{\mathrm{R}}^n)]^{\mathrm{T}}$. Each neuron in the SOM has a sensory weight vector $\mathrm{w_i} = [\mathrm{u_{s1i}}, \mathrm{u_{s2i}}]^{\mathrm{T}}$ as the neuron in the goal seeking behaviour. Based on each incoming sensory input $\mathrm{u_S}$, the SOM constructs a set of prototypical perceptions out of a set of perceptions experienced by the

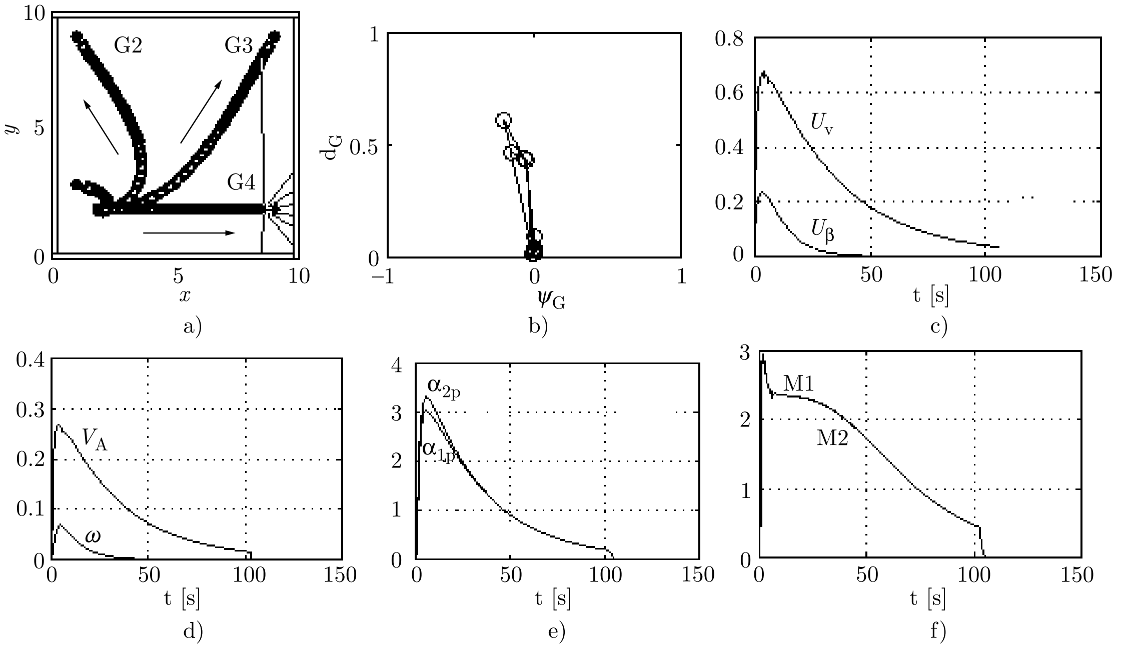


Fig. 4. Goal seeking behaviour

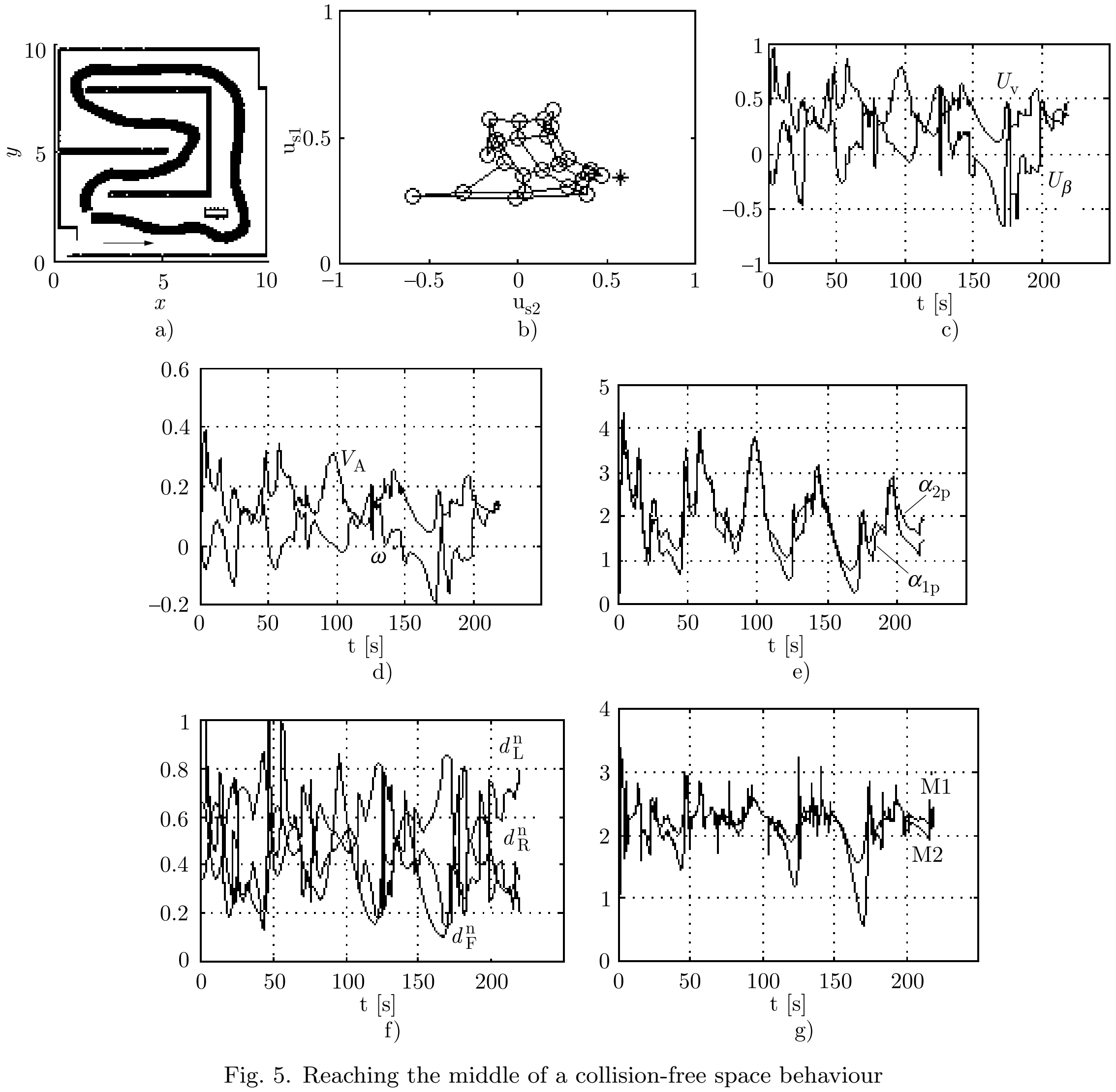


Fig. 5. Reaching the middle of a collision-free space behaviour

robot during motion. In this task the winning neuron determines the vector of multiplying coefficient in (19) with $\mathrm{u_B} = \mathrm{u_S}$. Computer simulation have been conducted to illustrate the performance of the proposed solution. In this simulations we used the same parameters as in point 4.1.

An example of the resulting navigation reaching the middle of a collision-free space behaviour and control is shown in Fig. 5. Figure 5a depicts the trajectory performed by the mobile robot to reach the middle of a collision-free space of the complex workspace.

Figure 5b prints a final set of prototypical perceptions out of a set of perceptions experienced by the robot during motion. The actions u_β, u_v, generated by SOM navigator are depicted on the Figs. 5c. The real linear velocity of point A and angular velocity of the frame, where $\dot{\beta} = \omega$ and angular velocities of the wheels, where $\dot{\alpha}_1 = \alpha_{1\mathrm{p}}$, $\dot{\alpha}_2 = \alpha_{2\mathrm{p}}$ and are depicted in the Fig. 5d, 5e respectively. Figure 5f shows the normalised measured distances on the right d^n_R, on the left d^n_L and in the front d^n_F during the motion. For this example driving torque M1 and M2 [Nm] generated by neural controller are presented in Fig. 5g.

Another experiment show that the learning process has effect on robot motion. For repeated motion with initial weights w_S, as depicted in Fig. 5b, an error of the behaviour was smaller.

4.3. Fusion of elementary behaviours. In this section, the proposed combiner is applied to two behaviours: obstacle avoidance and goal seeking, to show its performance and applicability. When the mobile robot encounters an obstacle which obstructs the goal, these two behaviours are in the conflict. In this paper we adopt the concept based on the artificial potential fields [1,6] to solve this conflict. These approaches use a vector summation i.e. outputs from different behaviours are combined by vector summation. In other words the individual decision of different behaviours are fused into combined decision. In the proposed navigator low-level modules are denoted as obstacle avoider (OA) and goal seeker (GS). Each module receives distances sensed by the ultrasonic sensors $d_{(\cdot)}$ and produces output signals. The GS determines the action $\mathrm{u_{B1}} = [\mathrm{u_{vGS}}, \mathrm{u_{\beta GS}}]^\mathrm{T}$ for the behaviour of goal seeking, while the OA determines the action $\mathrm{u_{B2}} = [\mathrm{u_{vOA}}, \mathrm{u_{\beta OA}}]^\mathrm{T}$ for the behaviour of obstacle avoidance. These two behavioural modules work independently and their actions are fused to produce action $\mathrm{u_B} = [\mathrm{u_v}, \mathrm{u_\beta}]^\mathrm{T}$ for the navigation. It is assumed that each low-level module has been well designed. The final multiplying coefficient applied to the maximum angular velocity of the frame is generated by equation

$$u_\beta = a_1 u_{\beta\mathrm{GS}} + a_2 u_{\beta\mathrm{OA}} \tag{20}$$

where a_1 and a_2 are coefficients adjusted by experimentation to get the best trajectory generation. A multiplying coefficient for the linear speed is given by

$$u_\mathrm{v} = \min(u_\mathrm{vGS} + a_2 u_\mathrm{vOA}) \text{ if } \mathrm{d_G} \geq R \tag{21}$$

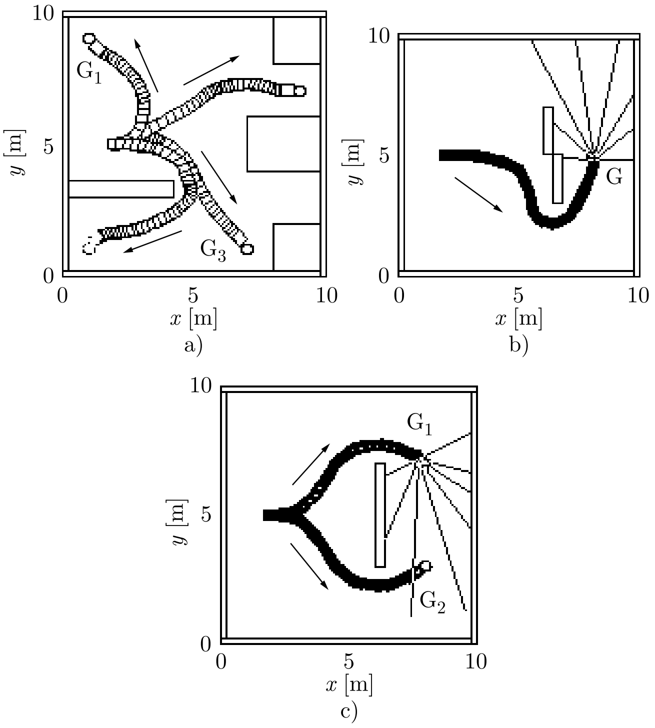


Fig. 6. Trajectories of point A of mobile robot

It is supposed that no obstacle exists in the circle of $R = 1$[m] diameter. The parameters set $\{a_1, a_2\}$ are equal for all tests $\{1, 1.5\}$. Based on each incoming set of prototypical perceptions, as described in earlier points, the winning neuron for each low-level task determines the vector of action $\mathrm{u_{B1}}, \mathrm{u_{B2}}$. The successive activation of the different behaviours can be observed in Fig. 6. The mobile robot received the mission to reach a given goal position $\mathrm{G_i}$ from the given start position with reaching the middle of a collision-free space. The environment was considered as fixed and completely unknown.

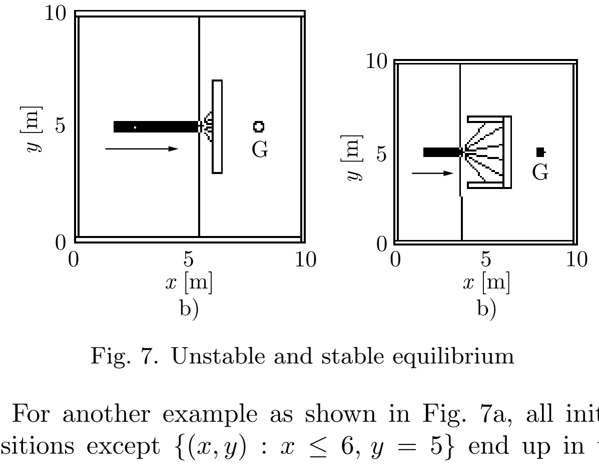
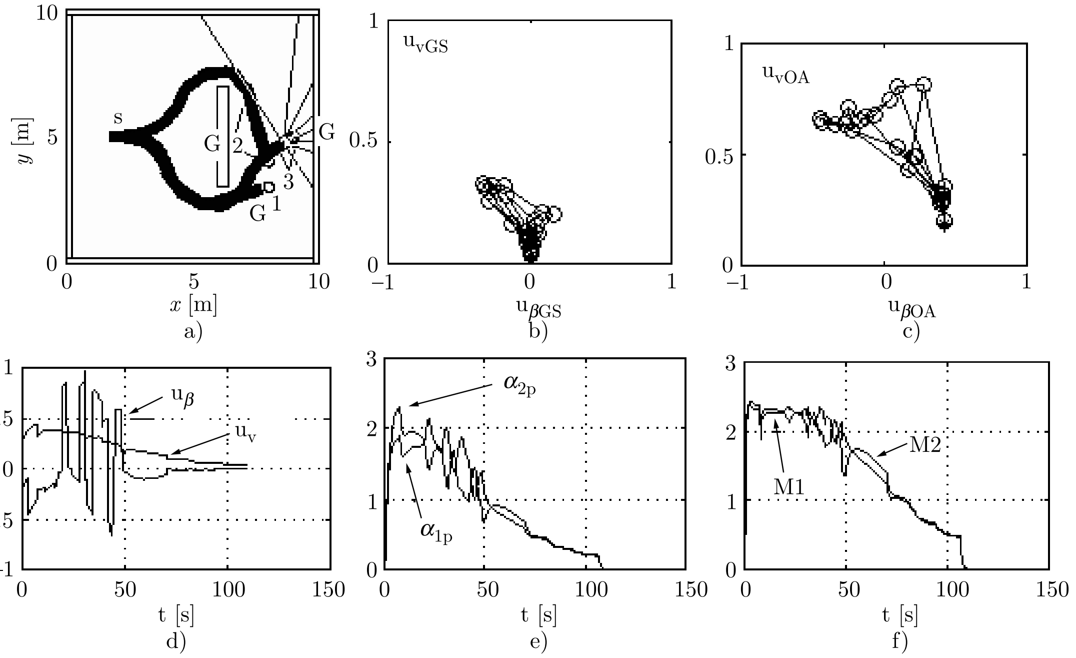


Fig. 7. Unstable and stable equilibrium

For another example as shown in Fig. 7a, all initial positions except $\{(x, y) : x \leq 6, y = 5\}$ end up in the goal position. When initial positions belong to this set, there is a point in which those two behaviours cancel each other. This is an unstable equilibrium, i.e. any small perturbation away from the line will allow the mobile robot to escape towards the goal. If the obstacle was e.g. ∪-shaped, shown in Fig. 7b, however the equilibrium would be stable and adding small perturbation would accomplish nothing. In order to get out of those blocking situation an additional training phase can take place.

The result of the training phase is graphically depicted in Fig. 8. The robot starts from the same initial positions and is forced to reach the goal G1. Then this process is repeated for G2 goal. Received experience of avoiding obstacles, reaching goals G1 and G2 in form of self organizing net, was enough to solve the task of unstable equilibrium point. It is shown as trajectory 3, is generated by the robot. Obtained neurons configurations for elementary behaviours; reach the goal, reach the middle of the free space, are presented in Figs. b,c respectively. The action u_{B} generated by SOM shown in Fig. 8d. Generated angle velocities should be realized by neural control system introduced in Fig. 8e. To realize these signals, moments propelling driving wheels M1, M2 are needed, shown in Fig. 8f.



a)

b)

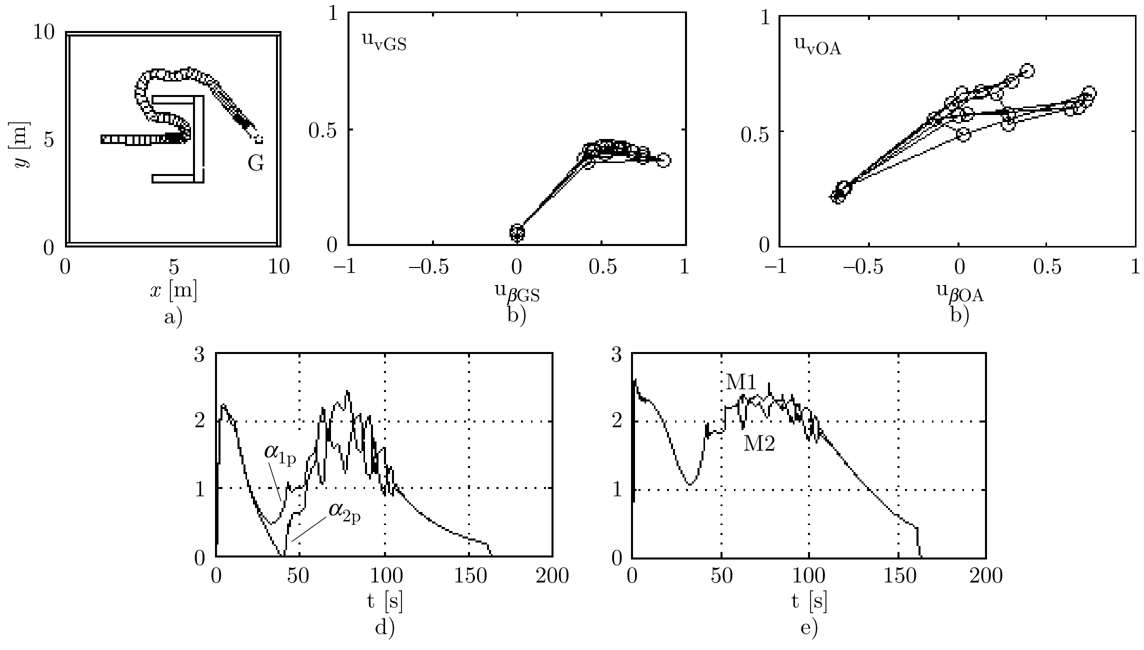
c)

d)

e)

f)

Fig. 8. The numerical results for solving unstable equilibrium



a)

b)

b)

d)

e)

Fig. 9. The numerical results for solving stable equilibrium

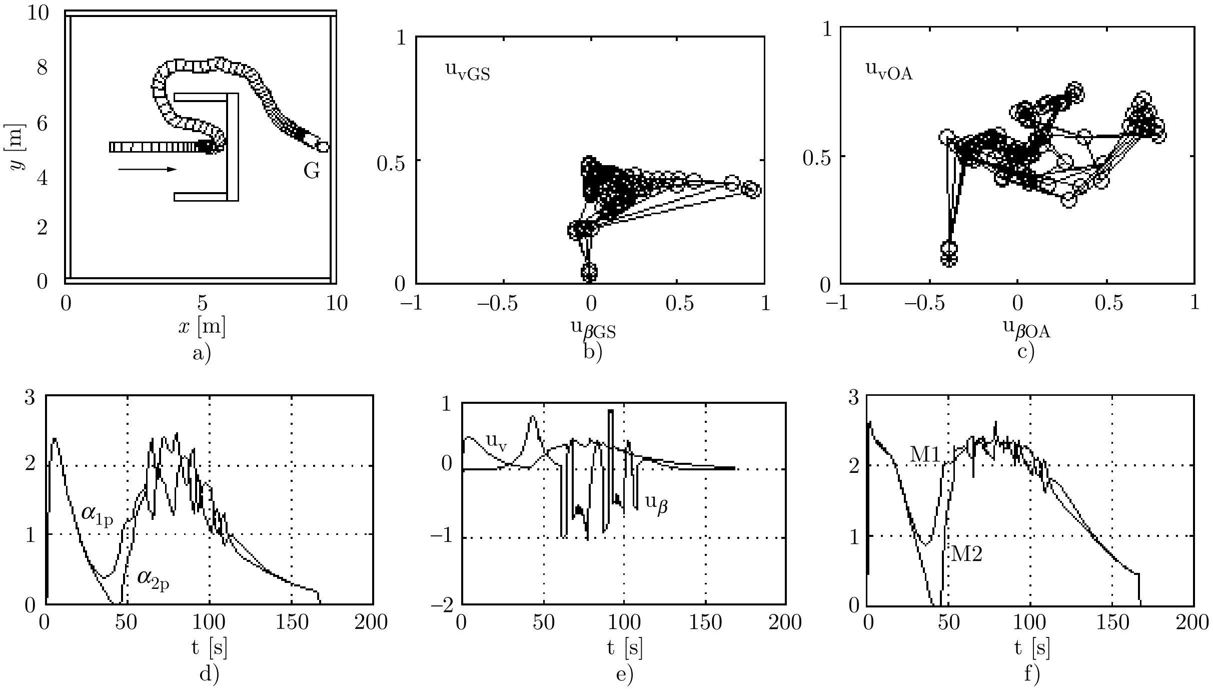


Fig. 10. Another numerical results for solving stable equilibrium

Similar SOM learning procedure, is implemented to solve the task of stable equilibrium point generated by concave obstacle. Received signals of this experiment are presented in Fig. 9.

Another test proved that increasing network dimensions to 10×10 neurons, provides the solution of this task without additional learning procedure. At the beginning of the learning each neuron, to small random value, was initialised. Solutions confirm this conclusion as depicted in Fig. 10.

5. Conclusion

In this work, the decision of using a SOM-like network seems to be confirmed by its data of topology-conserving character which is supposed to favour in some way the learning of suitable perception-action pairs for planning and behaviour control of wheeled mobile robot in the unknown environment. In our considerations the learning machine is the Kohonen Map. We observed that this neural network model can solve paths planning in complex unknown environment. A strategy of reactive navigation was developed including two main behaviours: a reaching the middle of collision-free space behaviour, and a goal-seeking behaviour. The paper presents new numerical results obtained from tasks: an unstable equilibrium, and a stable equilibrium. In both cases an additional training phase of SOM can take place and its efficacy was established.

References

[1] R.C. Arkin, *Behaviour-Based Robotics*, The MIT Press, 1998.

[2] M. Benreguieg, H. Maaref and C. Barret, “ Navigation of an autonomus mobile robot by coordination of behaviours”, *Proceedings of 3rd IFAC Symposium on Intelligent Autonomous Vehicles*, Madrid, Spain, 589–594 (1998).

[3] J. Berenstain and J. Koren, “Real time obstacle avoidance for fast mobile robots”, *IEEE Transaction on Systems, Man, and Cybernetics* 19(5), 1179–1186 (1989).

[4] D. Driankov and A. Saffiotti, ed., *Fuzzy Logic Techniques for Autonomous Vehicle Navigation*, Springer-Verlag, 2001.

[5] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots”, *The International Journal of Robotics Research* 5(1), (1986).

[6] J.C. Latombe, *Robot Motion Planning*, Kluwer Academic Publishers, 1991.

[7] A.M.S. Zalzala and A.S. Morris, *Neural Networks for Robotic Control*, Ellis Horwood, 1996.

[8] T. Kohonen, “Self-organising maps”, vol. 30, Springer Series, in: *Information Sciences*, Springer, Berlin, Heidelberg, 1995.

[9] K.H. Low, W.K. Leow and M.H. Ang Jr., “Integrated planning and control of mobile robot with self-organising neural network”, in: *Proceedings of IEEE International Conference on Robotics and Automation* 4, 3870–3875 (2002).

[10] J.R. Millan, “Reinforcement learning of goal-directed obstacle-avoiding reaction strategies in an autonomous mobile robot”, *Robotics and Autonomous Systems* 15(3), 275-299 (1995).

[11] L.M. Gambardella and C. Versino, “Learning high-level navigation strategies from sensor information and planner experience”, *Proc. PerAc94, From Perception to Action Conference*, Lausanne, Switzerland, September 7–9, 428-431 (1994).

[12] Z. Hendzel, "Neural network reactive navigation and control of wheeled mobile robot", *Advances in Soft Computing*, eds., Rutkowski, Kacprzyk,Phisica-Verlag, 686–691 (2003).

[13] Z. Hendzel, "Fuzzy reactive control of wheeled mobile robot", *J. Theor. Appl. Mech.* 42(3) 503–517 (2004).

[14] Z. Hendzel, "Fuzzy combinier of behaviours for reactive control of wheeled mobile robot", (Eds.), L. Rutkowski et al., in: *Artificial Intelligence and Soft Computing*, Springer-Verlag Berlin Heidelberg, 774–779 (2004).

[15] M. Giergiel, Z. Hendzel and W. Żylski, *Modelling and Control of Wheeled Mobile Robots*, WNT, Warsaw, 2002, (in Polish).

[16] F.L. Levis, K. Liu and A. Yesildirek, "Neural net robot controller with guaranteed tracking performance", *IEEE Transaction on Neural Networks* 6(3) 703–715 (1995).

[17] J.J. Slotine and W. Li, *Applied Non-linear Control*, Prentice Hall, New Jersey, 1991.

[18] H. Ritter and K. Schulten, "Extending Kohonen's Self-Organising Mapping Algorithm to Learn Ballistic Movements", (eds.) R. Eck-miller and E. von der Marlsburg, *Neural Computers*, Springer, Heidelberg, 1987.