

ROBUST NEURAL NETWORKS CONTROL OF OMNI-MECANUM WHEELED ROBOT WITH HAMILTON-JACOBI INEQUALITY

ZENON HENDZEL

Rzeszow University of Technology, Faculty of Mechanical Engineering and Aeronautics, Rzeszów, Poland

e-mail: zenhen@prz.edu.pl

This paper presents a novel approach to the problem of controlling mechanical objects of unspecified description, considering variable operating conditions. The controlled object is a mobile robot with mecanum wheels (MRK_M). To solve the control task, taking into account compensation for nonlinearity and the object variable operating conditions, the Lyapunov stability theory is applied, including the Hamilton-Jacobi (HJ) inequality. A neural network with basic sigmoid functions is used to compensate for the nonlinearity and variable operating conditions of the robot. A simulation example is provided in order to evaluate the analytical considerations. The simulation results obtained confirmed high accuracy of the predicted robot motion in variable operating conditions.

Keywords: mechatronics, mobile robot, mecanum wheels, Hamilton-Jacobi inequality

1. Introduction

The motivation for analytical considerations of the behaviour of a mobile robot with Swedish wheels (Canudas de Wit *et al.*, 1996), referred to in the literature as mecanum wheels (Becker *et al.*, 2014; Abdelrahman *et al.*, 2014; Hendzel and Rykała, 2015) comes from the fact that there is a relatively small amount of literature in this area, especially with regards to the impact of resistance to motion and variable operating conditions on the quality of motion and its control in real time. Most often, control synthesis for these objects is carried out based on kinematics equations (Siegwart *et al.*, 2011; Tai *et al.*, 2011). Kinematics control does not take into account nonlinearity of the robot and variable operating conditions. The literature offers solutions which take into consideration dynamic equations of motion (Han *et al.*, 2009; Park *et al.*, 2010; Tsai *et al.*, 2010) typically in a simplified form not including, for example, resistance to motion. The work of (Lin and Shih, 2013) includes control synthesis for a mobile robot with mecanum wheels based on the Lagrange equations using an adaptive control algorithm. The current paper presents a new approach to the problem of controlling a mobile robot with mecanum wheels as a mechanical object of unspecified description, considering variable operating conditions. To solve the mobile robot control task, the Lyapunov stability theory is applied, including the HJ inequality. The method of synthesis of robust neural networks control proposed in this paper aims at extending the existing solutions with a particular emphasis on its application in the field of control methods for intelligent mobile wheeled robots. In Section 2, kinematic and dynamic equations for MRK_M motion are formulated. Section 3 provides a theoretical basis of dissipation of nonlinear systems and L_2 stability. In Section 4, the assumed structure of the neural network is discussed, and its description is given. The synthesis of robust neural networks control of MRK_M, including the HJ inequality, is provided in Section 5. Section 6 contains a simulation example and simulation results of the adopted solution.

2. Mobile wheeled robot

When describing motion of mobile wheeled robots, we are interested in the issues of kinematics and dynamics of such systems (Żyłski, 1996). From the perspective of controlling these objects in the areas of both kinematics and dynamics, we solve reverse tasks. To describe kinematics of a mobile robot with mecanum wheels, the model shown in Fig. 1 is adopted. In this figure, x , y and z are the axes of the fixed system. The basic units of this model are: frame 5 and the driving units. The driving unit consists of wheels 1, 2, 3, 4, which are mounted on semi-axles and set in motion by a drive module associated with a given wheel. The points A_1, A_2, A_3, A_4 lie in their centres of symmetry. These wheels rotate around their own axes which do not change their position relative to the frame. The rollers are placed on the periphery of the wheels, and they are set at an angle of $\alpha = \pi/4$ radians, relative to the axis of the driving wheel. Figure 1c shows appropriate geometric dimensions and characteristic points of the system. Point S is the centre of mass of the frame, point H is the point lying on the axis A_1A_2 halfway between these points. Point B , like point H , is the point belonging to the frame. The angle β is the angle of temporary rotation of the frame. Figures 1a and 1b exemplify the angular velocity vectors of wheel 2, ω_2 and roller ω_{r2} , respectively. To describe kinematics of any point in the system, it is useful to give the kinematics equation. Knowing the geometry of the system and applying classical methods used in mechanics, the description of the kinematics of the analysed system is as follows

$$\begin{aligned} \dot{x}_s \cos(\beta - \alpha) + \dot{y}_s \sin(\beta - \alpha) - \dot{\beta}(l \cos \alpha + l_1 \sin \alpha) &= \omega_1(R + r) \cos \alpha \\ \dot{x}_s \cos(\beta + \alpha) + \dot{y}_s \sin(\beta + \alpha) + \dot{\beta}(l \cos \alpha + l_1 \sin \alpha) &= \omega_2(R + r) \cos \alpha \\ \dot{x}_s \cos(\beta + \alpha) + \dot{y}_s \sin(\beta + \alpha) - \dot{\beta}(l \cos \alpha + l_1 \sin \alpha) &= \omega_3(R + r) \cos \alpha \\ \dot{x}_s \cos(\beta - \alpha) + \dot{y}_s \sin(\beta - \alpha) + \dot{\beta}(l \cos \alpha + l_1 \sin \alpha) &= \omega_4(R + r) \cos \alpha \end{aligned} \quad (2.1)$$

The adopted description will enable determination of the reverse kinematics task, which will be the set motion trajectory for the S point of the robot in the control system.

When describing the dynamics of complex systems such as mobile wheeled robots, especially robots with mecanum wheels, Lagrange's equations with multipliers or Maggi's equations can be used (Giergiel *et al.*, 2002). To determine the dynamic equations of MRK_M motion, Maggi's equations are used (Żyłski, 1996) with an additional disturbance element, which are convenient for synthesis of control. In the analysed case, take the general form

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F}(\boldsymbol{\omega}) + \boldsymbol{\tau}_d(t) = \mathbf{u} \quad (2.2)$$

where the vector of disturbances meets the restriction $\|\boldsymbol{\tau}_d(t)\| < b$, $b = \text{const} > 0$, matrices \mathbf{M} , \mathbf{C} and vectors \mathbf{F} , \mathbf{u} take the following form

$$\begin{aligned} \mathbf{M}(\mathbf{q}) &= \begin{bmatrix} a_1 \sin \beta + a_2 \cos(\beta - \alpha) & a_2 \sin(\beta - \alpha) - a_1 \cos \beta & -a_3 \\ a_2 \cos(\beta + \alpha) - a_8 \sin(\beta - \alpha) & a_2 \sin(\beta + \alpha) - a_8 \cos(\beta - \alpha) & -a_4 \\ a_1 \cos \beta + a_2 \cos(\beta - \alpha) & a_2 \sin(\beta - \alpha) + a_1 \sin(\beta - \alpha) & a_3 \end{bmatrix} \\ \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) &= \begin{bmatrix} -a_2 \sin(\beta - \alpha) \dot{\beta} & a_2 \cos(\beta - \alpha) \dot{\beta} & 0 \\ -a_2 \sin(\beta + \alpha) \dot{\beta} & a_2 \cos(\beta + \alpha) \dot{\beta} & 0 \\ -a_2 \sin(\beta - \alpha) \dot{\beta} & a_2 \sin(\beta - \alpha) \dot{\beta} & 0 \end{bmatrix} \\ \mathbf{F}(\boldsymbol{\omega}) &= \begin{bmatrix} a_5 \operatorname{sgn} \omega_1 \\ a_6 \operatorname{sgn} \omega_3 \\ a_7 \operatorname{sgn} \omega_4 \end{bmatrix} \quad \mathbf{q} = \begin{bmatrix} x_s \\ y_s \\ \beta \end{bmatrix} \quad \mathbf{u} = \begin{bmatrix} M_1 \\ M_3 \\ M_4 \end{bmatrix} \quad \boldsymbol{\tau}_d(t) = \begin{bmatrix} \boldsymbol{\tau}_{d1} \\ \boldsymbol{\tau}_{d3} \\ \boldsymbol{\tau}_{d4} \end{bmatrix} \end{aligned} \quad (2.3)$$

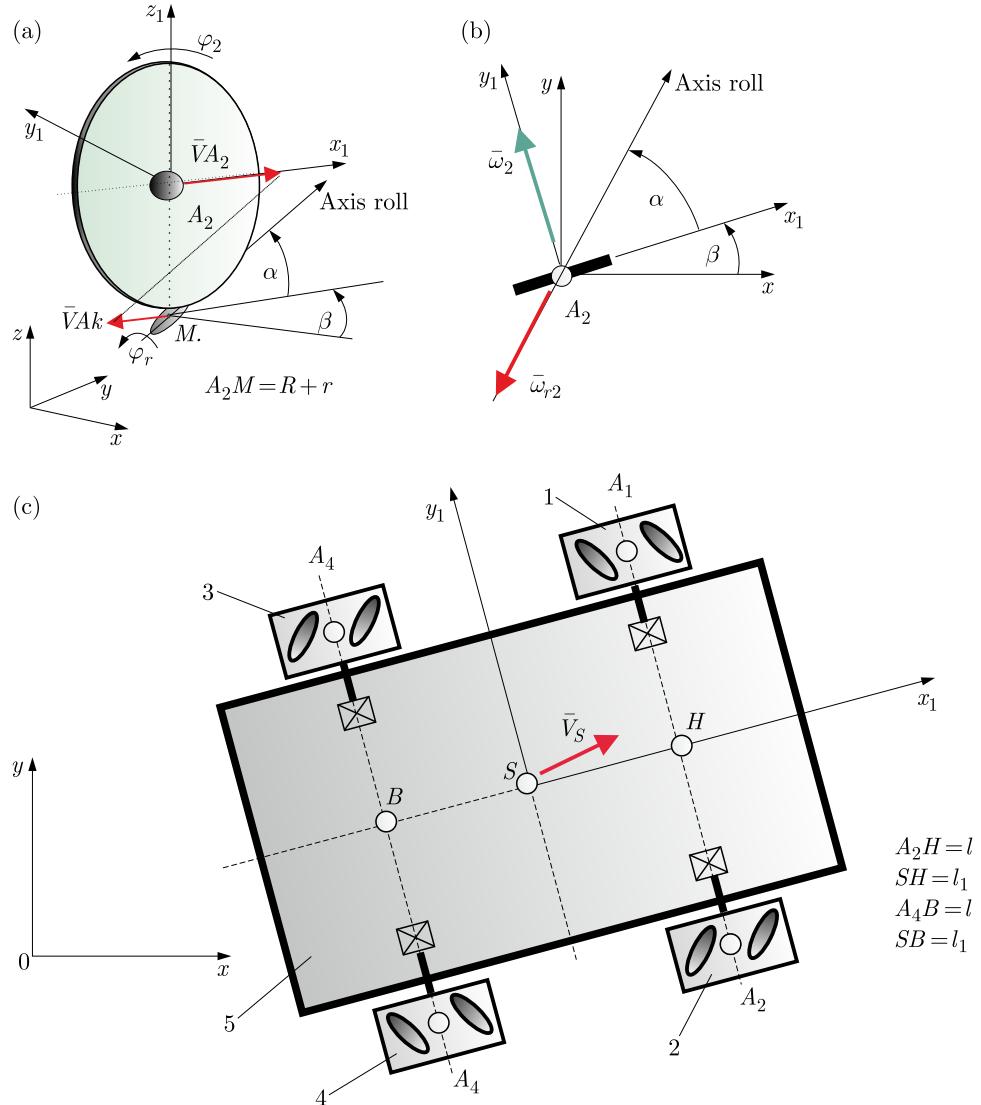


Fig. 1. MRK_M model

Since the analysed robot has 3 degrees of freedom and four drive modules, it is an object of an “over actuated” type, the missing moment M_2 is determined on the basis of the power balance of the drive systems

$$M_2 = M_3 + M_4 - M_1 \quad (2.4)$$

The vector $\mathbf{a} = [a_1, \dots, a_8]^T$ contains parameters resulting from the geometry, mass distribution and the robot resistance to motion. We further assume that the trajectory of the robot motion $\mathbf{q}_d(t) \in \mathbb{R}^3$ and its derivatives are known. Equation (2.2) allows solving the direct and inverse dynamics tasks. The direct task will be used in simulation of the controlled object, and the inverse task in the feedback system will be used to determine the values of the driving moments of the mobile robot wheels, i.e. the control vector.

3. Continuous dissipative systems and L_2 gain

From the perspective of control theory, the dissipation and L_2 gain properties of dynamic non-linear systems are important in the control synthesis of these systems in the context of the

input-output relationship. Further considerations will focus on finite energy signals. This approach to control synthesis is important when considering the effects of disturbances in closed control systems.

Definition 2.1 (Abu-Khalaf and Lewis, 2006; van der Schaft, 1992, 2000). For a given continuous function $f(t) : [0, \infty) \rightarrow \mathbb{R}^n$, its norm L_p , $\|f(t)\|_{L_p}$ is defined as

$$\|f(t)\|_{L_p} = \left(\int_0^\infty \|f(t)\|_p^p dt \right)^{1/p} \quad (3.1)$$

and for $p = 2$ we have

$$\|f(t)\|_{L_2} = \left(\int_0^\infty \|f(t)\|^2 dt \right)^{1/2} \quad (3.2)$$

In order to define dissipation of the system, let us consider the description of the dynamic system in the form

$$\dot{\mathbf{x}}_o = \mathbf{f}_x(\mathbf{x}_o) + \mathbf{b}(\mathbf{x}_o)\mathbf{d}(t) \quad \mathbf{z}(t) = \mathbf{g}(\mathbf{x}_o) \quad (3.3)$$

where $\mathbf{x}_o(t) \in \mathbb{R}^n$ is the state vector, $\mathbf{f}_x(\mathbf{x}_o) \in \mathbb{R}^n$, $\mathbf{b}(\mathbf{x}_o) \in \mathbb{R}^{n \times m}$, $\mathbf{g}(\mathbf{x}_o) \in \mathbb{R}^p$ and $\mathbf{d}(t) \in \mathbb{R}^m$ is the disturbances signal, and $\mathbf{z}(t) \in \mathbb{R}^p$ is the system output signal, which can be interpreted as a tracking control error, additionally $\mathbf{f}_x(0) = \mathbf{0}$ and $\mathbf{x}_o = \mathbf{0}$ is the system equilibrium point.

Definition 2.2 (Abu-Khalaf, Lewis, 2006; Fariwata *et al.*, 2000; Nash, 1951; Slotine and Li, 1991; van der Schaft, 1992). The dynamic system described by equation (3.3) with the supplied rate $w(\mathbf{d}, \mathbf{z})$ is dissipative if there exists a function $V(\mathbf{x}_o) \geq 0$ which is interpreted as a storage function, such that there is a dissipation inequality

$$V(\mathbf{x}_o(t_1)) - V(\mathbf{x}_o(t_0)) \leq \int_{t_0}^{t_1} w(\mathbf{d}(t), \mathbf{z}(t)) dt \quad (3.4)$$

Generally, it can be stated that the dynamic system is dissipative if it loses cumulative energy during operation until the state of the system reaches the equilibrium point.

Definition 2.3 (Abu-Khalaf and Lewis, 2006; Basar and Bernard, 1995; Lewis *et al.*, 2012). Dynamic system (3.3) has L_2 gain less than or equal to γ , for $\gamma \geq 0$, if the following inequality is true

$$\|\mathbf{z}(t)\|_{L_2} \leq \gamma \|\mathbf{d}(t)\|_{L_2} \quad (3.5)$$

This means that the analysed system has $L_2 < \gamma$ gain if there exists $0 \leq \hat{\gamma} < \gamma$ such that equation (3.5) is true for $\hat{\gamma}$. As has been demonstrated (van der Schaft, 2000), if the level of energy supplied to the system is defined as $w(\mathbf{d}(t), \mathbf{z}(t)) = \gamma^2 \|\mathbf{d}(t)\|^2 - \|\mathbf{z}(t)\|^2$ and the system is dissipative, then the dynamic system has gain L_2 , i.e. it is stable. The determination of dissipativity of the system and, hence, the stability of the analysed nonlinear L_2 system requires demonstrating that the function $V(\mathbf{x}_o)$ is bounded from below. Demonstrating this condition requires solving the optimization problem, which comes down to solving the HJ equation or HJ inequality (Basar and Bernard, 1995), written as

$$\dot{V}(\mathbf{x}_o) = \frac{\partial V(\mathbf{x}_o)}{\partial \mathbf{x}_o} [\mathbf{f}_x(\mathbf{x}_o) + \mathbf{b}(\mathbf{x}_o)\mathbf{d}(t)] \leq \frac{1}{2} (\gamma^2 \|\mathbf{d}(t)\|^2 - \|\mathbf{z}(t)\|^2) \quad (3.6)$$

Satisfying this inequality leads to the solution $V(\mathbf{x}_o) \geq 0$ for every $\mathbf{d}(t) \in \mathbb{R}^m$ and $\mathbf{z}(t) = \mathbf{g}(\mathbf{x}_o)$. Determining the system gain L_2 as

$$J = \sup_{d \neq 0} \frac{\|\mathbf{z}(t)\|_{L_2}}{\|\mathbf{d}(t)\|_{L_2}} \quad (3.7)$$

this equation can be interpreted as an indicator of the system resistance to interference. The smaller it is, the more the designed control system is robust to disturbance, provided that inequality (3.6) is satisfied, i.e. for

$$\dot{V}(\mathbf{x}_o) \leq \frac{1}{2}(\gamma^2 \|\mathbf{d}(t)\|^2 - \|\mathbf{z}(t)\|^2) \quad (3.8)$$

$J \leq \gamma$ occurs. To solve the robust neural network control of MRK_M motion, we will apply HJ inequality (3.8) with the evaluation of the control by (3.7).

4. Neural network structure

Due to nonlinearity as well as complexity of the structure of dynamic equations of motion of mobile wheeled robots and the need to include a mathematical model in motion control algorithms for these objects, the application of artificial neural networks theory is an alternative and attractive approach to solve these tasks. From the perspective of control theory, the possibilities of approximation of nonlinear mappings are the most important properties of neural networks. These properties will be used in synthesis of neural networks control for the implementation of MRK_M tracking motion (Hendzel, 2007). In the real-time control, linear networks are used due to weights, with the structure shown in Fig. 2.

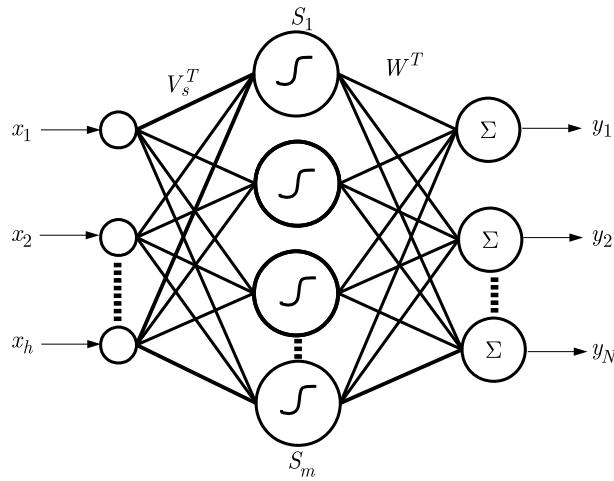


Fig. 2. Neural network structure

The input layer weight matrix \mathbf{V}_s is a constant matrix generated randomly. The network input-output mapping from Fig. 2 has the form (Lewis *et al.*, 1999)

$$y_i = \sum_{j=1}^m \left[w_{ij} S \left(\sum_{k=1}^h v_{jk} x_k + v_{j0} \right) + w_{i0} \right] \quad i = 1, 2, \dots, N \quad (4.1)$$

Assuming the element of the input vector to be $\mathbf{x}_0 \equiv \mathbf{1}$ and the vector of threshold values $[v_{10}, v_{20}, \dots, v_{m0}]^T$ as the first column of the matrix \mathbf{V}_s^T , we get

$$\mathbf{y} = \mathbf{W}^T \mathbf{S}(\mathbf{V}_s^T \mathbf{x}) = \mathbf{W}^T \varphi_x(\bar{\mathbf{x}}) \quad (4.2)$$

where $\bar{\mathbf{x}} = \mathbf{V}_s^T \mathbf{x}$, $\mathbf{S} = [1, S_1(\cdot), S_2(\cdot), \dots, S_m(\cdot)]^T$ is a vector describing neuron functions, whose first element is equal to 1, and the vector $[w_{10}, w_{20}, \dots, w_{N0}]^T$ is the first column of the matrix \mathbf{W}^T in order to account for the threshold value of linear neurons in the network output layer.

5. Synthesis of robust neural networks control of MRK_M motion

The purpose of the synthesis of the control algorithm is to determine the control rule and the rule for adaptation of network weights, for which the trajectory of the selected point S of the robot, Fig. 1, will coincide with the set trajectory (Hendzel and Szuster, 2012, 2015; Szuster and Hendzel, 2018), despite the occurrence of variable operating conditions of the robot. To this end, the tracking error $\mathbf{e} \in \mathbb{R}^3$, the generalized error \mathbf{s} and the auxiliary vector \mathbf{v} are defined as

$$\mathbf{e} = \mathbf{q}_d - \mathbf{q} \quad \mathbf{s} = \dot{\mathbf{e}} + \Lambda \mathbf{e} \quad \mathbf{v} = \dot{\mathbf{q}}_d + \Lambda \mathbf{e} \quad (5.1)$$

where Λ is a diagonal matrix, positively defined. Then equation (2.2) can be transformed into

$$\mathbf{M}(\mathbf{q})\dot{\mathbf{s}} = -\mathbf{u} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{s} + \mathbf{M}\dot{\mathbf{v}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{v} + \mathbf{F}(\boldsymbol{\omega}) + \boldsymbol{\tau}_d(t) \quad (5.2)$$

If we define the nonlinear function

$$\mathbf{f}(\mathbf{x}_r) = \mathbf{M}(\mathbf{q})\dot{\mathbf{v}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{v} + \mathbf{F}(\boldsymbol{\omega}) \quad (5.3)$$

where $\mathbf{x}_r = [\mathbf{v}^T, \dot{\mathbf{v}}^T, \dot{\mathbf{q}}^T, q^T, \boldsymbol{\omega}^T]^T$, then equation (5.2) will be written as

$$\mathbf{M}(\mathbf{q})\dot{\mathbf{s}} = -\mathbf{u} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{s} + \mathbf{f}(\mathbf{x}_r) + \boldsymbol{\tau}_d(t) \quad (5.4)$$

In practice, an approximation of the control compensating for the robot nonlinearity $\mathbf{f}(\mathbf{x}_r)$ is applied. For the approximation of the nonlinearity, the neural network described in Section 4 is used, assuming $\mathbf{x} = \mathbf{x}_r$, $\mathbf{y} = \mathbf{f}(\mathbf{x}_r)$. The nonlinear function $\mathbf{f}(\mathbf{x}_r)$ is written as

$$\mathbf{f}(\mathbf{x}_r) = \mathbf{W}^T \varphi_x(\bar{\mathbf{x}}) + \boldsymbol{\varepsilon} \quad (5.5)$$

where $\boldsymbol{\varepsilon}$ is the approximation error. The estimate the function $\mathbf{f}(\mathbf{x}_r)$ is written as

$$\hat{\mathbf{f}}(\mathbf{x}_r) = \widehat{\mathbf{W}}^T \varphi_x(\bar{\mathbf{x}}) \quad (5.6)$$

where $\widehat{\mathbf{W}}$ is an estimate of weight of the ideal neural network. In addition, it is assumed that ideal network weights are fixed, $\mathbf{W} = \text{const.}$ Using equation (5.6), we adopt the control rule in the following form

$$\mathbf{u} = \widehat{\mathbf{W}}^T \varphi_x(\bar{\mathbf{x}}) + \frac{1}{2} \left(1 + \frac{1}{\gamma^2} \right) \mathbf{s} \quad (5.7)$$

It should be noted that the second component of equation (5.7) has an interpretation of the PD type controller since the second component of (5.7) can be written as

$$K_D \mathbf{s} = K_D \dot{\mathbf{e}} + K_D \Lambda \mathbf{e} \quad (5.8)$$

where $K_D = 0.5(1 + 1/\gamma^2)$. Substituting (5.7) and (5.5) for (5.4), we obtain

$$\mathbf{M}(\mathbf{q})\dot{\mathbf{s}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{s} + \frac{1}{2} \left(1 + \frac{1}{\gamma^2} \right) \mathbf{s} = \tilde{\mathbf{f}}(\mathbf{x}_r) + \boldsymbol{\tau}_d(t) \quad (5.9)$$

where $\tilde{\mathbf{f}}(\mathbf{x}_r)$ is a function of the approximation error $\mathbf{f}(\mathbf{x}_r)$, which is

$$\tilde{\mathbf{f}}(\mathbf{x}_r) = \mathbf{f}(\mathbf{x}_r) - \hat{\mathbf{f}}(\mathbf{x}_r) = \mathbf{W}^T \varphi_x(\bar{\mathbf{x}}) - \widehat{\mathbf{W}}^T \varphi_x(\bar{\mathbf{x}}) + \boldsymbol{\varepsilon} = \widetilde{\mathbf{W}}^T \varphi_x(\bar{\mathbf{x}}) + \boldsymbol{\varepsilon} \quad (5.10)$$

where $\widetilde{\mathbf{W}} = \mathbf{W} - \widehat{\mathbf{W}}$ is the error of estimation of neural network weights. Using equation (5.10), equation (5.9) is written as follows

$$\mathbf{M}(\mathbf{q})\dot{\mathbf{s}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{s} + \frac{1}{2}\left(1 + \frac{1}{\gamma^2}\right)\mathbf{s} = \widetilde{\mathbf{W}}^T \varphi_x(\bar{\mathbf{x}}) + \boldsymbol{\varepsilon} + \boldsymbol{\tau}_d(t) \quad (5.11)$$

In order to apply the considerations described in Section 3, the designations in equation (3.3) are interpreted as

$$\begin{aligned} \dot{\mathbf{x}}_o &= \dot{\mathbf{s}} & \mathbf{x}_o &= \mathbf{s} & \mathbf{b}(\mathbf{x}_o) &= \mathbf{M}(\mathbf{q})^{-1} \\ \mathbf{g}(\mathbf{x}_o) &= \mathbf{s} & \mathbf{d}(t) &= \boldsymbol{\varepsilon} + \boldsymbol{\tau}_d(t) \\ \mathbf{f}_x(\mathbf{x}_o) &= \mathbf{M}(\mathbf{q})^{-1} \left[-\frac{1}{2}\left(1 + \frac{1}{\gamma^2}\right)\mathbf{s} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{s} + \widetilde{\mathbf{W}}^T \varphi_x(\bar{\mathbf{x}}) \right] \end{aligned}$$

In order to synthesize the neural network control of MRK_M motion, the Lyapunov stability theory is used. As a candidate for the Lyapunov function, the following function has been selected

$$V(\mathbf{s}) = \frac{1}{2}\mathbf{s}^T \mathbf{M}(\mathbf{q})\mathbf{s} + \frac{1}{2} \operatorname{tr}(\widetilde{\mathbf{W}}^T \mathbf{P}^{-1} \widetilde{\mathbf{W}}) \quad (5.12)$$

where $\mathbf{P} = \mathbf{P}^T > \mathbf{0}$ is a design matrix. The derivative of the function $V(\mathbf{s})$ relative to time along the trajectory of system (5.11) is

$$\dot{V}(\mathbf{s}) = \mathbf{s}^T \mathbf{M}(\mathbf{q})\dot{\mathbf{s}} + \frac{1}{2}\mathbf{s}^T \dot{\mathbf{M}}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{s} + \operatorname{tr}(\widetilde{\mathbf{W}}^T \mathbf{P}^{-1} \dot{\widetilde{\mathbf{W}}}) \quad (5.13)$$

Substituting from equation (5.11) the expression $\mathbf{M}(\mathbf{q})\dot{\mathbf{s}}$ and taking advantage of the fact that $0.5[\dot{\mathbf{M}}(\mathbf{q}, \dot{\mathbf{q}}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})]$ is a zero matrix (for $\beta(t) = 0$), the following is obtained

$$\dot{V}(\mathbf{s}) = -\mathbf{s}^T \frac{1}{2}\left(1 + \frac{1}{\gamma^2}\right)\mathbf{s} + \operatorname{tr}\{\widetilde{\mathbf{W}}^T [\mathbf{P}^{-1} \dot{\widetilde{\mathbf{W}}} + \varphi_x(\bar{\mathbf{x}})\mathbf{s}^T]\} + \mathbf{s}^T [\boldsymbol{\varepsilon} + \boldsymbol{\tau}_d(t)] \quad (5.14)$$

Selecting the following as the rule for adaptation of neural network weights

$$\dot{\widetilde{\mathbf{W}}} = -\mathbf{P} \varphi_x(\bar{\mathbf{x}}) \mathbf{s}^T \quad (5.15)$$

equation (5.14) will be transformed into

$$\dot{V}(\mathbf{s}) = -\frac{1}{2\gamma^2}\mathbf{s}^T \mathbf{s} - \frac{1}{2}\mathbf{s}^T \mathbf{s} + \mathbf{s}^T [\boldsymbol{\varepsilon} + \boldsymbol{\tau}_d(t)] \quad (5.16)$$

Let D denote transformed equation (3.8)

$$D = \dot{V}(\mathbf{s}) - \frac{1}{2}\gamma^2 \|\boldsymbol{\varepsilon} + \boldsymbol{\tau}_d(t)\|^2 + \frac{1}{2}\|\mathbf{s}\|^2 \quad (5.17)$$

and then, on the basis of (5.16), equation (5.17) will be transformed into

$$D = -\frac{1}{2\gamma^2}\mathbf{s}^T \mathbf{s} - \frac{1}{2}\mathbf{s}^T \mathbf{s} + \mathbf{s}^T [\boldsymbol{\varepsilon} + \boldsymbol{\tau}_d(t)] - \frac{1}{2}\gamma^2 \|\boldsymbol{\varepsilon} + \boldsymbol{\tau}_d(t)\|^2 + \frac{1}{2}\|\mathbf{s}\|^2 \quad (5.18)$$

Because $-0.5\mathbf{s}^T \mathbf{s} + 0.5\|\mathbf{s}\|^2 = 0$, equation (5.18) will take the form

$$\mathbf{s}^T [\boldsymbol{\varepsilon} + \boldsymbol{\tau}_d(t)] - \frac{1}{2\gamma^2}\mathbf{s}^T \mathbf{s} - \frac{1}{2}\gamma^2 \|\boldsymbol{\varepsilon} + \boldsymbol{\tau}_d(t)\|^2 = -\frac{1}{2} \left\| \frac{1}{\gamma} \mathbf{s} + \gamma [\boldsymbol{\varepsilon} + \boldsymbol{\tau}_d(t)] \right\|^2 \leq 0 \quad (5.19)$$

From inequality (5.19) it follows that $D \leq 0$ and, based on equation (5.17), we get

$$\dot{V}(\mathbf{s}) \leq \frac{1}{2}\gamma^2 \|\boldsymbol{\varepsilon} + \boldsymbol{\tau}_d(t)\|^2 - \frac{1}{2}\|\mathbf{s}\|^2 \quad (5.20)$$

Determining $\mathbf{z}(t) = \mathbf{s}(t)$, $\mathbf{d}(t) = \boldsymbol{\varepsilon} + \boldsymbol{\tau}_d(t)$, we have thus shown that condition (3.8) is satisfied, i.e. inequality $J \leq \gamma$ is satisfied for the structure adopted as control rule (5.7) and, furthermore, that the analysed system is stable according to the definition in (2.3).

5.1. Caution

In the proposed control synthesis, based on the Lyapunov stability theory, there is a component of the form $\mathbf{s}^T 0.5[\dot{\mathbf{M}}(\mathbf{q}, \dot{\mathbf{q}}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})]\mathbf{s}$ which takes a value of zero if the matrix in brackets is an obliquely symmetrical matrix. This situation takes place in the analysed case only for the configuration of the robot frame $\beta(t) = 0$. In the case where $\dot{\beta}(t) \neq 0$, the indicated equation does not take place. Then, when determining the expression $\varepsilon_1 = 0.5[\dot{\mathbf{M}}(\mathbf{q}, \dot{\mathbf{q}}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})]\mathbf{s}$, the last component of equation (5.14) needs to be modified to the form $\mathbf{s}^T [\varepsilon + \tau_d(t) + \varepsilon_1]$ and, consequently, equation (5.20) will be $\dot{V}(\mathbf{s}) \leq 0.5\gamma^2\|\varepsilon + \tau_d(t) + \varepsilon_1\|^2 - 0.5\|\mathbf{s}\|^2$, and the expression ε_1 needs to be interpreted as a structural interference.

The next Section presents a simulation example, the purpose of which is to confirm the solutions arrived at in the analytical considerations and to obtain quantitative solutions.

6. Simulation example

In the example, simulation of the suggested solution has been carried out for the movement of the selected point S of the mobile robot on the trajectory set in the form of a loop, consisting of five characteristic stages of motion: moving on a rectilinear track, starting, moving at a fixed speed when, $V_s^* = 0.4$ m/s, moving on a circular path with the radius $R = 0.7$ m, exit from the curve taking into account a transitional period, then moving on the rectilinear track at a fixed speed and braking. For the assumed stages of motion, Figure 3 shows the trajectory set for the point S with the assumed orientation angle of the robot, assuming the initial conditions of motion: $x_s(0) = 1$ m, $y_s(0) = 5$ m, $\beta(0) = 0$.

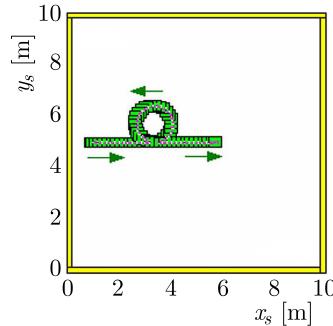


Fig. 3. The trajectory set for motion of the point S and the robot orientation

During motion, there is no change in the orientation angle of the robot frame $\beta(t)$. Figure 4 presents a solution to the task of inverse kinematics. For the time $t \geq 7.5$ s, there is a movement of the point S of the robot on a circular trajectory with the radius $R = 0.7$ m.

Table 1. Values of a_i parameters of a mobile robot

| a_1 | a_2 | a_3 | a_4 | a_5 | a_6 | a_7 | a_8 |
|-------|---------|--------|--------|-------|-------|-------|--------|
| 0.044 | 11.5708 | 2.4558 | 2.4545 | 3.001 | 3.001 | 3.001 | 0.0622 |

Then, the self-rotation angle of wheel 2 increases and the self-rotation angle of wheel 1 decreases resulting from the structure of the robot, as shown in Fig. 4a. The change of these values is shifted in time. The change in motion is confirmed by a change in the angular velocity of the wheels, see Fig. 4b, and the course of angular accelerations of the wheels, see Fig. 4c. When the point S turns an angle $\pi/2$ rad then the angular velocities and angular accelerations have the same values, which is followed by a change in the values and turns of vectors of these kinematics parameters. The obtained parameters of motion of the driving wheels constitute the

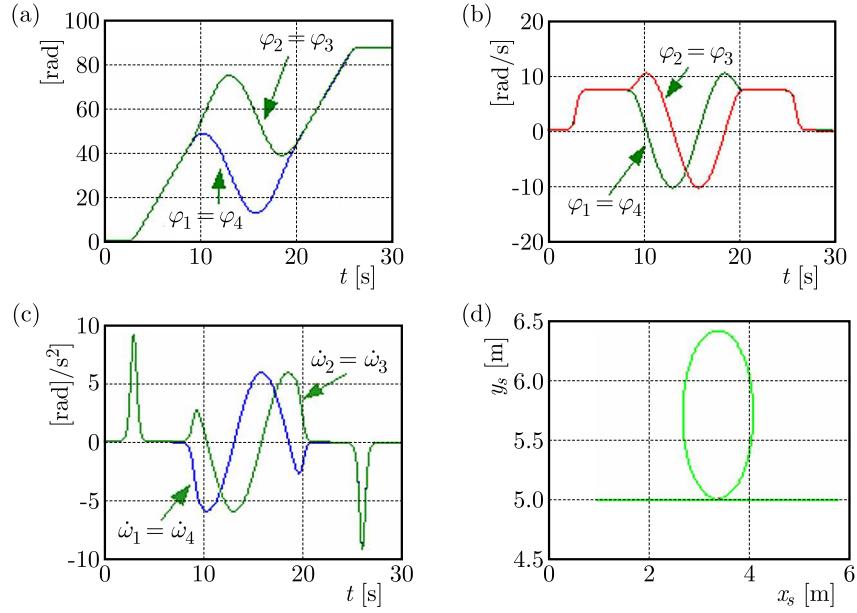


Fig. 4. The parameters set for the movement of the robot wheel and the path of the point S : (a) values of self-rotation angles of the driving wheels, φ_1, φ_4 and φ_2, φ_3 , (b) angular velocity of self-rotation of the driving wheels ω_1, ω_4 and ω_2, ω_3 , (c) angular accelerations $\dot{\omega}_1, \dot{\omega}_4, \dot{\omega}_2, \dot{\omega}_3$, (d) set trajectory for the point S

set trajectory of motion for the control system. For simulation of the control system, values of the robot parameters have been adopted as specified in Table 1 and the remaining data are: $\Lambda = \text{diag}[2, 2, 2]$, $K_D = 0.5(1 + 1/\gamma)$, $\mathbf{P} = \text{diag}[100, \dots, 100]$, $\gamma = 0.2$, $\tau_d = \mathbf{0}$. In order to check the robustness of the proposed solution, it has been also assumed that for the time $t \geq 15$ s there is a parametric disturbance $\mathbf{a} + \Delta\mathbf{a}$ where $\Delta\mathbf{a} = [0, 0, 0, 0, 1, 1, 1, 0]^T$, associated with the change of the rolling friction of the robot wheels. The neural network adopted in simulation to compensate for the robot nonlinearity and the variable operating conditions consist of 6 sigmoidal bipolar neurons whose first layer weights are generated by a random number generator from the range $[-0.1, 0.1]$. A separate neural network for each element is used for approximation of the nonlinear function $\mathbf{f}(\mathbf{x}_r) \in \mathbb{R}^3$. The calculations are made by adopting the Euler method of integration with the time discretization step 0.01 s. In the process of initializing the neural network, zero initial values of the output layer weights are assumed. In order to evaluate quality of the generated robust control and implementation of tracking motion, the following quality indicators have been adopted, restricted to the assessment of motion only for wheels 1 and 2:

- maximum value of the angle of rotation error $e_1 = (\varphi_{1d} - \varphi_1)$ [rad], $e_2 = (\varphi_{2d} - \varphi_2)$ [rad], $e_{max(\cdot)} = \max |e_{(\cdot)}|$ [rad], $\varphi_{1d}, \varphi_{2d}$ are the set angles of wheel rotation, Fig. 4a,
- maximum value of the angular velocity error $\dot{e}_{max(\cdot)} = \max |\dot{e}_{(\cdot)}|$ [rad/s],
- root-mean-squared error for wheel rotation angles, $\varepsilon_1 = \sqrt{(1/n) \sum_{k=1}^n e_{1k}^2}$, $\varepsilon_2 = \sqrt{(1/n) \sum_{k=1}^n e_{2k}^2}$ [rad], where k is the number of subsequent discrete measurements, $n = 3001$ – total number of discrete measurements,
- root-mean-squared error for tracking the given angular velocity $\dot{\varepsilon}_1 = \sqrt{(1/n) \sum_{k=1}^n \dot{e}_{1k}^2}$, $\dot{\varepsilon}_2 = \sqrt{(1/n) \sum_{k=1}^n \dot{e}_{2k}^2}$ [rad/s],
- generalised root-mean-squared error: $s_1 = \dot{e}_1 + \lambda e_1$, $s_2 = \dot{e}_2 + \lambda e_2$, $\nu_1 = \sqrt{(1/n) \sum_{k=1}^n s_{1k}^2}$, $\nu_2 = \sqrt{(1/n) \sum_{k=1}^n s_{2k}^2}$ [rad/s], mean-squared error of distance d_k , $\rho = \sqrt{(1/n) \sum_{k=1}^n d_k^2}$ [m], where $d_k = \sqrt{(x_{Sk} - x_{dSk})^2 + (y_{Sk} - y_{dSk})^2}$ [m], $k = n$ is the distance between the set

(x_{DS}, y_{DS}) , Fig. 4d and implemented (x_S, y_S) locations of the selected point S of the mobile robot in the xy plane during motion,

- maximum distance $d_{max} = \max(d_k)$ [m].

Implementing adopted control algorithm (5.7), patterns of control errors have been obtained as shown in Fig. 5.

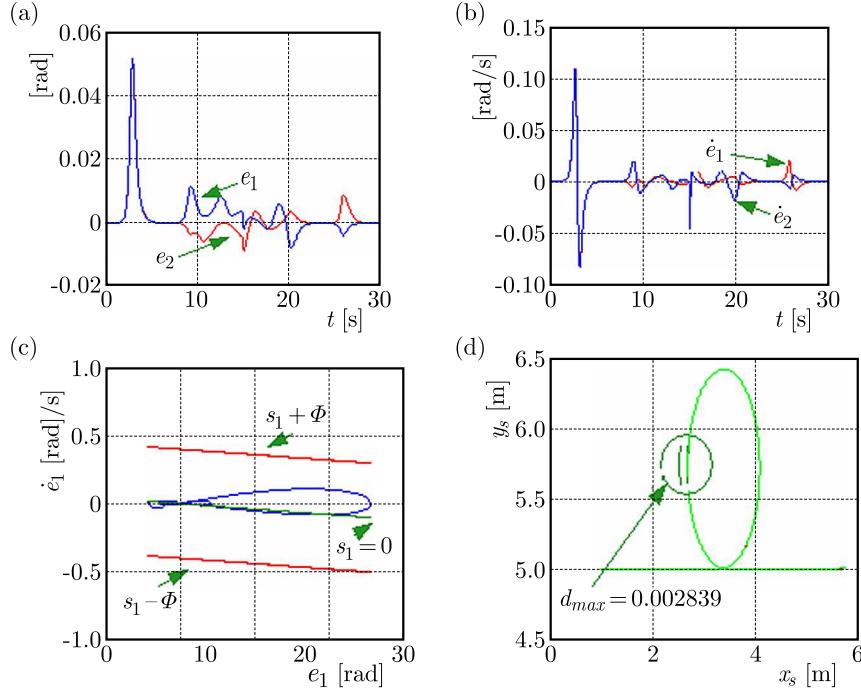


Fig. 5. Results for the obtained errors: (a) patterns of errors of tracking the rotation angles of wheels 1 and 2, (b) patterns of errors of tracking the angular velocity of wheels 1 and 2, (c) pattern of error in the phase plane for the generalized error s_1 , (d) implementation of the set trajectory for the point S

The quantitative assessment of the solution is given in Tables 2 and 3.

Table 2. Values of the quality indicators for the robust neural networks control

| Indicator | e_{imax} [rad] | ε_i [rad] | \dot{e}_{imax} [rad/s] | $\dot{\varepsilon}_i$ [rad/s] | ν_i [rad/s] |
|------------------|------------------|-----------------------|--------------------------|-------------------------------|-----------------|
| wheel 1, $i = 1$ | 0.05139 | 0.007379 | 0.1087 | 0.01529 | 0.02125 |
| wheel 2, $i = 2$ | 0.05184 | 0.007703 | 0.1097 | 0.01578 | 0.02205 |

Table 3. Values of the indicators d_{max} , ρ for the robust neural networks control

| Indicator | d_{max} [m] | ρ [m] |
|-----------|---------------|------------|
| Value | 0.002839 | 0.0004151 |

Having analysed the changes in the individual indicators of the quality of motion implementation in the presence of variable operating conditions, it can be concluded that a high accuracy in the implementation of the adopted method of solution has been obtained. However, Fig. 6 presents patterns of control signals in the presence of variable operating conditions of the object, resulting from the assumed resistance to motion in the form of a change in the rolling friction. The neural networks compensation of the robot nonlinearity plays a predominant role in the control structure, as shown in Fig. 6b, in comparison with the PD control, Fig. 6c. The observed parametric disturbance is particularly visible for driving wheel 2.

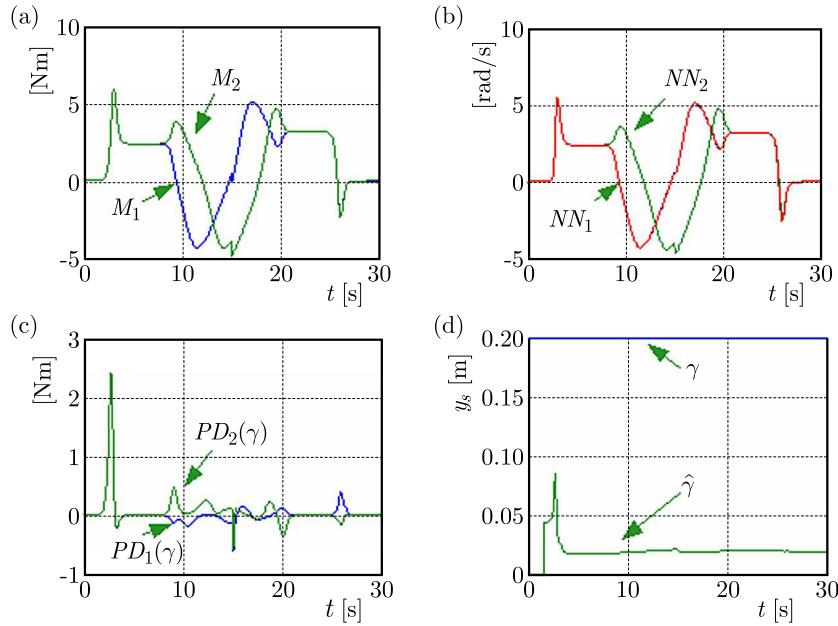


Fig. 6. Results of simulation of the neural networks control system: (a) total control, (b) $NN_{(.)}$ neural networks control, (c) PD control (γ), (d) evaluation of the system robustness $\hat{\gamma} \leq \gamma$

The limitation of the signals shown in Fig. 6 confirms stability of the adopted solution in accordance with Lyapunov's interpretation of stability. Figure 6d presents the results of the solution for the robust neural networks control of MRK_M motion which changes over time, with equation (3.7) constituting the evaluation of this control method. As indicated by the patterns shown in Fig. 6d, the analysed system has gain $L_2 < \gamma$, for $\gamma = 0.2$, and the condition $0 \leq \hat{\gamma} < \gamma$ is met, thus equation (3.5) is satisfied.

7. Summary

This paper presents a novel approach to the problem of controlling mechanical objects of unspecified description considering variable operating conditions. In order to solve the task of controlling a mobile robot with mecanum wheels taking into account the compensation for nonlinearity and variable operating conditions of the object, Lyapunov's stability theory has been applied including the HJ inequality. A neural network with basic sigmoid functions has been used to compensate for the nonlinearity and variable operating conditions of the robot. Additionally, there is a $PD(\gamma)$ controller in the control structure, i.e. the structure of moments driving the wheels. The gain values of the $PD(\gamma)$ controller have been determined based on the HJ inequality, thus formulating an algorithm which is stable and interference-resistant from the perspective of input-output signals.

References

1. ABDELRAHMAN M., ZEIDIS I., BONDAREV O., ADAMOV B., BECKER F., ZIMMERMANN K., 2014, A description of the dynamics of a four wheel mecanum mobile system as a basis for a platform concept for special purpose vehicles for disabled persons, *58-th Ilmenau Scientific Colloquium*
2. ABU-KHALAF M., LEWIS F.L., 2006, *Nonlinear H_2/H_∞ Constrained Feedback Control*, Springer Verlag, London

3. BASAR T., BERNARD P., 1995, *H_∞ Optimal Control and Related Minimax Design Problems*, Birkhäuser
4. BECKER F., BONDAREV O., ZEIDIS I., ZIMMERMANN K., ABDELRAHMAN M., ADAMOV B., 2014, An approach to the kinematics and dynamics of a four-wheeled mecanum vehicles, *Scientific Journal of IfToMM „Problems of Mechanics”*, special issue, **2**, 55, 27-37
5. CANUDAS DE WIT C., SICILIANO B., BASTIN G., 1996, *Theory of Robot Control*, Springer-Verlag, London
6. FARIWATA S.S., FILEV D., LANGARI R., 2000, *Fuzzy Control*, John Wiley & Sons, Ltd, England
7. GIERGIEL M., HENDZEL Z., ŹYLSKI W., 2002, *Modelling and Control of Mobile Robots* (in Polish), PWN, Warszawa
8. HAN K.-L., CHOI O.-K., KIM J., KIM H., LEE J.S., 2009, Design and control of mobile robot with mecanum wheel, *ICROS-SICE International Joint Conference*, Fukuoka International Congress Center, Japan, 2932-2937
9. HENDZEL Z., 2007, An adaptive critic neural network for motion control of a wheeled mobile robot, *Nonlinear Dynamics*, **50**, 4, 849855, DOI 10.1007/s11071-007-9234-1
10. HENDZEL Z., RYKAŁA Ł., 2015, Description of kinematics of a mobile wheeled robot with wheels of the mecanum type (in Polish), *Modelowanie Inżynierskie*, **57**, 26, 5-12
11. LEWIS F.L., JAGANNATHAN S., YESILDIREK A., 1999, *Neural Network Control of Robot Manipulators and Nonlinear Systems*, Taylor & Francis, London
12. LEWIS F.L., VRABIE D.L., SYROMS V.L., 2012, *Optimal Control*, 3rd ed., Wiley & Sons, New Jersey
13. LIN L.-C., SHIH H.-Y., 2013, Modeling and adaptive control of an omni-mecanum-wheeled robot, *Intelligent Control and Automation*, **4**, 166-179
14. NASH J., 1951, Non-cooperative games, *Annals of Mathematics*, **2**, 286-295
15. PARK J., KIM S., KIM J., KIM S., 2010, Driving control of mobile robot with mecanum wheel using fuzzy inference system, *International Conference on Control, Automation and Systems*, Gyeonggi-do, 2519-2523
16. SIEGWART R., NOURBAKHSH I.R., SCARAMUZZA D., 2011, *Introduction to Autonomous Mobile Robots*, 2nd Edition, MIT Press, London
17. SLOTINE J.-J.E., LI W., 1991, *Applied Nonlinear Control*, Prentice Hall, NJ
18. SZUSTER M., HENDZEL Z., 2018, *Intelligent Optimal Adaptive Control for Mechatronic Systems*, Springer International Publishing AG
19. TSAI C.-C., WU H.-L., 2010, Nonsingular terminal sliding control using fuzzy wavelet networks for mecanum- wheeled omni-directional vehicles, *IEEE International Conference on Fuzzy Systems*, 1-6
20. TSAI C.-C., TAI F.-C., LEE Y.-R., 2011, Motion controller design and embedded realization for mecanum wheeled omni-directional robots, *Proceedings of the 8th World Congress on Intelligent Control and Automation*, Taiwan, 546-551
21. VAN DER SCHAFT A.J., 1992, L2-gain analysis of nonlinear systems and nonlinear state feedback H_∞ control, *IEEE Transactions on Automatic Control*, **37**, 6, 770-784
22. VAN DER SCHAFT A.J., 2000, *L2-Gain and Passivity Techniques in Nonlinear Control*, Springer Verlag, London
23. ŹYLSKI W., 1996, *Kinematics and Dynamics of Wheeled Mobile Robots* (in Polish), Rzeszow University of Technology Publishing Press, Rzeszow